

# Package ‘RCy3’

October 14, 2021

**Type** Package

**Title** Functions to Access and Control Cytoscape

**Version** 2.12.4

**Date** 2021-08-09

**Imports** httr, methods, RJSONIO, XML, utils, BiocGenerics, igraph, stats, graph, R.utils, dplR, uchardet, glue, RCurl, base64url, base64enc, IRkernel, RColorBrewer

**Suggests** BiocStyle, knitr, rmarkdown

**SystemRequirements** Cytoscape (>= 3.7.1), CyREST (>= 3.8.0)

**Description** Vizualize, analyze and explore networks using Cytoscape via R. Anything you can do using the graphical user interface of Cytoscape, you can now do with a single RCy3 function.

**License** MIT + file LICENSE

**URL** <https://github.com/cytoscape/RCy3>

**BugReports** <https://github.com/cytoscape/RCy3/issues>

**LazyLoad** yes

**biocViews** Visualization, GraphAndNetwork, ThirdPartyClient, Network

**NeedsCompilation** no

**VignetteBuilder** knitr

**RoxygenNote** 7.1.1

**git\_url** <https://git.bioconductor.org/packages/RCy3>

**git\_branch** RELEASE\_3\_13

**git\_last\_commit** 5bb5011

**git\_last\_commit\_date** 2021-08-09

**Date/Publication** 2021-10-14

**Author** Alex Pico [aut, cre] (<<https://orcid.org/0000-0001-5706-2163>>),  
Tanja Muetze [aut],  
Paul Shannon [aut],  
Ruth Isserlin [ctb],  
Shraddha Pai [ctb],

Julia Gustavsen [ctb],  
 Georgi Kolishovski [ctb],  
 Yihang Xin [ctb]

**Maintainer** Alex Pico <alex.pico@gladstone.ucsf.edu>

## R topics documented:

.getDefaultSandbox . . . . .	10
.getRequester . . . . .	10
addAnnotationText . . . . .	11
addCyEdges . . . . .	12
addCyNodes . . . . .	13
AddToGroup . . . . .	14
analyzeNetwork . . . . .	15
applyFilter . . . . .	15
bundleEdges . . . . .	17
checkNotebookIsRunning . . . . .	17
checkRunningRemote . . . . .	18
clearEdgeBends . . . . .	18
clearEdgePropertyBypass . . . . .	19
clearNetworkCenterBypass . . . . .	20
clearNetworkPropertyBypass . . . . .	20
clearNetworkZoomBypass . . . . .	21
clearNodeOpacityBypass . . . . .	22
clearNodePropertyBypass . . . . .	23
clearSelection . . . . .	24
cloneNetwork . . . . .	24
closeSession . . . . .	25
collapseGroup . . . . .	26
commandEcho . . . . .	27
commandOpenDialog . . . . .	27
commandPause . . . . .	28
commandQuit . . . . .	29
commandRunFile . . . . .	29
commandsAPI . . . . .	30
commandsGET . . . . .	30
commandsHelp . . . . .	31
commandSleep . . . . .	32
commandsPOST . . . . .	32
commandsRun . . . . .	33
copyVisualStyle . . . . .	34
createColumnFilter . . . . .	34
createCompositeFilter . . . . .	36
createDegreeFilter . . . . .	37
createGraphFromNetwork . . . . .	38
createGroup . . . . .	39
createGroupByColumn . . . . .	40

createIgraphFromNetwork . . . . .	41
createNetworkFromDataFrames . . . . .	42
createNetworkFromGraph . . . . .	43
createNetworkFromIgraph . . . . .	44
createSubnetwork . . . . .	45
createVisualStyle . . . . .	46
cybrowserClose . . . . .	47
cybrowserDialog . . . . .	48
cybrowserHide . . . . .	49
cybrowserList . . . . .	50
cybrowserSend . . . . .	50
cybrowserShow . . . . .	51
cybrowserVersion . . . . .	52
cyrestAPI . . . . .	52
cyrestDELETE . . . . .	53
cyrestGET . . . . .	54
cyrestPOST . . . . .	54
cyrestPUT . . . . .	55
cytoscapeApiVersions . . . . .	56
cytoscapeFreeMemory . . . . .	57
cytoscapeMemoryStatus . . . . .	57
cytoscapeNumberOfCores . . . . .	58
cytoscapePing . . . . .	59
cytoscapeVersionInfo . . . . .	59
deleteAllNetworks . . . . .	60
deleteAnnotation . . . . .	61
deleteDuplicateEdges . . . . .	61
deleteGroup . . . . .	62
deleteNetwork . . . . .	63
deleteSelectedEdges . . . . .	64
deleteSelectedNodes . . . . .	64
deleteSelfLoops . . . . .	65
deleteStyleMapping . . . . .	66
deleteTableColumn . . . . .	66
deleteVisualStyle . . . . .	67
diffusionAdvanced . . . . .	68
diffusionBasic . . . . .	69
disableApp . . . . .	69
dockPanel . . . . .	70
doInitializeSandbox . . . . .	71
doRequestRemote . . . . .	71
doSetSandbox . . . . .	72
enableApp . . . . .	73
expandGroup . . . . .	73
exportFilters . . . . .	74
exportImage . . . . .	75
exportNetwork . . . . .	76
exportNetworkToNDEx . . . . .	77

exportVisualStyles . . . . .	78
findRemoteCytoscape . . . . .	79
fitContent . . . . .	79
floatPanel . . . . .	80
getAbsSandboxPath . . . . .	81
getAllEdges . . . . .	81
getAllNodes . . . . .	82
getAllStyleMappings . . . . .	83
getAnnotationList . . . . .	83
getAppInformation . . . . .	84
getAppStatus . . . . .	85
getAppUpdates . . . . .	85
getArrowShapes . . . . .	86
getAvailableApps . . . . .	86
getBackgroundColorDefault . . . . .	87
getBrowserClientChannel . . . . .	88
getBrowserClientJs . . . . .	88
getCollectionList . . . . .	89
getCollectionName . . . . .	89
getCollectionNetworks . . . . .	90
getCollectionSuid . . . . .	91
getCurrentSandbox . . . . .	91
getCurrentSandboxName . . . . .	92
getCurrentSandboxPath . . . . .	92
getCurrentStyle . . . . .	93
getDefaultSandbox . . . . .	93
getDefaultSandboxPath . . . . .	94
getDisabledApps . . . . .	94
getEdgeColor . . . . .	95
getEdgeCount . . . . .	95
getEdgeInfo . . . . .	96
getEdgeLineStyle . . . . .	97
getEdgeLineWidth . . . . .	97
getEdgeProperty . . . . .	98
getEdgeSelectionColorDefault . . . . .	99
getEdgeTargetArrowShape . . . . .	100
getFilterList . . . . .	100
getFirstNeighbors . . . . .	101
getGroupInfo . . . . .	102
getInstalledApps . . . . .	103
getJupyterBridgeURL . . . . .	103
getLayoutNameMapping . . . . .	104
getLayoutNames . . . . .	104
getLayoutPropertyNames . . . . .	105
getLayoutPropertyType . . . . .	106
getLayoutPropertyValue . . . . .	107
getLineStylees . . . . .	108
getNetworkCenter . . . . .	108

getNetworkCount	109
getNetworkList	110
getNetworkName	110
getNetworkNDExId	111
getNetworkProperty	112
getNetworkSuid	112
getNetworkViews	113
getNetworkViewSuid	114
getNetworkZoom	114
getNodeColor	115
getNodeCount	116
getNodeHeight	116
getNodePosition	117
getNodeProperty	118
getNodeSelectionColorDefault	119
getNodeShapes	119
getNodeSize	120
getNodeWidth	121
getNotebookIsRunning	121
getSandboxReinitialize	122
getSelectedEdgeCount	122
getSelectedEdges	123
getSelectedNodeCount	124
getSelectedNodes	124
getStyleDependencies	125
getStyleMapping	126
getTableColumnNames	127
getTableColumns	128
getTableColumnType	129
getTableValue	130
getUninstalledApps	131
getVisualPropertyDefault	131
getVisualPropertyNames	132
getVisualStyleNames	133
hideAllPanels	133
hideEdges	134
hideNodes	135
hidePanel	136
hideSelectedEdges	136
hideSelectedNodes	137
importFilters	138
importNetworkFromFile	139
importNetworkFromNDEx	139
importVisualStyles	140
installApp	141
invertEdgeSelection	141
invertNodeSelection	142
layoutCopycat	143

layoutNetwork . . . . .	144
listGroups . . . . .	145
loadTableData . . . . .	145
lockNodeDimensions . . . . .	146
makeSimpleGraph . . . . .	147
makeSimpleIgraph . . . . .	147
mapTableColumn . . . . .	148
mapVisualProperty . . . . .	149
matchArrowColorToEdge . . . . .	151
mergeNetworks . . . . .	152
openAppStore . . . . .	153
openSession . . . . .	154
paletteColorBrewerAccent . . . . .	155
paletteColorBrewerBlues . . . . .	155
paletteColorBrewerBrBG . . . . .	156
paletteColorBrewerBuGn . . . . .	156
paletteColorBrewerBuPu . . . . .	157
paletteColorBrewerDark2 . . . . .	157
paletteColorBrewerGnBu . . . . .	158
paletteColorBrewerGreens . . . . .	159
paletteColorBrewerGreys . . . . .	159
paletteColorBrewerOranges . . . . .	160
paletteColorBrewerOrRd . . . . .	160
paletteColorBrewerPaired . . . . .	161
paletteColorBrewerPastel1 . . . . .	161
paletteColorBrewerPastel2 . . . . .	162
paletteColorBrewerPiYG . . . . .	163
paletteColorBrewerPRGn . . . . .	163
paletteColorBrewerPuBu . . . . .	164
paletteColorBrewerPuBuGn . . . . .	164
paletteColorBrewerPuOr . . . . .	165
paletteColorBrewerPuRd . . . . .	165
paletteColorBrewerPurples . . . . .	166
paletteColorBrewerRdBu . . . . .	167
paletteColorBrewerRdPu . . . . .	167
paletteColorBrewerRdYIBu . . . . .	168
paletteColorBrewerReds . . . . .	168
paletteColorBrewerSet1 . . . . .	169
paletteColorBrewerSet2 . . . . .	169
paletteColorBrewerSet3 . . . . .	170
paletteColorBrewerYlGn . . . . .	171
paletteColorBrewerYlGnBu . . . . .	171
paletteColorBrewerYlOrBr . . . . .	172
paletteColorBrewerYlOrRd . . . . .	172
paletteColorRandom . . . . .	173
RCy3 . . . . .	173
RemoveFromGroup . . . . .	174
removeNodeCustomGraphics . . . . .	175

renameNetwork . . . . .	175
renameTableColumn . . . . .	176
resetDefaultSandbox . . . . .	177
runningRemoteCheck . . . . .	178
sandboxGetFileInfo . . . . .	178
sandboxGetFrom . . . . .	179
sandboxInitializer . . . . .	180
sandboxRemove . . . . .	181
sandboxRemoveFile . . . . .	181
sandboxSendTo . . . . .	182
sandboxSet . . . . .	183
sandboxUrlTo . . . . .	184
saveSession . . . . .	185
selectAllEdges . . . . .	186
selectAllNodes . . . . .	186
selectEdges . . . . .	187
selectEdgesAdjacentToSelectedNodes . . . . .	188
selectEdgesConnectingSelectedNodes . . . . .	189
selectFirstNeighbors . . . . .	190
selectNodes . . . . .	190
selectNodesConnectedBySelectedEdges . . . . .	191
setBackgroundDefault . . . . .	192
setCatchupFilterSecs . . . . .	193
setCatchupNetworkSecs . . . . .	193
setCurrentNetwork . . . . .	194
setCurrentSandbox . . . . .	195
setCurrentView . . . . .	195
setDefaultSandbox . . . . .	196
setDefaultSandboxPath . . . . .	196
setEdgeColorBypass . . . . .	197
setEdgeColorDefault . . . . .	198
setEdgeColorMapping . . . . .	199
setEdgeFontFaceBypass . . . . .	200
setEdgeFontFaceDefault . . . . .	201
setEdgeFontFaceMapping . . . . .	201
setEdgeFontSizeBypass . . . . .	203
setEdgeFontSizeDefault . . . . .	204
setEdgeFontSizeMapping . . . . .	204
setEdgeLabelBypass . . . . .	206
setEdgeLabelColorBypass . . . . .	207
setEdgeLabelColorDefault . . . . .	208
setEdgeLabelColorMapping . . . . .	208
setEdgeLabelDefault . . . . .	210
setEdgeLabelMapping . . . . .	210
setEdgeLabelOpacityBypass . . . . .	211
setEdgeLabelOpacityDefault . . . . .	212
setEdgeLabelOpacityMapping . . . . .	213
setEdgeLineStyleBypass . . . . .	214

setEdgeLineStyleDefault . . . . .	215
setEdgeLineStyleMapping . . . . .	216
setEdgeLineWidthBypass . . . . .	217
setEdgeLineWidthDefault . . . . .	218
setEdgeLineWidthMapping . . . . .	218
setEdgeOpacityBypass . . . . .	220
setEdgeOpacityDefault . . . . .	221
setEdgeOpacityMapping . . . . .	221
setEdgePropertyBypass . . . . .	223
setEdgeSelectionColorDefault . . . . .	224
setEdgeSourceArrowColorBypass . . . . .	225
setEdgeSourceArrowColorDefault . . . . .	226
setEdgeSourceArrowColorMapping . . . . .	226
setEdgeSourceArrowMapping . . . . .	228
setEdgeSourceArrowShapeBypass . . . . .	229
setEdgeSourceArrowShapeDefault . . . . .	230
setEdgeSourceArrowShapeMapping . . . . .	230
setEdgeTargetArrowColorBypass . . . . .	231
setEdgeTargetArrowColorDefault . . . . .	232
setEdgeTargetArrowColorMapping . . . . .	233
setEdgeTargetArrowMapping . . . . .	234
setEdgeTargetArrowShapeBypass . . . . .	235
setEdgeTargetArrowShapeDefault . . . . .	236
setEdgeTargetArrowShapeMapping . . . . .	237
setEdgeTooltipBypass . . . . .	238
setEdgeTooltipDefault . . . . .	239
setEdgeTooltipMapping . . . . .	240
setLayoutProperties . . . . .	241
setModelPropagationSecs . . . . .	242
setNetworkCenterBypass . . . . .	243
setNetworkPropertyBypass . . . . .	244
setNetworkZoomBypass . . . . .	245
setNodeBorderColorBypass . . . . .	246
setNodeBorderColorDefault . . . . .	247
setNodeBorderColorMapping . . . . .	247
setNodeBorderOpacityBypass . . . . .	249
setNodeBorderOpacityDefault . . . . .	250
setNodeBorderOpacityMapping . . . . .	250
setNodeBorderWidthBypass . . . . .	252
setNodeBorderWidthDefault . . . . .	253
setNodeBorderWidthMapping . . . . .	253
setNodeColorBypass . . . . .	255
setNodeColorDefault . . . . .	256
setNodeColorMapping . . . . .	256
setNodeComboOpacityMapping . . . . .	257
setNodeCustomBarChart . . . . .	258
setNodeCustomBoxChart . . . . .	260
setNodeCustomHeatMapChart . . . . .	261



setNodeCustomLinearGradient . . . . .	262
setNodeCustomLineChart . . . . .	263
setNodeCustomPieChart . . . . .	264
setNodeCustomPosition . . . . .	265
setNodeCustomRadialGradient . . . . .	266
setNodeCustomRingChart . . . . .	267
setNodeFillOpacityBypass . . . . .	268
setNodeFillOpacityDefault . . . . .	270
setNodeFillOpacityMapping . . . . .	270
setNodeFontFaceBypass . . . . .	272
setNodeFontFaceDefault . . . . .	273
setNodeFontFaceMapping . . . . .	273
setNodeFontSizeBypass . . . . .	275
setNodeFontSizeDefault . . . . .	276
setNodeFontSizeMapping . . . . .	276
setNodeHeightBypass . . . . .	278
setNodeHeightDefault . . . . .	279
setNodeHeightMapping . . . . .	279
setNodeLabelBypass . . . . .	280
setNodeLabelColorBypass . . . . .	281
setNodeLabelColorDefault . . . . .	282
setNodeLabelColorMapping . . . . .	283
setNodeLabelDefault . . . . .	284
setNodeLabelMapping . . . . .	285
setNodeLabelOpacityBypass . . . . .	285
setNodeLabelOpacityDefault . . . . .	286
setNodeLabelOpacityMapping . . . . .	287
setNodeOpacityBypass . . . . .	288
setNodePropertyBypass . . . . .	289
setNodeSelectionColorDefault . . . . .	290
setNodeShapeBypass . . . . .	291
setNodeShapeDefault . . . . .	292
setNodeShapeMapping . . . . .	293
setNodeSizeBypass . . . . .	294
setNodeSizeDefault . . . . .	295
setNodeSizeMapping . . . . .	295
setNodeTooltipBypass . . . . .	296
setNodeTooltipDefault . . . . .	297
setNodeTooltipMapping . . . . .	298
setNodeWidthBypass . . . . .	299
setNodeWidthDefault . . . . .	300
setNodeWidthMapping . . . . .	300
setNotebookIsRunning . . . . .	302
setSandboxReinitialize . . . . .	302
setStyleDependencies . . . . .	303
setVisualPropertyDefault . . . . .	304
setVisualStyle . . . . .	304
spoofResponse-class . . . . .	305

syncNodeCustomGraphicsSize . . . . .	305
toggleGraphicsDetails . . . . .	306
unhideAll . . . . .	307
unhideEdges . . . . .	308
unhideNodes . . . . .	309
uninstallApp . . . . .	310
updateApp . . . . .	310
updateNetworkInNDEx . . . . .	311
updateStyleDefaults . . . . .	312
updateStyleMapping . . . . .	312

## **Index** **314**

---

*.getDefaultSandbox*      *.getDefaultSandbox*

---

### **Description**

*.getDefaultSandbox*

### **Usage**

*.getDefaultSandbox()*

### **Value**

None

### **Examples**

*.getDefaultSandbox()*

---

*.getRequester*      *.getRequester*

---

### **Description**

*.getRequester*

### **Usage**

*.getRequester()*

### **Value**

None

**Examples**

```
.getRequester()
```

---

addAnnotationText	<i>Add Text Annotation</i>
-------------------	----------------------------

---

**Description**

Adds a text annotation to a Cytoscape network view. The object will also be added to the Annotation Panel in the GUI.

**Usage**

```
addAnnotationText(
  text = NULL,
  x.pos = NULL,
  y.pos = NULL,
  fontSize = NULL,
  fontFamily = NULL,
  fontStyle = NULL,
  color = NULL,
  angle = NULL,
  name = NULL,
  canvas = NULL,
  z.order = NULL,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

text	The text to be displayed
x.pos	(optional) X position in pixels from left; default is center of current view
y.pos	(optional) Y position in pixels from top; default is center of current view
fontSize	(optional) Numeric value; default is 12
fontFamily	(optional) Font family; default is Arial
fontStyle	(optional) Font style; default is
color	(optional) Hexidecimal color; default is #000000 (black)
angle	(optional) Angle of text orientation; default is 0.0 (horizontal)
name	(optional) Name of annotation object; default is "Text"
canvas	(optional) Canvas to display annotation, i.e., foreground (default) or background
z.order	(optional) Arrangement order specified by number (larger values are in front of smaller values); default is 0

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A named list of annotation properties, including UUID

**Examples**

```
addAnnotationText("test1")
addAnnotationText("test2", 1000, 1000, name="T2")
addAnnotationText("test3", 1200, 1000, 30, "Helvetica", "bold", "#990000",
  40, name="T3", canvas="foreground", z=4)
```

---

 addCyEdges

*Add CyEdges*


---

**Description**

Add one or more edges to a Cytoscape network by listing source and target node pairs.

**Usage**

```
addCyEdges(
  source.target.list,
  edgeType = "interacts with",
  directed = FALSE,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

source.target.list	A list (or list of lists) of source and target node name or SUID pairs
edgeType	The type of interaction. Default is 'interacts with'.
directed	boolean for whether interactions are directed. Default is FALSE.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A list of named lists of SUID, source and target for each edge added.

**Examples**

```
addCyEdges(c('sourceNode', 'targetNode'))
addCyEdges(list(c('s1', 't1'), c('s2', 't2')))
```

---

 addCyNodes

*Add CyNodes*


---

**Description**

Add one or more nodes to a Cytoscape network.

**Usage**

```
addCyNodes(
  node.names,
  skip.duplicate.names = TRUE,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

node.names	A list of node names
skip.duplicate.names	Skip adding a node if a node with the same name is already in the network. If FALSE then a duplicate node (with a unique SUID) will be added. Default is TRUE.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A list of named lists of name and SUID for each node added.

**Examples**

```
addCyNodes(c('Node A', 'Node B', 'Node C'))
```

---

 AddToGroup

*Add to Group*


---

### Description

Adds the specified nodes and edges to the specified group.

### Usage

```
AddToGroup(
  group.name,
  nodes = NULL,
  nodes.by.col = "SUID",
  edges = NULL,
  edges.by.col = "SUID",
  network = NULL,
  base.url = .defaultBaseUrl
)
```

### Arguments

<code>group.name</code>	Specifies the name used to identify the group
<code>nodes</code>	List of node SUIDs, names, other column values, or keyword: selected, unselected or all. Default is currently selected nodes.
<code>nodes.by.col</code>	name of node table column corresponding to provided nodes list. Default is 'SUID'.
<code>edges</code>	List of edge SUIDs, names, other column values, or keyword: selected, unselected or all. Default is currently selected edges.
<code>edges.by.col</code>	name of edge table column corresponding to provided edges list. Default is 'SUID'.
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Value

None

### Examples

```
AddToGroup('myGroup')
```

---

analyzeNetwork	<i>Analyze Network</i>
----------------	------------------------

---

**Description**

Calculate various network statistics.

**Usage**

```
analyzeNetwork(directed = FALSE, base.url = .defaultBaseUrl)
```

**Arguments**

`directed` (optional) If TRUE, the network is considered a directed graph. Default is FALSE.

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Details**

The results are added to the Node and Edge tables and the Results Panel. The summary statistics in the Results Panel are also returned by the function as a list of named values.

**Value**

Named list of summary statistics

**Examples**

```
analyzeNetwork()  
analyzeNetwork(TRUE)
```

---

applyFilter	<i>Apply Filter</i>
-------------	---------------------

---

**Description**

Run an existing filter by supplying the filter name.

## Usage

```
applyFilter(  
  filter.name = "Default filter",  
  hide = FALSE,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

## Arguments

<code>filter.name</code>	Name of filter to apply. Default is "Default filter".
<code>hide</code>	Whether to hide filtered out nodes and edges. Default is FALSE. Ignored if all nodes or edges are filtered out. This is an alternative to filtering for node and edge selection.
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

## Details

Known bug: selection (or hiding) of edges using edge-based column filters does not work. As a workaround, simply repeat the `createColumnFilter` operation to perform selection (or hiding) of edges.

## Value

List of selected nodes and edges.

## See Also

`unhideAll`

## Examples

```
applyFilter('myFilter')  
applyFilter('myFilter', hide = TRUE)
```



---

`bundleEdges`*Bundle Edges*

---

**Description**

Apply edge bundling to the network specified. Edge bundling is executed with default parameters; optional parameters are not supported.

**Usage**

```
bundleEdges(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
bundleEdges()
```

---

`checkNotebookIsRunning`*checkNotebookIsRunning*

---

**Description**

`checkNotebookIsRunning`

**Usage**

```
checkNotebookIsRunning()
```

**Value**

None

**Examples**

```
checkNotebookIsRunning()
```

---

```
checkRunningRemote      checkRunningRemote
```

---

**Description**

Determine whether we're running locally or on a remote server. If locally (either via raw R or via a locally installed Notebook), we prefer to connect to Cytoscape over a local socket. If remote, we have to connect over Jupyter-Bridge. Either way, we can determine which by whether Cytoscape answers to a version check. If Cytoscape doesn't answer, we have no information ... and we have to wait until Cytoscape is started and becomes reachable before we can determine local vs remote.

**Usage**

```
checkRunningRemote()
```

**Value**

None

**Examples**

```
checkRunningRemote()
```

---

```
clearEdgeBends          Clear Edge Bends
```

---

**Description**

Clear all edge bends, e.g., those created from edge bundling.

**Usage**

```
clearEdgeBends(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
clearEdgeBends()
```

---

```
clearEdgePropertyBypass
```

*Clear Edge Property Bypass*

---

**Description**

Clear bypass values for any edge property of the specified edges, effectively restoring any previously defined style defaults or mappings.

**Usage**

```
clearEdgePropertyBypass(  
  edge.names,  
  visual.property,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

<code>edge.names</code>	List of edge names or SUIDs. Pending CyREST updates, if set to 'all', then the property is cleared for all edges.
<code>visual.property</code>	Name of a visual property. See <a href="#">getVisualPropertyNames</a> .
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**See Also**

[setEdgePropertyBypass](#)

**Examples**

```
clearEdgePropertyBypass()
```

---

```
clearNetworkCenterBypass
```

*Clear Network Center Bypass*

---

**Description**

Clear the bypass value for center x and y for the network, effectively restoring prior default values.

**Usage**

```
clearNetworkCenterBypass(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
clearNetworkCenterBypass()
```

---

```
clearNetworkPropertyBypass
```

*Clear Network Property Bypass*

---

**Description**

Clear bypass values for any network property, effectively restoring any previously defined style defaults or mappings.

**Usage**

```
clearNetworkPropertyBypass(
    visual.property,
    network = NULL,
    base.url = .defaultBaseUrl
)
```

**Arguments**

visual.property	Name of a visual property. See <a href="#">getVisualPropertyNames</a> .
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**See Also**

[setNodePropertyBypass](#)

**Examples**

```
clearNetworkPropertyBypass()
```

---

```
clearNetworkZoomBypass
```

*Clear Network Zoom Bypass*

---

**Description**

Clear the bypass value for the scale factor for the network, effectively restoring prior default values.

**Usage**

```
clearNetworkZoomBypass(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
clearNetworkZoomBypass()
```

---

```
clearNodeOpacityBypass
```

*Clear Node Opacity Bypass*

---

**Description**

Clear the bypass value for node fill, label and border opacity for the specified node or nodes, effectively restoring any previously defined style defaults or mappings.

**Usage**

```
clearNodeOpacityBypass(node.names, network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>node.names</code>	List of node names or SUIDs
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**See Also**

[setNodeOpacityBypass](#)

**Examples**

```
clearNodeOpacityBypass(c('Node 1', 'Node 2'))
```

---

`clearNodePropertyBypass`*Clear Node Property Bypass*

---

### Description

Clear bypass values for any node property of the specified nodes, effectively restoring any previously defined style defaults or mappings.

### Usage

```
clearNodePropertyBypass(  
  node.names,  
  visual.property,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

<code>node.names</code>	List of node names or SUIDs. Pending CyREST updates, if set to 'all', then the property is cleared for all nodes.
<code>visual.property</code>	Name of a visual property. See <a href="#">getVisualPropertyNames</a> .
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Value

None

### See Also

[setNodePropertyBypass](#)

### Examples

```
clearNodePropertyBypass()
```

---

clearSelection	<i>Clear Selection</i>
----------------	------------------------

---

**Description**

If any nodes are selected in the network, they will be unselected.

**Usage**

```
clearSelection(type = "both", network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

type	'nodes', 'edges' or 'both' (default)
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Author(s)**

AlexanderPico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
clearSelection()
```

---

cloneNetwork	<i>Clone a Cytoscape Network</i>
--------------	----------------------------------

---

**Description**

Makes a copy of a Cytoscape Network with all of its edges and nodes.

**Usage**

```
cloneNetwork(network = NULL, base.url = .defaultBaseUrl)
```



**Arguments**

network	(optional) Name or SUID of the network you want to clone; default is "current" network
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

The suid of the new network

**Author(s)**

Alexander Pico, Julia Gustavsen

**Examples**

```
cloneNetwork("cloned network")
```

---

closeSession	<i>Close Session</i>
--------------	----------------------

---

**Description**

Closes the current session in Cytoscape, destroying all unsaved work.

**Usage**

```
closeSession(save.before.closing, filename = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

save.before.closing	boolean Whether to save before closing the current session. If FALSE, then all unsaved work will be lost.
filename	(optional) If save.before.closing is TRUE and the session has not previously been saved, then the path and name of the session file to save should be provided. Default is NULL.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

A boolean for whether to save before closing is required since you could lose data by closing without saving.

**Value**

None

**Examples**

```
closeSession(FALSE)
closeSession(TRUE, '/fullpath/mySession')
closeSession(TRUE)
```

---

collapseGroup	<i>Collapse Group</i>
---------------	-----------------------

---

**Description**

Replaces the representation of all of the nodes and edges in a group with a single node.

**Usage**

```
collapseGroup(groups = NULL, network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

groups	(optional) List of group names or keywords: all, selected, unselected. Default is the currently selected group.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
collapseGroup()
```

---

commandEcho	<i>Command Echo</i>
-------------	---------------------

---

**Description**

The echo command will display the value of the variable specified by the variableName argument, or all variables if variableName is not provided.

**Usage**

```
commandEcho(variable.name = "*", base.url = .defaultBaseUrl)
```

**Arguments**

variable.name	(optional) The name of the variable to display. Default is to display all variable values using "*".
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Value of variable

**Examples**

```
commandEcho()
```

---

commandOpenDialog	<i>Command Open Dialog</i>
-------------------	----------------------------

---

**Description**

The command line dialog provides a field to enter commands and view results. It also provides the help command to display namespaces, commands, and arguments

**Usage**

```
commandOpenDialog(base.url = .defaultBaseUrl)
```

**Arguments**

base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.
----------	--

**Value**

None

**Examples**

```
commandOpenDialog()
```

---

`commandPause`*Command Pause*

---

**Description**

The pause command displays a dialog with the text provided in the message argument and waits for the user to click OK.

**Usage**

```
commandPause(message = "", base.url = .defaultBaseUrl)
```

**Arguments**

<code>message</code>	(optional) Text to display in pause dialog
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
commandPause('Please click OK to continue.')
```

---

commandQuit	<i>Command Quit</i>
-------------	---------------------

---

**Description**

This command causes Cytoscape to exit. It is typically used at the end of a script file

**Usage**

```
commandQuit(base.url = .defaultBaseUrl)
```

**Arguments**

base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.
----------	--

**Value**

None

**Examples**

```
commandQuit()
```

---

commandRunFile	<i>Command Run File</i>
----------------	-------------------------

---

**Description**

The run command will execute a command script from the file pointed to by the file argument, which should contain Cytoscape commands, one per line. Arguments to the script are provided by the args argument

**Usage**

```
commandRunFile(file, args = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

file	Path to command script file
args	The script arguments as key:value pairs separated by commas
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
commandRunFile('/path/to/my/file.txt')
```

---

 commandsAPI

*Open Swagger docs for CyREST Commands API*


---

**Description**

Opens swagger docs in default browser for a live instance of Commands available via CyREST.

**Usage**

```
commandsAPI(base.url = .defaultBaseUrl)
```

**Arguments**

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Web page in browser

**Examples**

```
commandsAPI()
```

---

 commandsGET

*Commands GET*


---

**Description**

Using the same syntax as Cytoscape's Command Line Dialog, this function converts a command string into a CyREST query URL, executes a GET request, and parses the result content into an R list object. The function check whether actual call is local or remote first. If remote, requests will go through Jupyter-Bridge.

**Usage**

```
commandsGET(cmd.string, base.url = .defaultBaseUrl)
```

**Arguments**

cmd.string (char) command  
 base.url (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A list, status or None.

**Examples**

```
commandsGET('layout get preferred network=current')
commandsGET('network list properties network=current')
commandsGET('layout force-directed defaultNodeMass=1')
```

---

commandsHelp *Commands Help*

---

**Description**

Using the same syntax as Cytoscape’s Command Line Dialog, this function returns a list of available commands or args. The function check whether actual call is local or remote first. If remote, requests will go through Jupyter-Bridge.

**Usage**

```
commandsHelp(cmd.string = "help", base.url = .defaultBaseUrl)
```

**Arguments**

cmd.string (char) command  
 base.url (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Works with or without 'help' command prefix. Note that if you ask about a command that doesn't have any arguments, this function will run the command!

**Value**

List of available commands or args

**Examples**

```

commandsHelp()
commandsHelp('node')
commandsHelp('node get attribute')

```

---

commandSleep	<i>Command Sleep</i>
--------------	----------------------

---

**Description**

The sleep command will pause processing for a period of time as specified by duration seconds. It is typically used as part of a command script.

**Usage**

```
commandSleep(duration = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

duration	(optional) The time in seconds to sleep
base.url	(optional) Ignore unless you need to specify a custom domain, #' port or version to connect to the CyREST API. Default is http://localhost:1234 #' and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
commandSleep(5)
```

---

commandsPOST	<i>Commands POST</i>
--------------	----------------------

---

**Description**

Using the same syntax as Cytoscape's Command Line Dialog, this function converts a command string into a CyREST query URL, executes a POST request, and parses the result content into an R list object. The function check whether actual call is local or remote first. If remote, requests will go through Jupyter-Bridge.

**Usage**

```
commandsPOST(cmd.string, base.url = .defaultBaseUrl)
```



**Arguments**

cmd.string (char) command  
 base.url (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A list, named list, status or None.

**Examples**

```
commandsPOST('layout get preferred')
commandsPOST('network list properties')
commandsPOST('layout force-directed defaultNodeMass=1')
```

---

commandsRun	<i>Run a Command</i>
-------------	----------------------

---

**Description**

Using the same syntax as Cytoscape’s Command Line Dialog, this function converts a command string into a CyREST query URL, executes a GET request, and parses the result content into an R list object. Same as commandsGET.

**Usage**

```
commandsRun(cmd.string, base.url = .defaultBaseUrl)
```

**Arguments**

cmd.string (char) command  
 base.url (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A list, status or None.

**Examples**

```
commandsRun('layout get preferred')
commandsRun('network list properties')
commandsRun('layout force-directed defaultNodeMass=1')
```

---

copyVisualStyle	<i>Copy Visual Style</i>
-----------------	--------------------------

---

**Description**

Create a new visual style by copying a specified style.

**Usage**

```
copyVisualStyle(from.style, to.style, base.url = .defaultBaseUrl)
```

**Arguments**

from.style	Name of visual style to copy
to.style	Name of new visual style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
copyVisualStyle()
```

---

createColumnFilter	<i>Create Column Filter</i>
--------------------	-----------------------------

---

**Description**

Creates a filter to control node or edge selection. Works on columns of boolean, string, numeric and lists. Note the unique restrictions for criterion and predicate depending on the type of column being filtered.

**Usage**

```
createColumnFilter(  
  filter.name,  
  column,  
  criterion,  
  predicate,  
  caseSensitive = FALSE,
```

```

    anyMatch = TRUE,
    type = "nodes",
    hide = FALSE,
    network = NULL,
    base.url = .defaultBaseUrl,
    apply = TRUE
)

```

### Arguments

<code>filter.name</code>	Name for filter.
<code>column</code>	Table column to base filter upon.
<code>criterion</code>	For boolean columns: TRUE or FALSE. For string columns: a string value, e.g., "hello". If the predicate is REGEX then this can be a regular expression as accepted by the Java Pattern class ( <a href="https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html">https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html</a> ). For numeric columns: If the predicate is BETWEEN or IS_NOT_BETWEEN then this is a two-element vector of numbers, example: c(1,5), otherwise a single number.
<code>predicate</code>	For boolean columns: IS, IS_NOT. For string columns: IS, IS_NOT, CONTAINS, DOES_NOT_CONTAIN, REGEX. For numeric columns: IS, IS_NOT, GREATER_THAN, GREATER_THAN_OR_EQUAL, LESS_THAN, LESS_THAN_OR_EQUAL, BETWEEN, IS_NOT_BETWEEN
<code>caseSensitive</code>	(optional) If string matching should be case sensitive. Default is FALSE.
<code>anyMatch</code>	(optional) Only applies to List columns. If true then at least one element in the list must pass the filter, if false then all the elements in the list must pass the filter. Default is TRUE.
<code>type</code>	(optional) Apply filter to "nodes" (default) or "edges".
<code>hide</code>	Whether to hide filtered out nodes and edges. Default is FALSE. Ignored if all nodes or edges are filtered out. This is an alternative to filtering for node and edge selection.
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.
<code>apply</code>	(bool) True to execute filter immediately (default); False to define filter but not execute it (available in Cytoscape 3.9+).

### Value

List of selected nodes and edges.

### Examples

```

createColumnFilter('myFilter', 'log2FC', c(-1,1), "IS_NOT_BETWEEN")
createColumnFilter('myFilter', 'pValue', 0.05, "LESS_THAN")

```

```

createColumnFilter('myFilter', 'function', "kinase", "CONTAINS", FALSE)
createColumnFilter('myFilter', 'name', "^Y.*C$", "REGEX")
createColumnFilter('myFilter', 'isTarget', TRUE, "IS", apply=FALSE)
createColumnFilter('myFilter', 'isTarget', TRUE, "IS", hide=TRUE)

```

---

## createCompositeFilter *Create Composite Filter*

---

### Description

Combines filters to control node and edge selection based on previously created filters.

### Usage

```

createCompositeFilter(
  filter.name,
  filter.list,
  type = "ALL",
  hide = FALSE,
  network = NULL,
  base.url = .defaultBaseUrl,
  apply = TRUE
)

```

### Arguments

<code>filter.name</code>	Name for filter.
<code>filter.list</code>	List of filters to combine.
<code>type</code>	(optional) Type of composition, requiring ALL (default) or ANY filters to pass for final node and edge selection.
<code>hide</code>	Whether to hide filtered out nodes and edges. Default is FALSE. Ignored if all nodes or edges are filtered out. This is an alternative to filtering for node and edge selection.
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.
<code>apply</code>	(bool) True to execute filter immediately (default); False to define filter but not execute it (available in Cytoscape 3.9+).

### Value

List of selected nodes and edges.

**Examples**

```
createCompositeFilter("comp1", c("filter1", "filter2"))
createCompositeFilter("comp2", c("filter1", "filter2"), "ANY")
createCompositeFilter("comp3", c("comp1", "filter3"), apply=FALSE)
```

---

createDegreeFilter      *Create Degree Filter*

---

**Description**

Creates a filter to control node selection base on in/out degree.

**Usage**

```
createDegreeFilter(
  filter.name,
  criterion,
  predicate = "BETWEEN",
  edgeType = "ANY",
  hide = FALSE,
  network = NULL,
  base.url = .defaultBaseUrl,
  apply = TRUE
)
```

**Arguments**

filter.name	Name for filter.
criterion	A two-element vector of numbers, example: c(1,5).
predicate	BETWEEN (default) or IS_NOT_BETWEEN
edgeType	(optional) Type of edges to consider in degree count: ANY (default), UNDIRECTED, INCOMING, OUTGOING, DIRECTED
hide	Whether to hide filtered out nodes and edges. Default is FALSE. Ignored if all nodes or edges are filtered out. This is an alternative to filtering for node and edge selection.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.
apply	(bool) True to execute filter immediately (default); False to define filter but not execute it (available in Cytoscape 3.9+).

**Value**

List of selected nodes and edges.

**Examples**

```
createDegreeFilter('myFilter', c(4,5))
createDegreeFilter('myFilter', c(2,5), apply=FALSE)
```

---

createGraphFromNetwork

*createGraphFromNetwork*

---

**Description**

Returns the Cytoscape network as a Bioconductor graph.

**Usage**

```
createGraphFromNetwork(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A Bioconductor graph object.

**Author(s)**

Alexander Pico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
cw <- CytoscapeWindow('network', graph=make_graphnel())
g <- createGraphFromNetwork()
g <- createGraphFromNetwork('myNetwork')
```

---

createGroup	<i>Create Group</i>
-------------	---------------------

---

## Description

Create a group from the specified nodes.

## Usage

```
createGroup(  
  group.name,  
  nodes = NULL,  
  nodes.by.col = "SUID",  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

## Arguments

group.name	The name used to identify and optionally label the group
nodes	List of node SUIDs, names, other column values, or keyword: selected, unselected or all. Default is currently selected nodes.
nodes.by.col	name of node table column corresponding to provided nodes list. Default is 'SUID'.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

## Value

Group SUID

## Examples

```
createGroup('myGroup')
```

---

createGroupByColumn    *Create Group by Column*

---

### Description

Create a group of nodes defined by a column value.

### Usage

```
createGroupByColumn(  
  group.name,  
  column = NULL,  
  value = NULL,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

group.name	The name used to identify and optionally label the group
column	The name or header of the Node Table column to use for selecting nodes to group
value	The value in the column to use for selecting nodes to group
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

### Value

Group SUID

### Examples

```
createGroupByColumn('myGroup', 'Cluster', 'A')
```



---

`createIgraphFromNetwork`*Create an igraph network from a Cytoscape network*

---

**Description**

Takes a Cytoscape network and generates data frames for vertices and edges to send to the `graph_from_data_frame` function. Returns the `network.suid` and applies the preferred layout set in Cytoscape preferences.

**Usage**

```
createIgraphFromNetwork(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Nodes and edges from the Cytoscape network will be translated into vertices and edges in igraph. Associated table columns will also be passed to igraph as vertex and edge attributes. Note: all networks are implicitly modeled as directed in Cytoscape. Round-trip conversion of an undirected network in igraph via `createNetworkFromIgraph` to Cytoscape and back to igraph will result in a directed network.

**Value**

(igraph) an igraph network

**See Also**

`createNetworkFromDataFrames`, `createNetworkFromIgraph`

**Examples**

```
ig <- createIgraphFromNetwork()  
ig <- createIgraphFromNetwork('myNetwork')
```

---

```
createNetworkFromDataFrames
```

*Create a network from data frames*

---

### Description

Takes data frames for nodes and edges, as well as naming parameters to generate the JSON data format required by the "networks" POST operation via CyREST. Returns the network.suid and applies the preferred layout set in Cytoscape preferences.

### Usage

```
createNetworkFromDataFrames(
  nodes = NULL,
  edges = NULL,
  title = "From dataframe",
  collection = "My Dataframe Network Collection",
  base.url = .defaultBaseUrl,
  ...
)
```

### Arguments

nodes	(data.frame) see details and examples below; default NULL to derive nodes from edge sources and targets
edges	(data.frame) see details and examples below; default NULL for disconnected set of nodes
title	(char) network name
collection	(char) network collection name
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.
...	params for nodeSet2JSON() and edgeSet2JSON()

### Details

NODES should contain a column of character strings named: id. This name can be overridden by the arg: node.id.list. Additional columns are loaded as node attributes. EDGES should contain columns of character strings named: source, target and interaction. These names can be overridden by args: source.id.list, target.id.list, interaction.type.list. Additional columns are loaded as edge attributes. The 'interaction' list can contain a single value to apply to all rows; and if excluded altogether, the interaction type will be set to "interacts with". NOTE: attribute values of types (num) will be imported as (Double); (int) as (Integer); (chr) as (String); and (logical) as (Boolean). (Lists) will be imported as (Lists) in CyREST v3.9+.

**Value**

(int) network SUID

**Examples**

```

nodes <- data.frame(id=c("node 0","node 1","node 2","node 3"),
  group=c("A","A","B","B"), # categorical strings
  score=as.integer(c(20,10,15,5))) # integers
edges <- data.frame(source=c("node 0","node 0","node 0","node 2"),
  target=c("node 1","node 2","node 3","node 3"),
  interaction=c("inhibits","interacts",
    "activates","interacts"), # optional
  weight=c(5.1,3.0,5.2,9.9)) # numeric

createNetworkFromDataFrames(nodes,edges)

```

---

```
createNetworkFromGraph
```

*Create Network From Graph*

---

**Description**

Creates a Cytoscape network from a Bioconductor graph.

**Usage**

```

createNetworkFromGraph(
  graph,
  title = "From graph",
  collection = "My GraphNEL Network Collection",
  base.url = .defaultBaseUrl
)

```

**Arguments**

graph	A GraphNEL object
title	(char) network name
collection	(char) network collection name
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Network SUID

**Author(s)**

Alexander Pico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
library(igraph)
g <- makeSimpleGraph()
createNetworkFromGraph(g)
```

---

createNetworkFromIgraph

*Create a Cytoscape network from an igraph network*

---

**Description**

Takes an igraph network and generates data frames for nodes and edges to send to the createNetwork function. Returns the network.suid and applies the preferred layout set in Cytoscape preferences.

**Usage**

```
createNetworkFromIgraph(  
  igraph,  
  title = "From igraph",  
  collection = "My Igraph Network Collection",  
  base.url = .defaultBaseUrl,  
  ...  
)
```

**Arguments**

igraph	(igraph) igraph network object
title	(char) network name
collection	(char) network collection name
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.
...	params for nodeSet2JSON() and edgeSet2JSON(); see createNetwork

**Details**

Vertices and edges from the igraph network will be translated into nodes and edges in Cytoscape. Associated attributes will also be passed to Cytoscape as node and edge table columns. Note: undirected networks will be implicitly modeled as directed in Cytoscape. Conversion back via createIgraphFromNetwork will result in a directed network. Also note: igraph attributes of type "other" denoted by "x" are converted to "String" in Cytoscape.

**Value**

(int) network SUID

**See Also**

createNetworkFromDataFrames, createIgraphFromNetwork

**Examples**

```
library(igraph)
ig <- makeSimpleIgraph()
createNetworkFromIgraph(ig)
```

---

createSubnetwork	<i>Create subnetwork from existing network</i>
------------------	--

---

**Description**

Copies a subset of nodes and edges into a newly created subnetwork.

**Usage**

```
createSubnetwork(
  nodes = NULL,
  nodes.by.col = "SUID",
  edges = NULL,
  edges.by.col = "SUID",
  exclude.edges = FALSE,
  subnetwork.name = NULL,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

nodes	list of nodes by SUID, by specified nodes.by.col value (e.g., name) or by key-word: selected, unselected or all. Default is currently selected nodes.
nodes.by.col	name of node table column corresponding to provided nodes list; default is 'SUID'
edges	list of edges by SUID, by specified nodes.by.col value (e.g., name) or by key-word: selected, unselected or all. Default is currently selected edges.
edges.by.col	name of edge table column corresponding to provided edges list; default is 'SUID'
exclude.edges	(boolean) whether to exclude connecting edges; default is FALSE

subnetwork.name	name of new subnetwork to be created; default is to add a numbered suffix to source network name
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

### Details

If you specify both nodes and edges, the resulting subset will be the union of those sets. Typical usage only requires specifying either nodes or edges. Note that selected nodes will bring along their connecting edges by default (see `exclude.edges` arg) and selected edges will always bring along their source and target nodes.

### Value

SUID of new subnetwork

### Examples

```
createSubnetwork()
createSubnetwork("all")
createSubnetwork(subnetwork.name="mySubnetwork")
createSubnetwork(c("node 1", "node 2", "node 3"), "name")
createSubnetwork(c("AKT1", "TP53", "PIK3CA"), "display name")
createSubnetwork(edges="all") #subnetwork of all connected nodes
```

---

createVisualStyle      *Create a visual style from components*

---

### Description

Create a style from defaults and predefined mappings.

### Usage

```
createVisualStyle(style.name, defaults, mappings, base.url = .defaultBaseUrl)
```

### Arguments

style.name	(char) name for style
defaults	(list) key-value pairs for default mappings.
mappings	(list) visual property mappings, see <code>mapVisualStyleProperty</code>
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Requires attribute mappings to be previously created, see `mapVisualProperty`.

**Value**

None

**See Also**

`applyStyle`, `mapVisualProperty`

**Examples**

```
#first there has to be a network to apply style to
example(createNetworkFromDataFrames)

#then prepare style variables
style.name = "myStyle"
defaults <- list(NODE_SHAPE="diamond",
                NODE_SIZE=30,
                EDGE_TRANSPARENCY=120,
                NODE_LABEL_POSITION="W,E,c,0.00,0.00")
nodeLabels <- mapVisualProperty('node label','id','p')
nodeFills <- mapVisualProperty('node fill color','group','d',c("A","B"), c("#FF9900","#66AAAA"))
arrowShapes <- mapVisualProperty('Edge Target Arrow Shape','interaction','d',
                                 c("activates","inhibits","interacts"),c("Arrow","T","None"))
edgeWidth <- mapVisualProperty('edge width','weight','p')

#and then create the style
createVisualStyle(style.name, defaults, list(nodeLabels,nodeFills,arrowShapes,edgeWidth))

#finsh by applying the style
setVisualStyle(style.name)
```

---

cybrowserClose

*Cybrowser Close*

---

**Description**

Close an internal web browser and remove all content. Provide an id for the browser you want to close.

**Usage**

```
cybrowserClose(id = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

id	(optional) The identifier for the browser window to close
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
cybrowserClose('554')
```

---

cybrowserDialog	<i>Cybrowser Dialog</i>
-----------------	-------------------------

---

**Description**

Launch Cytoscape's internal web browser in a separate window. Provide an id for the window if you want subsequent control of the window e.g., via cybrowser hide.

**Usage**

```
cybrowserDialog(
  id = NULL,
  text = NULL,
  title = NULL,
  url = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

id	(optional) The identifier for the browser window
text	(optional) HTML text to initially load into the browser
title	(optional) Text to be shown in the title bar of the browser window
url	(optional) The URL the browser should load
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None



**See Also**[cybrowserShow](#)[cybrowserHide](#)**Examples**

```
cybrowserDialog(url='http://cytoscape.org')
```

---

cybrowserHide

*Cybrowser Hide*

---

**Description**

Hide an existing browser, whether it's in the Results panel or a separate window.

**Usage**

```
cybrowserHide(id = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>id</code>	(optional) The identifier for the browser window to hide
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**See Also**[cybrowserShow](#)[cybrowserDialog](#)**Examples**

```
cybrowserHide()
```

---

cybrowserList	<i>Cybrowser List</i>
---------------	-----------------------

---

**Description**

List all browsers that are currently open, whether as a dialog or in the results panel.

**Usage**

```
cybrowserList(base.url = .defaultBaseUrl)
```

**Arguments**

base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.
----------	--

**Value**

List of open cybrowser windows

**Examples**

```
cybrowserList()
```

---

cybrowserSend	<i>Cybrowser Send</i>
---------------	-----------------------

---

**Description**

Send the text to the browser indicated by the id and return the response, if any. Note that the JSON result field could either be a bare string or JSON formatted text.

**Usage**

```
cybrowserSend(id = NULL, script, base.url = .defaultBaseUrl)
```

**Arguments**

id	(optional) The identifier for the browser window
script	(optional) A string that represents a JavaScript variable, script, or call to be executed in the browser. Note that only string results are returned.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

String result

**Examples**

```
cybrowserSend(id="Window 1", script="navigator.userAgent;")
```

---

cybrowserShow

*Cybrowser Show*

---

**Description**

Launch Cytoscape's internal web browser in a pane in the Result Panel. Provide an id for the window if you want subsequent control of the window via cybrowser hide.

**Usage**

```
cybrowserShow(  
  id = NULL,  
  text = NULL,  
  title = NULL,  
  url = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

id	(optional) The identifier for the browser window
text	(optional) HTML text to initially load into the browser
title	(optional) Text to be shown in the title bar of the browser window
url	(optional) The URL the browser should load
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**See Also**

[cybrowserDialog](#)

[cybrowserHide](#)

**Examples**

```
cybrowserShow(url='http://cytoscape.org')
```

---

cybrowserVersion	<i>Cybrowser Version</i>
------------------	--------------------------

---

**Description**

Display the version of the CyBrowser app.

**Usage**

```
cybrowserVersion(base.url = .defaultBaseUrl)
```

**Arguments**

base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.
----------	--

**Value**

Version number

**Examples**

```
cybrowserVersion()
```

---

cyrestAPI	<i>Open Swagger docs for CyREST API</i>
-----------	---

---

**Description**

Opens swagger docs in default browser for a live instance of CyREST operations.

**Usage**

```
cyrestAPI(base.url = .defaultBaseUrl)
```

**Arguments**

base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.
----------	--

**Value**

Web page in browser

**Examples**

```
cyrestAPI()
```

---

 cyrestDELETE

*CyREST DELETE*


---

**Description**

Constructs the query, makes DELETE call and processes the result. The function check whether actual call is local or remote first. If remote, requests will go through Jupyter-Bridge.

**Usage**

```
cyrestDELETE(operation = NULL, parameters = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

operation	A string to be converted to the REST query namespace
parameters	A named list of values to be converted to REST query parameters
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

CyREST result content

**Examples**

```
cyrestDELETE('session')
```

---

 cyrestGET

*CyREST GET*


---

**Description**

Constructs the query, makes GET call and processes the result. The function check whether actual call is local or remote first. If remote, requests will go through Jupyter-Bridge.

**Usage**

```
cyrestGET(operation = NULL, parameters = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

operation	A string to be converted to the REST query namespace
parameters	A named list of values to be converted to REST query parameters
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

CyREST result content

**Examples**

```
cyrestGET('version')
```

---

 cyrestPOST

*CyREST POST*


---

**Description**

Constructs the query and body, makes POST call and processes the result. The function check whether actual call is local or remote first. If remote, requests will go through Jupyter-Bridge.

**Usage**

```
cyrestPOST(
  operation,
  parameters = NULL,
  body = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

operation	A string to be converted to the REST query namespace
parameters	A named list of values to be converted to REST query parameters
body	A named list of values to be converted to JSON
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

CyREST result content

**Examples**

```
cyrestPOST('networks/51/views')
```

---

 cyrestPUT

*CyREST PUT*


---

**Description**

Constructs the query and body, makes PUT call and processes the result. The function check whether actual call is local or remote first. If remote, requests will go through Jupyter-Bridge.

**Usage**

```
cyrestPUT(
  operation,
  parameters = NULL,
  body = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

operation	A string to be converted to the REST query namespace
parameters	A named list of values to be converted to REST query parameters
body	A named list of values to be converted to JSON
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

CyREST result content

**Examples**

```
cyrestPUT()
```

---

cytoscapeApiVersions *Available CyREST API Versions*

---

**Description**

Get the list of available CyREST API versions

**Usage**

```
cytoscapeApiVersions(base.url = .defaultBaseUrl)
```

**Arguments**

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A list of API versions as character strings, e.g., "v1"

**Author(s)**

Alexander Pico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
cytoscapeApiVersions()  
# [1] "v1"
```



---

cytoscapeFreeMemory *Free Up Unused Memory for Cytoscape*

---

**Description**

Manually call Java's garbage collection `System.gc()` to free up unused memory. This process happens automatically, but may be useful to call explicitly for testing or evaluation purposes.

**Usage**

```
cytoscapeFreeMemory(base.url = .defaultBaseUrl)
```

**Arguments**

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

status message

**Author(s)**

Alexander Pico

**Examples**

```
cytoscapeFreeMemory()  
# [1] "Unused memory freed up."
```

---

cytoscapeMemoryStatus *Memory Available to Cytoscape*

---

**Description**

Returns the memory resources of the server running Cytoscape

**Usage**

```
cytoscapeMemoryStatus(base.url = .defaultBaseUrl)
```

**Arguments**

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A list of numeric values

**Author(s)**

Alexander Pico

**Examples**

```
cytoscapeMemoryStatus()
# usedMemory freeMemory totalMemory maxMemory
#          181         2624         2805         13653
```

---

cytoscapeNumberOfCores

*Number of Processors Available to Cytoscape*

---

**Description**

Returns the processor resources of the server running Cytoscape

**Usage**

```
cytoscapeNumberOfCores(base.url = .defaultBaseUrl)
```

**Arguments**

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A numeric value

**Author(s)**

Alexander Pico

**Examples**

```
cytoscapeNumberOfCores()
# [1] 8
```

---

cytoscapePing	<i>Ping Cytoscape</i>
---------------	-----------------------

---

**Description**

Tests the connection to Cytoscape via CyREST and verifies that supported versions of Cytoscape and CyREST API are loaded.

**Usage**

```
cytoscapePing(base.url = .defaultBaseUrl)
```

**Arguments**

base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.
----------	---

**Value**

status message

**Author(s)**

Alexander Pico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
cytoscapePing()  
# [1] "You are connected to Cytoscape!"
```

---

cytoscapeVersionInfo	<i>Cytoscape and CyREST API Versions</i>
----------------------	--

---

**Description**

Returns the versions of the current Cytoscape and CyREST API

**Usage**

```
cytoscapeVersionInfo(base.url = .defaultBaseUrl)
```

**Arguments**

base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.
----------	---

**Value**

list of versions

**Author(s)**

Alexander Pico

**Examples**

```
cytoscapeVersionInfo()  
# apiVersion cytoscapeVersion  
# "v1" "3.7.0-SNAPSHOT"
```

---

deleteAllNetworks      *Delete All Networks*

---

**Description**

Delete all networks from the current Cytoscape session.

**Usage**

```
deleteAllNetworks(base.url = .defaultBaseUrl)
```

**Arguments**

base.url      (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
deleteAllNetworks()
```

---

deleteAnnotation	<i>Delete Annotation</i>
------------------	--------------------------

---

**Description**

Remove an annotation from the current network view in Cytoscape

**Usage**

```
deleteAnnotation(names = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

names	List of annotations by UUID or Name
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
deleteAnnotation("016a4af1-69bc-4b99-8183-d6f118847f96")
deleteAnnotation(c("T2", "T3"))
deleteAnnotation(sapply(getAnnotationList(), '[', 'uuid'))
```

---

deleteDuplicateEdges	<i>Delete Duplicate Edges</i>
----------------------	-------------------------------

---

**Description**

Removes edges with duplicate names. Only considers cases with identical source, target, interaction and directionality.

**Usage**

```
deleteDuplicateEdges(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Duplicate edges are first selected and then deleted. Prior edge selections will be lost; node selections will not be affected.

**Value**

Lists of SUIDs for selected nodes and edges

**Examples**

```
deleteDuplicateEdges()
```

---

deleteGroup	<i>Delete (or Ungroup) a Group</i>
-------------	------------------------------------

---

**Description**

Deletes one or more groups, while leaving member nodes intact.

**Usage**

```
deleteGroup(
  groups = NULL,
  groups.by.col = "SUID",
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

groups	(optional) List of group SUIDs, names, other column values or keywords: all, selected, unselected. Default is the currently selected group.
groups.by.col	name of node table column corresponding to provided groups list. Default is 'SUID'.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Note: group nodes are ungrouped but not deleted in Cytoscape version 3.6.1

**Value**

None

**Examples**

```
deleteGroup()
```

---

<code>deleteNetwork</code>	<i>Delete Network</i>
----------------------------	-----------------------

---

**Description**

Delete a network from the current Cytoscape session.

**Usage**

```
deleteNetwork(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

- `network` (optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
- `base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
deleteNetwork()
```

deleteSelectedEdges *Delete Selected Edges*

---

**Description**

Delete the currently selected edges in the network.

**Usage**

```
deleteSelectedEdges(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

list of deleted edge SUIDs

**Author(s)**

AlexanderPico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
deleteSelectedEdges()
```

---

deleteSelectedNodes *Delete Selected Nodes*

---

**Description**

Delete currently selected nodes from the network.

**Usage**

```
deleteSelectedNodes(network = NULL, base.url = .defaultBaseUrl)
```



**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A named list of deleted node suids (\$nodes) as well as edge suids (\$edges) deleted as a result of the node deletion

**Examples**

```
deleteSelectedNodes()
```

---

deleteSelfLoops	<i>Delete Self Loops</i>
-----------------	--------------------------

---

**Description**

Removes edges that connect to a single node as both source and target.

**Usage**

```
deleteSelfLoops(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Self loop edges are first selected and then deleted. Prior edge and node selections will be lost.

**Value**

Lists of SUIDs for selected nodes and edges

**Examples**

```
deleteSelfLoops()
```

---

deleteStyleMapping	<i>Delete Style Mapping</i>
--------------------	-----------------------------

---

**Description**

Deletes a specified visual style mapping from specified style.

**Usage**

```
deleteStyleMapping(style.name, visual.prop, base.url = .defaultBaseUrl)
```

**Arguments**

style.name	(char) name for style
visual.prop	(char) name of visual property to map. See <code>getVisualPropertyNames()</code> .
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
deleteStyleMapping()
```

---

deleteTableColumn	<i>Delete a table column</i>
-------------------	------------------------------

---

**Description**

Delete a column from node, edge or network tables.

**Usage**

```
deleteTableColumn(
  column,
  table = "node",
  namespace = "default",
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

column	Name of the column to delete
table	Name of table, e.g., node (default), edge, network
namespace	Namespace of table. Default is "default".
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A data.frame of column values

**Examples**

```
deleteTableColumn('node', 'group')
```

---

deleteVisualStyle      *Delete Visual Style*

---

**Description**

Deletes the specified visual style from current session.

**Usage**

```
deleteVisualStyle(style.name, base.url = .defaultBaseUrl)
```

**Arguments**

style.name	(char) name of style to delete
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
deleteVisualStyle("myStyle")
```

---

diffusionAdvanced      *Diffusion Advanced*

---

### Description

Diffusion will send the selected network view and its selected nodes to a web-based REST service to calculate network propagation. Results are returned and represented by columns in the node table. Advanced operation supports parameters.

### Usage

```
diffusionAdvanced(  
  heat.column.name = NULL,  
  time = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

heat.column.name	(optional) A node column name intended to override the default table column 'diffusion_input'. This represents the query vector and corresponds to h in the diffusion equation.
time	(optional) The extent of spread over the network. This corresponds to t in the diffusion equation.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

### Details

Columns are created for each execution of Diffusion and their names are returned in the response. The nodes you would like to use as input should be selected. This will be used to generate the contents of the diffusion\_input column, which represents the query vector and corresponds to h in the diffusion equation.

### Value

Version number

### Examples

```
diffusionAdvanced()
```

---

`diffusionBasic`*Diffusion Basic*

---

**Description**

Diffusion will send the selected network view and its selected nodes to a web-based REST service to calculate network propagation. Results are returned and represented by columns in the node table.

**Usage**

```
diffusionBasic(base.url = .defaultBaseUrl)
```

**Arguments**

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Columns are created for each execution of Diffusion and their names are returned in the response. The nodes you would like to use as input should be selected. This will be used to generate the contents of the `diffusion_input` column, which represents the query vector and corresponds to `h` in the diffusion equation.

**Value**

Version number

**Examples**

```
diffusionBasic()
```

---

`disableApp`*Disable App*

---

**Description**

Disable an app to effectively remove it from your Cytoscape session without having to uninstall it.

**Usage**

```
disableApp(app, base.url = .defaultBaseUrl)
```

**Arguments**

app	Name of app
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
disableApp()
```

---

dockPanel

*Dock Panel*


---

**Description**

Dock a panel back into the UI of Cytoscape.

**Usage**

```
dockPanel(panel.name, base.url = .defaultBaseUrl)
```

**Arguments**

panel.name	Name of the panel. Multiple ways of referencing panels is supported: WEST, control panel, control, c SOUTH, table panel, table, ta SOUTH_WEST, tool panel, tool, to EAST, results panel, results, r
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
dockPanel('table')
```

---

doInitializeSandbox     *doInitializeSandbox*


---

**Description**

doInitializeSandbox

**Usage**

doInitializeSandbox(requester = NULL, base.url = .defaultBaseUrl)

**Arguments**

requester	requester
base.url	Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://127.0.0.1:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

doInitializeSandbox()

---

doRequestRemote     *doRequestRemote*


---

**Description**

Do requests remotely by connecting over Jupyter-Bridge.

**Usage**

doRequestRemote(method, qurl, qbody = NULL, headers = NULL)

**Arguments**

method	A string to be converted to the REST query namespace
qurl	A named list of values to be converted to REST query parameters
qbody	A named list of values to be converted to JSON
headers	httr headers

**Value**

httr response

**Examples**

```
doRequestRemote()
```

---

doSetSandbox

*doSetSandbox*

---

**Description**

doSetSandbox

**Usage**

```
doSetSandbox(sandboxToSet, requester = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

sandboxToSet    sandbox to set

requester       requester

base.url        Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://127.0.0.1:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
doSetSandbox()
```



---

 enableApp

*Enable App*


---

**Description**

Enable a previously installed and disabled app in Cytoscape.

**Usage**

```
enableApp(app, base.url = .defaultBaseUrl)
```

**Arguments**

app	Name of app
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
enableApp()
```

---

expandGroup

*Expand Group*


---

**Description**

Replaces the group node with member nodes for a set of groups.

**Usage**

```
expandGroup(groups = NULL, network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

groups	(optional) List of group names or keywords: all, selected, unselected. Default is the currently selected group.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
expandGroup()
```

---

```
exportFilters
```

```
Export Filters
```

---

**Description**

Saves filters to file in JSON format.

**Usage**

```
exportFilters(  
  filename = "filters.json",  
  base.url = .defaultBaseUrl,  
  overwriteFile = TRUE  
)
```

**Arguments**

filename	(char) Full path or path relative to current working directory, in addition to the name of the file. Default is "filters.json"
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.
overwriteFile	(optional) FALSE allows an error to be generated if the file already exists; TRUE allows Cytoscape to overwrite it without asking. Default value is TRUE.

**Details**

Unlike other export functions, Cytoscape will automatically overwrite files with the same name. You will not be prompted to confirm or reject overwrite. Use carefully!

**Value**

None

**Examples**

```
exportFilters()
```

---

 exportImage
*Export Image***Description**

Saves the current network view as an image file.

**Usage**

```
exportImage(
  filename = NULL,
  type = "PNG",
  resolution = NULL,
  units = NULL,
  height = NULL,
  width = NULL,
  zoom = NULL,
  network = NULL,
  base.url = .defaultBaseUrl,
  overwriteFile = TRUE
)
```

**Arguments**

filename	(character) Full path or path relative to current working directory, in addition to the name of the file. Extension is automatically added based on the type argument. If blank, the current network name will be used.
type	(character) Type of image to export, e.g., PNG (default), JPEG, PDF, SVG, PS (PostScript).
resolution	(numeric) The resolution of the exported image, in DPI. Valid only for bitmap formats, when the selected width and height 'units' is inches. The possible values are: 72 (default), 100, 150, 300, 600.
units	(character) The units for the 'width' and 'height' values. Valid only for bitmap formats, such as PNG and JPEG. The possible values are: pixels (default), inches.
height	(numeric) The height of the exported image. Valid only for bitmap formats, such as PNG and JPEG.
width	(numeric) The width of the exported image. Valid only for bitmap formats, such as PNG and JPEG.
zoom	(numeric) The zoom value to proportionally scale the image. The default value is 100.0. Valid only for bitmap formats, such as PNG and JPEG
network	(optional) Name or SUID of a network or view. Default is the "current" network active in Cytoscape. The first (presumably only) view associated a network is used if a specific view SUID is not provided.

base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.
overwriteFile	(optional) FALSE allows Cytoscape show a message box before overwriting the file if the file already exists; TRUE allows Cytoscape to overwrite it without asking. Default value is TRUE.

### Details

The image is cropped per the current view in Cytoscape. Consider applying [fitContent](#) prior to export.

### Value

server response

### Examples

```
exportImage('/fullpath/myNetwork', 'PDF')
```

---

exportNetwork	<i>Export Network</i>
---------------	-----------------------

---

### Description

Export a network to one of multiple file formats

### Usage

```
exportNetwork(  
  filename = NULL,  
  type = "SIF",  
  network = NULL,  
  base.url = .defaultBaseUrl,  
  overwriteFile = TRUE  
)
```

### Arguments

filename	Full path or path relative to current working directory, in addition to the name of the file. Extension is automatically added based on the type argument. If blank, then the current network name is used.
type	File type. SIF (default), CX, cyjs, graphML, NNF, xGMLL.
network	(optional) Name or SUID of a network or view. Default is the "current" network active in Cytoscape.

base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.
overwriteFile	(optional) FALSE allows Cytoscape show a message box before overwriting the file if the file already exists; TRUE. allows Cytoscape to overwrite it without asking. Default value is TRUE.

**Value**

None.

**Examples**

```
exportNetwork('/path/filename','SIF')
```

---

exportNetworkToNDEx	<i>Export Network To NDEx</i>
---------------------	-------------------------------

---

**Description**

Send a copy of a Cytoscape network to NDEx as a new submission.

**Usage**

```
exportNetworkToNDEx(
  username,
  password,
  isPublic,
  network = NULL,
  metadata = NULL,
  ndex.url = "http://ndexbio.org",
  ndex.version = "v2",
  base.url = .defaultBaseUrl
)
```

**Arguments**

username	NDEx account username
password	NDEx account password
isPublic	(Boolean) Whether to make the network publicly accessible at NDEx.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
metadata	(optional) A list of structured information describing the network
ndex.url	(optional) For alternative instances or deployments of NDEx; default is "http://ndexbio.org"
ndex.version	(optional) For alternative versions of the NDEx API; default is "v2"

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

### Value

NDEx identifier (`externalId`) for new submission

### Examples

```
exportNetworkToNDEx("user", "pass", TRUE)
```

---

`exportVisualStyles`      *Export Visual Styles*

---

### Description

Save one or more visual styles to file.

### Usage

```
exportVisualStyles(
  filename = NULL,
  type = "XML",
  styles = NULL,
  base.url = .defaultBaseUrl,
  overwriteFile = TRUE
)
```

### Arguments

<code>filename</code>	(char) Full path or path relative to current working directory, in addition to the name of the file. Extension is automatically added based on the <code>type</code> argument. Default is "styles.xml"
<code>type</code>	(optional) Type of data file to export, e.g., XML, JSON (case sensitive). Default is XML. Note: Only XML can be read by <code>importVisualStyles()</code> .
<code>styles</code>	(optional) The styles to be exported, listed as a comma-separated string. If no styles are specified, only the current one is exported.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.
<code>overwriteFile</code>	(optional) FALSE allows Cytoscape show a message box before overwriting the file if the file already exists; TRUE. allows Cytoscape to overwrite it without asking. Default value is TRUE.

**Value**

Path to saved file

**See Also**

`importVisualStyles`

**Examples**

```
exportVisualStyles('/fullpath/myStyle')  
exportVisualStyles('/fullpath/myStyle', type = 'JSON')  
exportVisualStyles('/fullpath/myStyle', style = 'Minimal,default,Directed')
```

---

`findRemoteCytoscape`     *findRemoteCytoscape*

---

**Description**

Figure out whether CyREST is local or remote. If remote, we'll want to go through Jupyter-Bridge.

**Usage**

```
findRemoteCytoscape()
```

**Value**

None

**Examples**

```
findRemoteCytoscape()
```

---

`fitContent`     *Fit Content*

---

**Description**

Zoom and pan network view to maximize either height or width of current network window.

**Usage**

```
fitContent(selected.only = FALSE, network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>selected.only</code>	(Boolean) Whether to fit only current selection. Default is false, i.e., to fit the entire network.
<code>network</code>	(optional) Name or SUID of a network or view. Default is the "current" network active in Cytoscape. The first (presumably only) view associated a network is used if a specific view SUID is not provided.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Takes first (presumably only) view associated with provided network

**Value**

None

**Examples**

```
fitContent()
```

---

`floatPanel`

*Float Panel*

---

**Description**

Pop out a panel from the UI of Cytoscape. Other panels will expand into the space.

**Usage**

```
floatPanel(panel.name, base.url = .defaultBaseUrl)
```

**Arguments**

<code>panel.name</code>	Name of the panel. Multiple ways of referencing panels is supported: WEST, control panel, control, c SOUTH, table panel, table, ta SOUTH_WEST, tool panel, tool, to EAST, results panel, results, r
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None



**Examples**

```
floatPanel('table')
```

---

<code>getAbsSandboxPath</code>	<i>getAbsSandboxPath</i>
--------------------------------	--------------------------

---

**Description**

Get absolute sandbox path.

**Usage**

```
getAbsSandboxPath(fileLocation)
```

**Arguments**

fileLocation    fileLocation

**Value**

file location

**Examples**

```
getAbsSandboxPath()
```

---

<code>getAllEdges</code>	<i>Get All Edges</i>
--------------------------	----------------------

---

**Description**

Retrieve the names of all the edges in the network.

**Usage**

```
getAllEdges(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network                    (optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.

base.url                    (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

list of node edges

**Examples**

```
getAllEdges()
```

---

getAllNodes

*Get All Nodes*

---

**Description**

Retrieve the names of all the nodes in the network.

**Usage**

```
getAllNodes(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network (optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.

base.url (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

list of node names

**Examples**

```
getAllNodes()
```

---

getAllStyleMappings     *Get All Style Mappings*

---

**Description**

Retrieve the values that define all the mappings per visual property in a given style.

**Usage**

```
getAllStyleMappings(style.name = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

style.name	Name for style; default is the 'default' style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

List of lists with named values defining the visual property mappings

**Examples**

```
getStyleMapping()
```

---

getAnnotationList     *Get Annotation List*

---

**Description**

A list of named lists with annotation information

**Usage**

```
getAnnotationList(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

You can obtain a list of UUIDs by applying a subset function like so: `sapply(getAnnotationList(), '[', 'uuid')`

**Value**

RETURN\_DESCRIPTION

**Examples**

```
getAnnotationList()
```

---

getAppInformation	<i>Get App Information</i>
-------------------	----------------------------

---

**Description**

Retrieve the name, brief description and version of a Cytoscape app.

**Usage**

```
getAppInformation(app, base.url = .defaultBaseUrl)
```

**Arguments**

<code>app</code>	Name of app
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

App name, brief description and version.

**Examples**

```
getAppInformation()
```

---

getAppStatus	<i>App Status</i>
--------------	-------------------

---

**Description**

Retrieve the current status of a Cytoscape app: Installed, Uninstalled or Disabled.

**Usage**

```
getAppStatus(app, base.url = .defaultBaseUrl)
```

**Arguments**

app	Name of app
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

App name and status

**Examples**

```
getAppStatus()
```

---

getAppUpdates	<i>List Apps With Updates</i>
---------------	-------------------------------

---

**Description**

Retrieve list of currently installed Cytoscape apps with updates available.

**Usage**

```
getAppUpdates(base.url = .defaultBaseUrl)
```

**Arguments**

base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.
----------	---

**Value**

A list of updatable app names, versions and statuses

**Examples**

```
getAppUpdates()
```

---

getArrowShapes	<i>Get Arrow Shapes</i>
----------------	-------------------------

---

**Description**

Retrieve the names of the currently supported 'arrows' – the decorations can (optionally) appear at the ends of edges, adjacent to the nodes they connect, and conveying information about the nature of the nodes' relationship.

**Usage**

```
getArrowShapes(base.url = .defaultBaseUrl)
```

**Arguments**

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A list of character strings, e.g., 'DIAMOND', 'T', 'ARROW'

**Examples**

```
getArrowShapes()
```

---

getAvailableApps	<i>List Available Apps</i>
------------------	----------------------------

---

**Description**

Retrieve a list of apps available for installation in Cytoscape.

**Usage**

```
getAvailableApps(base.url = .defaultBaseUrl)
```

**Arguments**

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A list of app names and latest versions

**Examples**

```
getAvailableApps()
```

---

`getBackgroundColorDefault`  
*Get Background Color Default*

---

**Description**

Retrieve the default background color.

**Usage**

```
getBackgroundColorDefault(style.name = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

`style.name` Name of style; default is "default" style

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
getBackgroundColorDefault()
```

---

`getBrowserClientChannel`  
*getBrowserClientChannel*

---

**Description**

Get the unique channel

**Usage**

```
getBrowserClientChannel()
```

**Value**

client channel

**Examples**

```
getBrowserClientChannel()
```

---

`getBrowserClientJs`     *getBrowserClientJs*

---

**Description**

Prepend channel number of client Javascript so it can communicate with this process via Jupyter-bridge

**Usage**

```
getBrowserClientJs()
```

**Value**

Javascript inject code

**Examples**

```
getBrowserClientJs()
```



---

getCollectionList      *Get Collection List*

---

**Description**

FUNCTION\_DESCRIPTION

**Usage**

```
getCollectionList(base.url = .defaultBaseUrl)
```

**Arguments**

base.url      (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

RETURN\_DESCRIPTION

**Examples**

```
getCollectionList()
```

---

getCollectionName      *Get Collection Name*

---

**Description**

FUNCTION\_DESCRIPTION

**Usage**

```
getCollectionName(collection.suid = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

collection.suid

DESCRIPTION

base.url

(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

RETURN\_DESCRIPTION

**Examples**`getCollectionName()`

---

`getCollectionNetworks` *Get Collection Networks*

---

**Description**

FUNCTION\_DESCRIPTION

**Usage**`getCollectionNetworks(collection.suid = NULL, base.url = .defaultBaseUrl)`**Arguments**`collection.suid`

DESCRIPTION

`base.url`(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.**Value**

RETURN\_DESCRIPTION

**Examples**`getCollectionNetworks()`

---

`getCollectionSuid`      *Get Collection Suid*

---

**Description**

FUNCTION\_DESCRIPTION

**Usage**

```
getCollectionSuid(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>network</code>	DESCRIPTION
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

RETURN\_DESCRIPTION

**Examples**

```
getCollectionSuid()
```

---

`getCurrentSandbox`      *getCurrentSandbox*

---

**Description**

Return both the current sandbox name and path.

**Usage**

```
getCurrentSandbox()
```

**Value**

current sandbox

**Examples**

```
getCurrentSandbox()
```

---

`getCurrentSandboxName` *getCurrentSandboxName*

---

**Description**

Return the current sandbox name.

**Usage**

`getCurrentSandboxName()`

**Value**

current sandbox name

**Examples**

`getCurrentSandboxName()`

---

`getCurrentSandboxPath` *getCurrentSandboxPath*

---

**Description**

Return the current sandbox path.

**Usage**

`getCurrentSandboxPath()`

**Value**

current sandbox path

**Examples**

`getCurrentSandboxPath()`

---

getCurrentStyle	<i>Get Current Style</i>
-----------------	--------------------------

---

**Description**

Get the current visual style applied to a network.

**Usage**

```
getCurrentStyle(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Name of style

**Examples**

```
getCurrentStyle()  
getCurrentStyle('myNetwork')
```

---

getDefaultSandbox	<i>getDefaultSandbox</i>
-------------------	--------------------------

---

**Description**

Return whatever is the current default sandbox properties.

**Usage**

```
getDefaultSandbox()
```

**Value**

default sandbox

**Examples**

```
getDefaultSandbox()
```

---

`getDefaultSandboxPath` *getDefaultSandboxPath*

---

**Description**

Return the default path, which isn't one of the properties tracked in the `default_sandbox`.

**Usage**

```
getDefaultSandboxPath()
```

**Value**

default sandbox path

**Examples**

```
getDefaultSandboxPath()
```

---

`getDisabledApps` *List Disabled Apps*

---

**Description**

Retrieve list of currently disabled apps in Cytoscape.

**Usage**

```
getDisabledApps(base.url = .defaultBaseUrl)
```

**Arguments**

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A list of disabled app names, versions and statuses

**Examples**

```
getDisabledApps()
```

---

getEdgeColor	<i>Get Edge Color</i>
--------------	-----------------------

---

**Description**

Retrieve the actual line color of specified edges.

**Usage**

```
getEdgeColor(edge.names = NULL, network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

edge.names	List of edge names or SUIDs. Default is NULL for all edges.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Named list of property values

**Examples**

```
getEdgeColor()
```

---

getEdgeCount	<i>Get Edge Count</i>
--------------	-----------------------

---

**Description**

Reports the number of the edges in the network.

**Usage**

```
getEdgeCount(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

numeric

**Author(s)**

Alexander Pico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
getEdgeCount()
```

---

getEdgeInfo

*Get Edge Information*

---

**Description**

Returns source, target and edge table row values.

**Usage**

```
getEdgeInfo(edges, network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

edges	List of SUIDs or names of edges, i.e., values in the "name" column. Can also input a single edge.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Details**

This function is kinda slow. It takes approximately 70ms per edge to return a result, e.g., 850 edges will take a one minute.

**Value**

named list of lists

**Author(s)**

Alexander Pico



**Examples**

```
getEdgeInfo()
```

---

getEdgeLineStyle      *Get Edge Line Style*

---

**Description**

Retrieve the actual line style of specified edges.

**Usage**

```
getEdgeLineStyle(edge.names = NULL, network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

- edge.names      List of edge names or SUIDs. Default is NULL for all edges.
- network      (optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
- base.url      (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Named list of property values

**Examples**

```
getEdgeLineStyle()
```

---

getEdgeLineWidth      *Get Edge Line Width*

---

**Description**

Retrieve the actual line width of specified edges.

**Usage**

```
getEdgeLineWidth(edge.names = NULL, network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

edge.names	List of edge names or SUIDs. Default is NULL for all edges.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Named list of property values

**Examples**

```
getEdgeLineWidth()
```

---

getEdgeProperty	<i>Get Edge Property Values</i>
-----------------	---------------------------------

---

**Description**

Get values for any edge property of the specified edges.

**Usage**

```
getEdgeProperty(
  edge.names = NULL,
  visual.property,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

edge.names	List of edge names or SUIDs. Default is NULL for all edges.
visual.property	Name of a visual property. See <a href="#">getVisualPropertyNames</a> .
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

This method retrieves the actual property of the node, given the current visual style, factoring together any default, mapping and bypass setting.

**Value**

Named list of property values

**Examples**

```
getEdgeProperty(c('node 0 (pp) node 1', 'node 0 (pp) node 2'), 'EDGE_WIDTH')
```

---

`getEdgeSelectionColorDefault`  
*Get Edge Selection Color Default*

---

**Description**

Retrieve the default selected edge color.

**Usage**

```
getEdgeSelectionColorDefault(style.name = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>style.name</code>	Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
getEdgeSelectionColorDefault()
```

---

`getEdgeTargetArrowShape`*Get Edge Target Arrow Shape*

---

**Description**

Retrieve the actual target arrow shape of specified edges.

**Usage**

```
getEdgeTargetArrowShape(  
  edge.names = NULL,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

<code>edge.names</code>	List of edge names or SUIDs. Default is NULL for all edges.
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Named list of property values

**Examples**

```
getEdgeTargetArrowShape()
```

---

`getFilterList`*Get Filter List*

---

**Description**

Retrieve list of named filters in current session.

**Usage**

```
getFilterList(base.url = .defaultBaseUrl)
```

**Arguments**

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

List of filter names

**Examples**

```
getFilterList()
```

---

`getFirstNeighbors`      *Get list of nodes neighboring provided list*

---

**Description**

Returns a non-redundant list of first neighbors of the supplied list of nodes or current node selection.

**Usage**

```
getFirstNeighbors(
  node.names = NULL,
  as.nested.list = FALSE,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

`node.names` A list of SUIDs or names from the name column of the node table. Default is currently selected nodes.

`as.nested.list` logical Whether to return lists of neighbors per query node

`network` (optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A list of unique node names, optionally nested per query node name.

**Author(s)**

Alexander Pico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**See Also**

selectNodes selectFirstNeighbors

**Examples**

```
getFirstNeighbors()
```

---

getGroupInfo

*Get Group Information*

---

**Description**

Retrieve information about a group by name or identifier.

**Usage**

```
getGroupInfo(group, network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

group	Group name or SUID.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Group information

**Examples**

```
getGroupInfo('myGroup')
```

---

getInstalledApps      *List Installed Apps*

---

**Description**

Retrieve list of currently installed apps in Cytoscape.

**Usage**

```
getInstalledApps(base.url = .defaultBaseUrl)
```

**Arguments**

`base.url`      (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A list of installed app names, versions and statuses

**Examples**

```
getInstalledApps()
```

---

getJupyterBridgeURL      *getJupyterBridgeURL*

---

**Description**

Get the jupyter bridge server url

**Usage**

```
getJupyterBridgeURL()
```

**Value**

jupyter bridge server url

**Examples**

```
getJupyterBridgeURL()
```

---

getLayoutNameMapping    *Get Layout Name Mapping*

---

### Description

The Cytoscape 'Layout' menu lists many layout algorithms, but the names presented there are different from the names by which these algorithms are known to layout method. This method returns a named list in which the names are from the GUI, and the values identify the names you must use to choose an algorithms in the programmatic interface.

### Usage

```
getLayoutNameMapping(base.url = .defaultBaseUrl)
```

### Arguments

base.url            (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

### Value

A named list of character strings

### Author(s)

Alexander Pico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

### Examples

```
getLayoutNameMapping()
# Degree Sorted Circle Layout    Group Attributes Layout    Edge-weighted Spring Embedded Layout
#            "degree-circle"            "attributes-layout"            "kamada-kawai"
```

---

getLayoutNames            *Get Layout Names*

---

### Description

Retrieve the names of the currently supported layout algorithms. These may be used in subsequent calls to the 'layoutNetwork' function.

### Usage

```
getLayoutNames(base.url = .defaultBaseUrl)
```



**Arguments**

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A list of character strings, e.g., "force-directed" "circular" "grid"

**Author(s)**

Alexander Pico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
getLayoutNames()
# [1] "degree-circle"      "attributes-layout"    "kamada-kawai"
# [4] "force-directed"    "cose"                 "hierarchical"
# [7] "attribute-circle"  "stacked-node-layout" "circular"
```

---

getLayoutPropertyNames

*Get Layout Property Names*

---

**Description**

Returns a list of the tunable properties for the specified layout.

**Usage**

```
getLayoutPropertyNames(layout.name, base.url = .defaultBaseUrl)
```

**Arguments**

`layout.name` (character) Name of the layout

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Run [getLayoutNames](#) to list available layouts.

**Value**

A list of character strings

**Author(s)**

Alexander Pico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
getLayoutPropertyNames('force-directed')
# [1] "numIterations"      "defaultSpringCoefficient" "defaultSpringLength"
# [4] "defaultNodeMass"   "isDeterministic"        "singlePartition"
```

---

getLayoutPropertyType *Get Layout Property Type*

---

**Description**

Returns the type of one of the tunable properties (property.name) for the specified layout.

**Usage**

```
getLayoutPropertyType(layout.name, property.name, base.url = .defaultBaseUrl)
```

**Arguments**

layout.name	(character) Name of the layout
property.name	(character) Name of the property
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Run [getLayoutNames](#) to list available layouts. Run [getLayoutPropertyNames](#) to list properties per layout.

**Value**

A character string specifying the type

**Author(s)**

Alexander Pico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
getLayoutPropertyType('force-directed', 'defaultSpringLength')
# "double"
```

---

`getLayoutPropertyValue`*Get Layout Property Value*

---

**Description**

Returns the appropriately typed value of the specified tunable property for the specified layout.

**Usage**

```
getLayoutPropertyValue(layout.name, property.name, base.url = .defaultBaseUrl)
```

**Arguments**

<code>layout.name</code>	(character) Name of the layout
<code>property.name</code>	(character) Name of the property
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Run [getLayoutNames](#) to list available layouts. Run [getLayoutPropertyNames](#) to list properties per layout.

**Value**

The current value set for this layout property. Typically an integer, numeric or character string value.

**Author(s)**

Alexander Pico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
getLayoutPropertyValue('force-directed', 'defaultSpringLength')  
# 80
```

---

getLineStyle	<i>Get Line Styles</i>
--------------	------------------------

---

**Description**

Retrieve the names of the currently supported line types – values which can be used to render edges, and thus can be used in calls to 'setEdgeLineStyleRule'.

**Usage**

```
getLineStyle(base.url = .defaultBaseUrl)
```

**Arguments**

base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.
----------	--

**Value**

A list of character strings, e.g., 'SOLID', 'DOT'

**Examples**

```
getLineStyle()
```

---

getNetworkCenter	<i>Get Network Center</i>
------------------	---------------------------

---

**Description**

Retrieve the center of specified network.

**Usage**

```
getNetworkCenter(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Named list of property values

**Examples**

```
getNetworkCenter()
```

---

<code>getNetworkCount</code>	<i>Get the number of Cytoscape networks</i>
------------------------------	---

---

**Description**

Returns the number of Cytoscape networks in the current Cytoscape session

**Usage**

```
getNetworkCount(base.url = .defaultBaseUrl)
```

**Arguments**

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

numeric

**Author(s)**

Alexander Pico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
getNetworkCount()  
# 3
```

---

getNetworkList      *Get the list of Cytoscape networks*

---

**Description**

Returns the list of Cytoscape network names in the current Cytoscape session

**Usage**

```
getNetworkList(base.url = .defaultBaseUrl)
```

**Arguments**

base.url      (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

list

**Author(s)**

Alexander Pico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
getNetworkList()  
# 3
```

---

getNetworkName      *Get the name of a network*

---

**Description**

Retrieve the title of a network

**Usage**

```
getNetworkName(suid = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

suid	(optional) SUID of the network; default is current network. If a name is provided, then it is validated and returned.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

network name

**Examples**

```
getNetworkName()
getNetworkName(1111)
```

---

getNetworkNDExId	<i>Get Network NDEx Id</i>
------------------	----------------------------

---

**Description**

Retrieve the NDEx externalId for a Cytoscape network, presuming it has already been exported to NDEx.

**Usage**

```
getNetworkNDExId(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

If the Cytoscape network is not associated with an NDEx network, the return value will be NULL.

**Value**

NDEx identifier (externalId) or NULL

**Examples**

```
getNetworkNDExId()
```

---

getNetworkProperty      *Get Network Property Values*

---

### Description

Get values for any network property.

### Usage

```
getNetworkProperty(visual.property, network = NULL, base.url = .defaultBaseUrl)
```

### Arguments

visual.property	Name of a visual property. See <a href="#">getVisualPropertyNames</a> .
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method retrieves the actual property of the node, given the current visual style, factoring together any default, mapping and bypass setting.

### Value

Named list of property values

### Examples

```
getNetworkProperty('NETWORK_SCALE_FACTOR')
```

---

getNetworkSuid      *Get the SUID of a network*

---

### Description

Retrieve the SUID of a network

### Usage

```
getNetworkSuid(title = NULL, base.url = .defaultBaseUrl)
```



**Arguments**

title	(optional) Name of the network; default is "current" network. If an SUID is provided, then it is validated and returned.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

(numeric) Network suid

**Author(s)**

Alexander Pico

**Examples**

```
getNetworkSuid()
getNetworkSuid("myNetwork")
# 80
```

---

getNetworkViews

*Get Network Views*


---

**Description**

Retrieve list of network view SUIDs

**Usage**

```
getNetworkViews(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

List of network view SUIDs

**Examples**

```
getNetworkViews()
```

---

getNetworkViewSuid      *Get the SUID of a network view*

---

**Description**

Retrieve the SUID of a network view

**Usage**

```
getNetworkViewSuid(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape. If a network view SUID is provided, then it is validated and returned.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

(numeric) Network view suid. The first (presumably only) view associated a network is returned.

**Author(s)**

Alexander Pico

**Examples**

```
getNetworkViewSuid()  
getNetworkViewSuid("myNetwork")  
# 90
```

---

getNetworkZoom      *Get Network Zoom*

---

**Description**

Retrieve the scale factor of specified network.

**Usage**

```
getNetworkZoom(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Named list of property values

**Examples**

```
getNetworkZoom()
```

---

getNodeColor	<i>Get Node Color</i>
--------------	-----------------------

---

**Description**

Retrieve the actual fill color of specified nodes.

**Usage**

```
getNodeColor(node.names = NULL, network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

node.names	List of node names or SUIDs. Default is NULL for all nodes.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Named list of property values

**Examples**

```
getNodeColor()
```

---

`getNodeCount`*Get Node Count*

---

**Description**

Reports the number of nodes in the network.

**Usage**

```
getNodeCount(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

numeric

**Author(s)**

Alexander Pico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
getNodeCount()
```

---

`getNodeHeight`*Get Node Height*

---

**Description**

Retrieve the actual height of specified nodes.

**Usage**

```
getNodeHeight(node.names = NULL, network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

node.names	List of node names or SUIDs. Default is NULL for all nodes.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Named list of property values

**Examples**

```
getNodeHeight()
```

---

getNodePosition	<i>Get Node Position</i>
-----------------	--------------------------

---

**Description**

Retrieve the actual x,y position of specified nodes.

**Usage**

```
getNodePosition(node.names = NULL, network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

node.names	List of node names or SUIDs. Default is NULL for all nodes.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Dataframe of x, y values

**Examples**

```
getNodePosition()
```

---

getNodeProperty	<i>Get Node Property Values</i>
-----------------	---------------------------------

---

### Description

Get values for any node property of the specified nodes.

### Usage

```
getNodeProperty(  
  node.names = NULL,  
  visual.property,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

node.names	List of node names or SUIDs. Default is NULL for all nodes.
visual.property	Name of a visual property. See <a href="#">getVisualPropertyNames</a> .
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method retrieves the actual property of the node, given the current visual style, factoring together any default, mapping and bypass setting.

### Value

Named list of property values

### Examples

```
getNodeProperty(c('node 0', 'node 1'), 'NODE_SHAPE')
```

---

`getNodeSelectionColorDefault`*Get Node Selection Color Default*

---

**Description**

Retrieve the default selection node color.

**Usage**

```
getNodeSelectionColorDefault(style.name = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>style.name</code>	Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
getNodeSelectionColorDefault()
```

---

`getNodeShapes`*Get Node Shapes*

---

**Description**

Retrieve the names of the currently supported node shapes, which can then be used in calls to `setNodeShapeRule` and `setDefaultVizMapValue`.

**Usage**

```
getNodeShapes(base.url = .defaultBaseUrl)
```

**Arguments**

<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.
-----------------------	---

**Value**

A list of character strings, e.g. 'ELLIPSE', 'RECTANGLE'

**Examples**

```
getNodeShapes()
```

---

getNodeSize

*Get Node Size*

---

**Description**

Retrieve the actual size of specified nodes.

**Usage**

```
getNodeSize(node.names = NULL, network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

node.names	List of node names or SUIDs. Default is NULL for all nodes.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Named list of property values

**Examples**

```
getNodeSize()
```



---

getNodeWidth	<i>Get Node Width</i>
--------------	-----------------------

---

**Description**

Retrieve the actual width of specified nodes.

**Usage**

```
getNodeWidth(node.names = NULL, network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

node.names	List of node names or SUIDs. Default is NULL for all nodes.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Named list of property values

**Examples**

```
getNodeWidth()
```

---

getNotebookIsRunning	<i>getNotebookIsRunning</i>
----------------------	-----------------------------

---

**Description**

```
getNotebookIsRunning
```

**Usage**

```
getNotebookIsRunning()
```

**Value**

None

**Examples**

```
getNotebookIsRunning()
```

---

```
getSandboxReinitialize
    getSandboxReinitialize
```

---

**Description**

Return flag indicating that next command should reinitialize the sandbox according to the default\_sandbox.

**Usage**

```
getSandboxReinitialize()
```

**Value**

sandbox reinitialize

**Examples**

```
getSandboxReinitialize()
```

---

```
getSelectedEdgeCount  Get Selected Edge Count
```

---

**Description**

Returns the number of edges currently selected in the network.

**Usage**

```
getSelectedEdgeCount(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

numeric

**Author(s)**

AlexanderPico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
getSelectedEdgeCount()
```

---

`getSelectedEdges`      *Get Selected Edges*

---

**Description**

Retrieve the names of all the edges selected in the network.

**Usage**

```
getSelectedEdges(  
    edge.suids = FALSE,  
    network = NULL,  
    base.url = .defaultBaseUrl  
)
```

**Arguments**

<code>edge.suids</code>	Whether to return edge SUIDs. Default is FALSE to return edge names.
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

list of selected edge names

**Examples**

```
getSelectedEdges()
```

---

`getSelectedNodeCount`    *Get Selected Node Count*

---

**Description**

Returns the number of nodes currently selected in the network.

**Usage**

```
getSelectedNodeCount(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

numeric

**Author(s)**

AlexanderPico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
getSelectedNodeCount()
```

---

`getSelectedNodes`    *Get Selected Nodes*

---

**Description**

Retrieve the names of all the nodes selected in the network.

**Usage**

```
getSelectedNodes(  
  node.suids = FALSE,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

<code>node.suids</code>	Whether to return node SUIDs. Default is FALSE to return node names.
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

list of selected node names

**Author(s)**

AlexanderPico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
getSelectedNodes()
```

---

`getStyleDependencies` *Get the values of dependencies in a style*

---

**Description**

Retrieves style dependency settings.

**Usage**

```
getStyleDependencies(style.name = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>style.name</code>	Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

server response

**Available Dependencies**

`arrowColorMatchesEdge` `nodeCustomGraphicsSizeSync` `nodeSizeLocked`

**Examples**

```
getStyleDependencies("myStyle")
```

---

getStyleMapping	<i>Get Style Mapping</i>
-----------------	--------------------------

---

**Description**

Retrieve the values that define the mappings for a given visual property in a style.

**Usage**

```
getStyleMapping(visual.prop, style.name = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>visual.prop</code>	Name of the visual property, e.g., <code>NODE_FILL_COLOR</code>
<code>style.name</code>	Name for style; default is the 'default' style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

List of named values defining the visual property mappings

**See Also**

`getVisualPropertyNames`

**Examples**

```
getStyleMapping()
```

---

getTableColumnNames    *Get Table Column Names*

---

## Description

Retrieve the names of all columns in a table

## Usage

```
getTableColumnNames(  
  table = "node",  
  namespace = "default",  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

## Arguments

table	name of table, e.g., node, edge, network; default is "node"
namespace	namespace of table, e.g., default
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

## Value

list of column names

## Examples

```
getTableColumnNames()  
getTableColumnNames('edge')  
getTableColumnNames('network')
```

---

getTableColumns	<i>Get table column values</i>
-----------------	--------------------------------

---

**Description**

Retrieve one or more columns of data from node, edge or network tables.

**Usage**

```
getTableColumns(
  table = "node",
  columns = NULL,
  namespace = "default",
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

table	name of table, e.g., node (default), edge, network
columns	names of columns to retrieve values from as list object or comma-separated list; default is all columns
namespace	namespace of table; default is "default"
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

**Details**

The 'SUID' column is always retrieved along with specified columns. The 'SUID' values are used as `row.names` in the returned `data.frame`.

**Value**

A `data.frame` of column values

**Examples**

```
getTableColumns()
getTableColumns('node', 'group')
```



---

getTableColumnTypes    *Get Table Column Types*

---

### Description

Retrieve the types of all columns in a table

### Usage

```
getTableColumnTypes(  
  table = "node",  
  namespace = "default",  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

table	name of table, e.g., node, edge, network; default is "node"
namespace	namespace of table, e.g., default
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

### Value

a named list of column types

### Examples

```
getTableColumnTypes()  
getTableColumnTypes('edge')  
getTableColumnTypes('network')
```

---

getTableValue	<i>Get table cell value</i>
---------------	-----------------------------

---

### Description

Retrieve the value from a specific row and column from node, edge or network tables.

### Usage

```
getTableValue(  
  table,  
  row.name,  
  column,  
  namespace = "default",  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

table	name of table, e.g., node, edge, network
row.name	Node, edge or network name, i.e., the value in the "name" column
column	name of column to retrieve values from
namespace	namespace of table; default is "default"
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

### Value

A data.frame of column values

### Examples

```
getTableValue('node', 'node 1', 'score')
```

---

`getUninstalledApps`     *List Uninstalled Apps*

---

**Description**

Retrieve list of apps not currently installed in Cytoscape.

**Usage**

```
getUninstalledApps(base.url = .defaultBaseUrl)
```

**Arguments**

`base.url`            (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A list of uninstalled app names, versions and statuses

**Examples**

```
getUninstalledApps()
```

---

`getVisualPropertyDefault`  
*Get Visual Property Default*

---

**Description**

Retrieve the default value for a visual property.

**Usage**

```
getVisualPropertyDefault(  
  property,  
  style.name = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

<code>property</code>	Name of property, e.g., <code>NODE_FILL_COLOR</code> (see <a href="#">getVisualPropertyNames</a> )
<code>style.name</code>	Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
getVisualPropertyDefault('NODE_SIZE')
```

---

```
getVisualPropertyNames
```

*Get Visual Property Names*

---

**Description**

Retrieve the names of all possible visual properties.

**Usage**

```
getVisualPropertyNames(base.url = .defaultBaseUrl)
```

**Arguments**

<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.
-----------------------	---

**Value**

List of names

**Examples**

```
getVisualPropertyNames()
```

---

getVisualStyleNames     *Get Visual Style Names*

---

**Description**

Retrieve a list of all visual style names.

**Usage**

```
getVisualStyleNames(base.url = .defaultBaseUrl)
```

**Arguments**

base.url            (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

A list of names

**Examples**

```
getVisualStyleNames()
```

---

hideAllPanels            *Hide All Panels*

---

**Description**

Hide control, table, tool and results panels.

**Usage**

```
hideAllPanels(base.url = .defaultBaseUrl)
```

**Arguments**

base.url            (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
hideAllPanels()
```

---

hideEdges

*Hide Edges*

---

**Description**

Hide (but do not delete) the specified edge or edges, by setting the Visible property bypass value to false.

**Usage**

```
hideEdges(edge.names, network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

edge.names	List of edge names or SUIDs
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

**Details**

This method permanently overrides any default values or mappings defined for this visual property of the edge or edges specified. This method ultimately calls the generic function, [setEdgePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [unhideEdges](#) or [unhideAll](#).

**Value**

None

**See Also**

[setEdgePropertyBypass](#), [hideSelectedEdges](#), [unhideEdges](#), [unhideAll](#)

**Examples**

```
hideEdges()
```

---

`hideNodes`*Hide Nodes*

---

**Description**

Hide (but do not delete) the specified node or nodes, by setting the Visible property bypass value to false.

**Usage**

```
hideNodes(node.names, network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>node.names</code>	List of node names or SUIDs
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Details**

This method permanently overrides any default values or mappings defined for this visual property of the node or nodes specified. This method ultimately calls the generic function, [setNodePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [unhideNodes](#) or [unhideAll](#).

**Value**

None

**See Also**

[setNodePropertyBypass](#), [hideSelectedNodes](#), [unhideNodes](#), [unhideAll](#)

**Examples**

```
hideNodes()
```

---

`hidePanel`*Hide Panel*

---

**Description**

Hide a panel in the UI of Cytoscape. Other panels will expand into the space.

**Usage**

```
hidePanel(panel.name, base.url = .defaultBaseUrl)
```

**Arguments**

<code>panel.name</code>	Name of the panel. Multiple ways of referencing panels is supported: WEST, control panel, control, c SOUTH, table panel, table, ta SOUTH_WEST, tool panel, tool, to EAST, results panel, results, r
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
hidePanel('table')
```

---

`hideSelectedEdges`*Hide Selected Edges*

---

**Description**

Hide (but do not delete) the currently selected edges, by setting the Visible property bypass value to false.

**Usage**

```
hideSelectedEdges(network = NULL, base.url = .defaultBaseUrl)
```



**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

This method permanently overrides any default values or mappings defined for this visual property of the edge or edges specified. This method ultimately calls the generic function, [setEdgePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [unhideEdges](#) or [unhideAll](#).

**Value**

None

**See Also**

[setEdgePropertyBypass](#), [hideEdges](#), [unhideEdges](#), [unhideAll](#)

**Examples**

```
hideSelectedEdges()
```

---

hideSelectedNodes	<i>Hide Selected Nodes</i>
-------------------	----------------------------

---

**Description**

Hide (but do not delete) the currently selected nodes, by setting the Visible property bypass value to false.

**Usage**

```
hideSelectedNodes(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

This method permanently overrides any default values or mappings defined for this visual property of the node or nodes specified. This method ultimately calls the generic function, [setNodePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [unhideNodes](#) or [unhideAll](#).

**Value**

None

**See Also**

[setNodePropertyBypass](#), [hideNodes](#), [unhideNodes](#), [unhideAll](#)

**Examples**

```
hideSelectedNodes()
```

---

importFilters

*Import Filters*

---

**Description**

Loads filters from a file in JSON format.

**Usage**

```
importFilters(filename, base.url = .defaultBaseUrl)
```

**Arguments**

filename	(char) Path and name of the filters file to load.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
importFilters()
```

---

importNetworkFromFile *Import Network From File*

---

**Description**

Loads a network from specified file

**Usage**

```
importNetworkFromFile(file = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

file	Name of file in any of the supported formats (e.g., SIF, GML, xGMML, etc). If NULL, a demo network file in SIF format is loaded.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

(int) network SUID

**Examples**

```
importNetworkFromFile()
```

---

importNetworkFromNDEx *Import Network From NDEx*

---

**Description**

Import a network from the NDEx database into Cytoscape.

**Usage**

```
importNetworkFromNDEx(  
  ndex.id,  
  username = NULL,  
  password = NULL,  
  accessKey = NULL,  
  ndex.url = "http://ndexbio.org",  
  ndex.version = "v2",  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

ndex.id	Network externalId provided by NDEx. This is not the same as a Cytoscape SUID.
username	(optional) NDEx account username; required for private content
password	(optional) NDEx account password; required for private content
accessKey	(optional) NDEx accessKey; alternate access to private content
ndex.url	(optional) For alternative instances or deployments of NDEx; default is "http://ndexbio.org"
ndex.version	(optional) For alternative versions of the NDEx API; default is "v2"
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

(integer) SUID of imported network

**Examples**

```
importNetworkFromNDEx(ndex.id)
```

---

```
importVisualStyles     Import Visual Styles
```

---

**Description**

Loads styles from an XML file and returns the names of the loaded styles.

**Usage**

```
importVisualStyles(filename = "styles.xml", base.url = .defaultBaseUrl)
```

**Arguments**

filename	(char) Name of the style file to load. Only reads XML files. Default is "styles.xml".
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

(list) Names of styles loaded

**See Also**

exportVisualStyles

**Examples**

```
importVisualStyles()
```

---

<code>installApp</code>	<i>Install App</i>
-------------------------	--------------------

---

**Description**

Installs an app in Cytoscape.

**Usage**

```
installApp(app, base.url = .defaultBaseUrl)
```

**Arguments**

<code>app</code>	Name of app
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
installApp()
```

---

<code>invertEdgeSelection</code>	<i>Invert Edge Selection</i>
----------------------------------	------------------------------

---

**Description**

Select all edges that were not selected and deselect all edges that were selected.

**Usage**

```
invertEdgeSelection(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

list of newly selected edge SUIDs

**Author(s)**

AlexanderPico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
invertEdgeSelection()
```

---

`invertNodeSelection`    *Invert Node Selection*

---

**Description**

Select all nodes that were not selected and deselect all nodes that were selected.

**Usage**

```
invertNodeSelection(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

list of newly selected node SUIDs

**Author(s)**

AlexanderPico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
invertNodeSelection()
```

---

 layoutCopycat

*Copy a layout from one network to another*


---

**Description**

Sets the coordinates for each node in the target network to the coordinates of a matching node in the source network.

**Usage**

```
layoutCopycat(
  sourceNetwork,
  targetNetwork,
  sourceColumn = "name",
  targetColumn = "name",
  gridUnmapped = TRUE,
  selectUnmapped = TRUE,
  base.url = .defaultBaseUrl
)
```

**Arguments**

sourceNetwork	(character) The name of network to get node coordinates from
targetNetwork	(character) The name of the network to apply coordinates to
sourceColumn	(optional character) The name of column in the sourceNetwork node table used to match nodes; default is 'name'
targetColumn	(optional character) The name of column in the targetNetwork node table used to match nodes; default is 'name'
gridUnmapped	(optional character) If this is set to true, any nodes in the target network that could not be matched to a node in the source network will be laid out in a grid; default is TRUE
selectUnmapped	(optional character) If this is set to true, any nodes in the target network that could not be matched to a node in the source network will be selected in the target network; default is TRUE
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Optional parameters such as `gridUnmapped` and `selectUnmapped` determine the behavior of target network nodes that could not be matched.

**Value**

None

**Examples**

```
layoutCopycat('network1', 'network2')
```

---

 layoutNetwork

*Apply a layout to a network*


---

**Description**

Apply a layout to a network

**Usage**

```
layoutNetwork(layout.name = NULL, network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

layout.name	(character) Name of the layout (with optional parameters). If not specified, then the preferred layout set in the Cytoscape UI is applied.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Run [getLayoutNames](#) to list available layouts.

**Value**

None

**Examples**

```
layoutNetwork()
layoutNetwork('force-directed')
layoutNetwork('force-directed defaultSpringCoefficient=.00006 defaultSpringLength=80')
```



---

listGroups	<i>List Groups</i>
------------	--------------------

---

**Description**

Retrieve a list of all group SUIDs in a network.

**Usage**

```
listGroups(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

List of group SUIDs

**Examples**

```
listGroups()
```

---

loadTableData	<i>Loads data into Cytoscape tables keyed by row</i>
---------------	--

---

**Description**

This function loads data into Cytoscape node/edge/network tables provided a common key, e.g., name. Data.frame column names will be used to set Cytoscape table column names.

**Usage**

```
loadTableData(  
  data,  
  data.key.column = "row.names",  
  table = "node",  
  table.key.column = "name",  
  namespace = "default",  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

data	(data.frame) each row is a node and columns contain node attributes
data.key.column	(char) name of data.frame column to use as key; default is "row.names"
table	(char) name of Cytoscape table to load data into, e.g., node, edge or network; default is "node"
table.key.column	(char) name of Cytoscape table column to use as key; default is "name"
namespace	namespace of table, e.g., default
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Numeric values will be stored as Doubles in Cytoscape tables. Integer values will be stored as Integers. Character or mixed values will be stored as Strings. Logical values will be stored as Boolean. Lists are stored as Lists by CyREST v3.9+. Existing columns with the same names will keep original type but values will be overwritten.

**Value**

server response

---

lockNodeDimensions      *Lock Node Dimensions*

---

**Description**

Set a boolean value to have node width and height fixed to a single size value.

**Usage**

```
lockNodeDimensions(new.state, style.name = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

new.state	(Boolean) Whether to lock node width and height
style.name	Name of style; default is "default" style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
lockNodeDimensions(TRUE)
```

---

makeSimpleGraph	<i>Make Simple Graph</i>
-----------------	--------------------------

---

**Description**

This function creates a simple graphNEL object with various node and edge attribute types to help demonstrate round trip conversion with Cytoscape networks via RCy3.

**Usage**

```
makeSimpleGraph()
```

**Value**

A graphNEL object with a few nodes, edges and attributes

**See Also**

createNetworkFromGraph, createGraphFromNetwork, makeSimpleIgraph

**Examples**

```
{  
  makeSimpleGraph()  
}
```

---

makeSimpleIgraph	<i>Make Simple Igraph</i>
------------------	---------------------------

---

**Description**

This function creates a simple iGraph object with various node and edge attribute types to help demonstrate round trip conversion with Cytoscape networks via RCy3.

**Usage**

```
makeSimpleIgraph()
```

**Value**

A igraph object with a few nodes, edges and attributes

**See Also**

createNetworkFromIgraph, createIgraphFromNetwork, makeSimpleGraph

**Examples**

```
{
  makeSimpleIgraph()
}
```

---

mapTableColumn

*Map Table Column*


---

**Description**

Perform identifier mapping using an existing column of supported identifiers to populate a new column with identifiers mapped to the originals.

**Usage**

```
mapTableColumn(
  column,
  species,
  map.from,
  map.to,
  force.single = TRUE,
  table = "node",
  namespace = "default",
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

column	Name of column containing identifiers of type specified by map.from.
species	Common name for species associated with identifiers, e.g., Human. See details.
map.from	Type of identifier found in specified column. See details.
map.to	Type of identifier to populate in new column. See details.
force.single	(optional) Whether to return only first result in cases of one-to-many mappings; otherwise the new column will hold lists of identifiers. Default is TRUE.
table	(optional) Name of table, e.g., node (default), edge or network
namespace	(optional) Namespace of table, e.g., default (default), shared or hidden

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

### Details

Supported species: Human, Mouse, Rat, Frog, Zebrafish, Fruit fly, Mosquito, Worm, Arabidopsis thaliana, Yeast, E. coli, Tuberculosis.

Supported identifier types (depending on species): Ensembl, Entrez Gene, Uniprot-TrEMBL, miR-Base, UniGene, HGNC (symbols), MGI, RGD, SGD, ZFIN, FlyBase, WormBase, TAIR.

### Value

dataframe with map.from and map.to columns. Beware: if map.to is not unique, it will be suffixed with an incrementing number in parentheses, e.g., if mapIdentifiers is repeated on the same network. However, the original map.to column will be returned regardless.

### Examples

```
mapped.cols <- mapTableColumn('name', 'Yeast', 'Ensembl', 'SGD')
#           name           SGD
#17920  YER145C S000000947
#17921  YMR058W S000004662
#17922  YJL190C S000003726
#...
```

---

mapVisualProperty	<i>Creates a mapping between an attribute and a visual property</i>
-------------------	---

---

### Description

Generates the appropriate data structure for the "mapping" parameter in updateStyleMapping.

### Usage

```
mapVisualProperty(
  visual.prop,
  table.column,
  mapping.type,
  table.column.values,
  visual.prop.values,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

visual.prop	(char) name of visual property to map
table.column	(char) name of table column to map
mapping.type	(char) continuous, discrete or passthrough (c,d,p)
table.column.values	(list) list of values paired with visual.prop.values; skip for passthrough mapping
visual.prop.values	(list) list of values paired with table.column.values; skip for passthrough mapping
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

The paired list of values must be of the same length or mapping will fail. For gradient mapping, you may include two additional visual.prop.values in the first and last positions to map respectively to values less than and greater than those specified in table.column.values. Mapping will also fail if the data type of table.column.values does not match that of the existing table.column. Note that all imported numeric data are stored as Integers or Doubles in Cytoscape tables; and character or mixed data are stored as Strings.

**Value**

(network=network, base.url=base.url) ready to convert into JSON by style mapping operations

**List of visual properties**

Node Border Stroke	Edge Bend	Network Background Paint
Node Border Paint	Edge Curved	Network Center X Location
Node Border Transparency	Edge Label	Network Center Y Location
Node Border Width	Edge Label Color	Network Center Z Location
Node CustomGraphics 1-9	Edge Label Font Face	Network Depth
Node CustomGraphics Position 1-9	Edge Label Font Size	Network Edge Selection
Node CustomGraphics Size 1-9	Edge Label Transparency	Network Height
Node CustomPaint 1-9	Edge Label Width	Network Node Selection
Node Depth	Edge Line Type	Network Scale Factor
Node Fill Color	Edge Paint	Network Size
Node Height	Edge Selected	Network Title
Node Label	Edge Selected Paint	Network Width
Node Label Color	Edge Source Arrow Selected Paint	
Node Label Font Face	Edge Source Arrow Shape	
Node Label Font Size	Edge Source Arrow Size	
Node Label Position	Edge Source Arrow Unselected Paint	

Node Label Transparency	Edge Stroke Selected Paint
Node Label Width	Edge Stroke Unselected Paint
Node Network Image Visible	Edge Target Arrow Selected Paint
Node Paint	Edge Target Arrow Shape
Node Selected	Edge Target Arrow Size
Node Selected Paint	Edge Target Arrow Unselected Paint
Node Shape	Edge Tooltip
Node Size	Edge Transparency
Node Tooltip	Edge Unselected Paint
Node Transparency	Edge Visible
Node Visible	Edge Visual Property
Node Width	Edge Width
Node X Location	
Node Y Location	
Node Z Location	

**See Also**

updateStyleMapping getVisualStylePropertyNames

**Examples**

```
mapVisualStyleProperty('node fill color', 'score', 'c', c(-4.0, 0.0, 9.0), c('#99CCFF', '#FFFFFF', '#FF7777'))
mapVisualStyleProperty('node shape', 'type', 'd', c('protein', 'metabolite'), c('ellipse', 'rectangle'))
mapVisualStyleProperty('node label', 'alias', 'p')
```

---

matchArrowColorToEdge *Match Arrow Color To Edge*

---

**Description**

Set a boolean value to have arrow shapes share the same color as the edge.

**Usage**

```
matchArrowColorToEdge(new.state, style.name = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

new.state	(Boolean) Whether to match arrow color to edge.
style.name	Name of style; default is "default" style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
matchArrowColorToEdge(TRUE)
```

---

mergeNetworks

*Merge Networks*


---

**Description**

Combine networks via union, intersection, or difference operations. Lots of optional parameters choose from!

**Usage**

```
mergeNetworks(
  sources = NULL,
  title = NULL,
  operation = "union",
  nodeKeys = NULL,
  nodeMergeMap = NULL,
  nodesOnly = FALSE,
  edgeKeys = NULL,
  edgeMergeMap = NULL,
  networkMergeMap = NULL,
  inNetworkMerge = TRUE,
  baseUrl = .defaultBaseUrl
)
```

**Arguments**

sources	List of network names to be merged.
title	(optional) Title of the resulting merged network. Default is a concatenation of operation and source network titles.
operation	(optional) Type of merge: union (default), intersection or difference.
nodeKeys	(optional) An order-dependent list of columns to match nodes across source networks. Default is "name" column for all sources.
nodeMergeMap	(optional) A list of column merge records specifying how to merge node table data. Each record should be of the form: c("network1 column", "network2 column", "merged column", "type"), where column names are provided and type is String, Integer, Double or List.
nodesOnly	(optional) If TRUE, this will merge the node tables and ignore edge and network table data. Default is FALSE.



edgeKeys	(optional) An order-dependent list of columns to match edges across source networks. Default is "name" column for all sources.
edgeMergeMap	(optional) A list of column merge records specifying how to merge edge table data. Each record should be of the form: c("network1 column", "network2 column", "merged column", "type"), where column names are provided and type is String, Integer, Double or List.
networkMergeMap	(optional) A list of column merge records specifying how to merge network table data. Each record should be of the form: c("network1 column", "network2 column", "merged column", "type"), where column names are provided and type is String, Integer, Double or List.
inNetworkMerge	(optional) If TRUE (default), nodes and edges with matching attributes in the same network will be merged.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

SUID of resulting merged network

**Examples**

```
mergeNetworks(c("Network 1", "Network 2"), "Merged Network")
mergeNetworks(c("my network", "string network"), "Merged Network",
              nodeKeys=c("HGNC", "query term"))
```

---

openAppStore

*Open App Store Page*

---

**Description**

Opens the Cytoscape App Store in a new tab in your default browser.

**Usage**

```
openAppStore(app, base.url = .defaultBaseUrl)
```

**Arguments**

app	Name of app
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
openAppStore()
```

---

`openSession`*Open Session File or URL*

---

**Description**

Open a session file or URL. This will clear all networks, tables and styles associated with current session. Be sure to [saveSession](#) first.

**Usage**

```
openSession(file.location = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

`file.location` File path or URL (with 'http' or 'https' prefix). Default is a sample session file.

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**See Also**`saveSession`**Examples**

```
openSession('/fullpath/mySession.CYS')
```

---

paletteColorBrewerAccent

*paletteColorBrewerAccent Qualitative*

---

**Description**

Generate a qualitative Accent Brewer palette of a given size

**Usage**

```
paletteColorBrewerAccent(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 8 . See RColorBrewer::display.brewer.all()

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerAccent()
```

---

paletteColorBrewerBlues

*paletteColorBrewerBlues Sequential*

---

**Description**

Generate a sequential Blues Brewer palette of a given size

**Usage**

```
paletteColorBrewerBlues(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 9 . See RColorBrewer::display.brewer.all()

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerBlues()
```

---

```
paletteColorBrewerBrBG
```

*paletteColorBrewerBrBG Divergent*

---

**Description**

Generate a divergent BrBG Brewer palette of a given size

**Usage**

```
paletteColorBrewerBrBG(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 9 . See RColorBrewer::display.brewer.all()

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerBrBG()
```

---

```
paletteColorBrewerBuGn
```

*paletteColorBrewerBuGn Sequential*

---

**Description**

Generate a sequential BuGn Brewer palette of a given size

**Usage**

```
paletteColorBrewerBuGn(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 9 . See RColorBrewer::display.brewer.all()

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerBuGn()
```

---

```
paletteColorBrewerBuPu  
paletteColorBrewerBuPu Sequential
```

---

**Description**

Generate a sequential BuPu Brewer palette of a given size

**Usage**

```
paletteColorBrewerBuPu(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 9 . See RColorBrewer::display.brewer.all()

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerBuPu()
```

---

```
paletteColorBrewerDark2  
paletteColorBrewerDark2 Qualitative
```

---

**Description**

Generate a qualitative Dark2 Brewer palette of a given size

**Usage**

```
paletteColorBrewerDark2(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 8 . See RColorBrewer::display.brewer.all()

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerDark2()
```

---

```
paletteColorBrewerGnBu
```

```
paletteColorBrewerGnBu Sequential
```

---

**Description**

Generate a sequential GnBu Brewer palette of a given size

**Usage**

```
paletteColorBrewerGnBu(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 9 . See RColorBrewer::display.brewer.all()

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerGnBu()
```

---

paletteColorBrewerGreens

*paletteColorBrewerGreens Sequential*

---

**Description**

Generate a sequential Greens Brewer palette of a given size

**Usage**

```
paletteColorBrewerGreens(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 9 . See RColorBrewer::display.brewer.all()

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerGreens()
```

---

paletteColorBrewerGreys

*paletteColorBrewerGreys Sequential*

---

**Description**

Generate a sequential Greys Brewer palette of a given size

**Usage**

```
paletteColorBrewerGreys(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 9 . See RColorBrewer::display.brewer.all()

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerGreys()
```

---

```
paletteColorBrewerOranges
```

```
paletteColorBrewerOranges Sequential
```

---

**Description**

Generate a sequential Oranges Brewer palette of a given size

**Usage**

```
paletteColorBrewerOranges(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 9 . See RColorBrewer::display.brewer.all()

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerOranges()
```

---

```
paletteColorBrewerOrRd
```

```
paletteColorBrewerOrRd Sequential
```

---

**Description**

Generate a sequential OrRd Brewer palette of a given size

**Usage**

```
paletteColorBrewerOrRd(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 9 . See RColorBrewer::display.brewer.all()



**Value**

List of palette colors

**Examples**

```
paletteColorBrewerOrRd()
```

---

`paletteColorBrewerPaired`  
*paletteColorBrewerPaired Qualitative*

---

**Description**

Generate a qualitative Paired Brewer palette of a given size

**Usage**

```
paletteColorBrewerPaired(value.count = 3)
```

**Arguments**

`value.count`      Number of colors to generate; min is 3 (default); max is 12. See `RColorBrewer::display.brewer.all()`

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerPaired()
```

---

`paletteColorBrewerPastel1`  
*paletteColorBrewerPastel1 Qualitative*

---

**Description**

Generate a qualitative Pastel1 Brewer palette of a given size

**Usage**

```
paletteColorBrewerPastel1(value.count = 3)
```

**Arguments**

`value.count`      Number of colors to generate; min is 3 (default); max is 9 . See `RColorBrewer::display.brewer.all()`

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerPastel1()
```

---

```
paletteColorBrewerPastel2  
                          paletteColorBrewerPastel2 Qualitative
```

---

**Description**

Generate a qualitative Pastel2 Brewer palette of a given size

**Usage**

```
paletteColorBrewerPastel2(value.count = 3)
```

**Arguments**

`value.count`      Number of colors to generate; min is 3 (default); max is 8 . See `RColorBrewer::display.brewer.all()`

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerPastel2()
```

---

`paletteColorBrewerPiYG`*paletteColorBrewerPiYG Divergent*

---

**Description**

Generate a divergent PiYG Brewer palette of a given size

**Usage**

```
paletteColorBrewerPiYG(value.count = 3)
```

**Arguments**

`value.count` Number of colors to generate; min is 3 (default); max is 9 . See `RColorBrewer::display.brewer.all()`

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerPiYG()
```

---

`paletteColorBrewerPRGn`*paletteColorBrewerPRGn Divergent*

---

**Description**

Generate a divergent PRGn Brewer palette of a given size

**Usage**

```
paletteColorBrewerPRGn(value.count = 3)
```

**Arguments**

`value.count` Number of colors to generate; min is 3 (default); max is 9 . See `RColorBrewer::display.brewer.all()`

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerPRGn()
```

---

```
paletteColorBrewerPuBu
```

```
paletteColorBrewerPuBu Sequential
```

---

**Description**

Generate a sequential PuBu Brewer palette of a given size

**Usage**

```
paletteColorBrewerPuBu(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 9 . See RColorBrewer::display.brewer.all()

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerPuBu()
```

---

```
paletteColorBrewerPuBuGn
```

```
paletteColorBrewerPuBuGn Sequential
```

---

**Description**

Generate a sequential PuBuGn Brewer palette of a given size

**Usage**

```
paletteColorBrewerPuBuGn(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 9 . See RColorBrewer::display.brewer.all()

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerPuBuGn()
```

---

```
paletteColorBrewerPuOr  
paletteColorBrewerPuOr Divergent
```

---

**Description**

Generate a divergent PuOr Brewer palette of a given size

**Usage**

```
paletteColorBrewerPuOr(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 9 . See RColorBrewer::display.brewer.all()

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerPuOr()
```

---

```
paletteColorBrewerPuRd  
paletteColorBrewerPuRd Sequential
```

---

**Description**

Generate a sequential PuRd Brewer palette of a given size

**Usage**

```
paletteColorBrewerPuRd(value.count = 3)
```

**Arguments**

`value.count`      Number of colors to generate; min is 3 (default); max is 9 . See `RColorBrewer::display.brewer.all()`

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerPuRd()
```

---

```
paletteColorBrewerPurples  
  paletteColorBrewerPurples Sequential
```

---

**Description**

Generate a sequential Purples Brewer palette of a given size

**Usage**

```
paletteColorBrewerPurples(value.count = 3)
```

**Arguments**

`value.count`      Number of colors to generate; min is 3 (default); max is 9 . See `RColorBrewer::display.brewer.all()`

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerPurples()
```

---

`paletteColorBrewerRdBu`*paletteColorBrewerRdBu Divergent*

---

**Description**

Generate a divergent RdBu Brewer palette of a given size

**Usage**

```
paletteColorBrewerRdBu(value.count = 3)
```

**Arguments**

`value.count`      Number of colors to generate; min is 3 (default); max is 9 . See `RColorBrewer::display.brewer.all()`

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerRdBu()
```

---

`paletteColorBrewerRdPu`*paletteColorBrewerRdPu Sequential*

---

**Description**

Generate a sequential RdPu Brewer palette of a given size

**Usage**

```
paletteColorBrewerRdPu(value.count = 3)
```

**Arguments**

`value.count`      Number of colors to generate; min is 3 (default); max is 9 . See `RColorBrewer::display.brewer.all()`

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerRdPu()
```

---

```
paletteColorBrewerRdYlBu
```

*paletteColorBrewerRdYlBu Divergent*

---

**Description**

Generate a divergent RdYlBu Brewer palette of a given size

**Usage**

```
paletteColorBrewerRdYlBu(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 9 . See RColorBrewer::display.brewer.all()

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerRdYlBu()
```

---

```
paletteColorBrewerReds
```

*paletteColorBrewerReds Sequential*

---

**Description**

Generate a sequential Reds Brewer palette of a given size

**Usage**

```
paletteColorBrewerReds(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 9 . See RColorBrewer::display.brewer.all()



**Value**

List of palette colors

**Examples**

```
paletteColorBrewerReds()
```

---

```
paletteColorBrewerSet1  
    paletteColorBrewerSet1 Qualitative
```

---

**Description**

Generate a qualitative Set1 Brewer palette of a given size

**Usage**

```
paletteColorBrewerSet1(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 9 . See RColorBrewer::display.brewer.all()

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerSet1()
```

---

```
paletteColorBrewerSet2  
    paletteColorBrewerSet2 Qualitative
```

---

**Description**

Generate a qualitative Set2 Brewer palette of a given size

**Usage**

```
paletteColorBrewerSet2(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 8 . See RColorBrewer::display.brewer.all()

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerSet2()
```

---

```
paletteColorBrewerSet3
```

```
    paletteColorBrewerSet3 Qualitative
```

---

**Description**

Generate a qualitative Set3 Brewer palette of a given size

**Usage**

```
paletteColorBrewerSet3(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 12. See RColorBrewer::display.brewer.all()

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerSet3()
```

---

`paletteColorBrewerYlGn`*paletteColorBrewerYlGn Sequential*

---

**Description**

Generate a sequential YlGn Brewer palette of a given size

**Usage**

```
paletteColorBrewerYlGn(value.count = 3)
```

**Arguments**

`value.count`      Number of colors to generate; min is 3 (default); max is 9 . See `RColorBrewer::display.brewer.all()`

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerYlGn()
```

---

`paletteColorBrewerYlGnBu`*paletteColorBrewerYlGnBu Sequential*

---

**Description**

Generate a sequential YlGnBu Brewer palette of a given size

**Usage**

```
paletteColorBrewerYlGnBu(value.count = 3)
```

**Arguments**

`value.count`      Number of colors to generate; min is 3 (default); max is 9 . See `RColorBrewer::display.brewer.all()`

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerYlGnBu()
```

---

```
paletteColorBrewerYlOrBr
```

```
paletteColorBrewerYlOrBr Sequential
```

---

**Description**

Generate a sequential YlOrBr Brewer palette of a given size

**Usage**

```
paletteColorBrewerYlOrBr(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 9 . See RColorBrewer::display.brewer.all()

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerYlOrBr()
```

---

```
paletteColorBrewerYlOrRd
```

```
paletteColorBrewerYlOrRd Sequential
```

---

**Description**

Generate a sequential YlOrRd Brewer palette of a given size

**Usage**

```
paletteColorBrewerYlOrRd(value.count = 3)
```

**Arguments**

value.count      Number of colors to generate; min is 3 (default); max is 9 . See RColorBrewer::display.brewer.all()

**Value**

List of palette colors

**Examples**

```
paletteColorBrewerYlOrRd()
```

---

`paletteColorRandom`     *paletteColorRandom Qualitative*

---

**Description**

Generate a qualitative random color map of a given size

**Usage**

```
paletteColorRandom(value.count = 1)
```

**Arguments**

`value.count`     Number of colors to generate; default is 1

**Value**

List of random colors

**See Also**

`genNodeColorMap` `genEdgeColorMap`

**Examples**

```
paletteColorRandom()
```

---

`RCy3`     *RCy3: Functions to Access and Control Cytoscape*

---

**Description**

Vizualize, analyze and explore networks using Cytoscape via R.

**Details**

To learn more about RCy3, start with the vignettes: `browseVignettes("RCy3")`

---

RemoveFromGroup	<i>Remove from Group</i>
-----------------	--------------------------

---

### Description

Removes the specified nodes and edges from the specified group.

### Usage

```
RemoveFromGroup(
  group.name,
  nodes = NULL,
  nodes.by.col = "SUID",
  edges = NULL,
  edges.by.col = "SUID",
  network = NULL,
  base.url = .defaultBaseUrl
)
```

### Arguments

group.name	Specifies the name used to identify the group
nodes	List of node SUIDs, names, other column values, or keyword: selected, unselected or all. Default is currently selected nodes.
nodes.by.col	name of node table column corresponding to provided nodes list. Default is 'SUID'.
edges	List of edge SUIDs, names, other column values, or keyword: selected, unselected or all. Default is currently selected edges.
edges.by.col	name of edge table column corresponding to provided edges list. Default is 'SUID'.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

### Value

None

### Examples

```
RemoveFromGroup('myGroup')
```

---

`removeNodeCustomGraphics`*Remove Node Custom Graphics*

---

**Description**

Remove the default custom charts, images and gradients.

**Usage**

```
removeNodeCustomGraphics(  
  slot = 1,  
  style.name = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

<code>slot</code>	(optional) Which custom graphics slot to modify. Slots 1-9 are available for independent charts, gradients and images. Default is 1.
<code>style.name</code>	(optional) Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
removeNodeCustomGraphics()
```

---

`renameNetwork`*Rename a network*

---

**Description**

Sets a new name for this network

**Usage**

```
renameNetwork(title, network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

title	New name for the network
network	(optional) Name or suid of the network that you want to rename; default is "current" network
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Duplicate network names are not allowed

**Value**

None

**Author(s)**

Alexander Pico, Julia Gustavsen

**Examples**

```
renameNetwork("renamed network")
```

---

renameTableColumn	<i>Rename Table Column</i>
-------------------	----------------------------

---

**Description**

Sets a new name for a column.

**Usage**

```
renameTableColumn(  
  column,  
  new.name,  
  table = "node",  
  namespace = "default",  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```



**Arguments**

column	Name of the column to rename
new.name	New name for the specified column
table	(optional) Name of table, e.g., node (default), edge or network
namespace	(optional) Namespace of table, e.g., default (default), shared or hidden
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
renameTableColumn('exp', 'log2FC')
```

---

resetDefaultSandbox    *resetDefaultSandbox*

---

**Description**

Reset the entire state of the sandbox system.

**Usage**

```
resetDefaultSandbox()
```

**Value**

None

**Examples**

```
resetDefaultSandbox()
```

runningRemoteCheck     *runningRmoteCheck*

---

**Description**

runningRmoteCheck

**Usage**

```
runningRemoteCheck(newState = NULL)
```

**Arguments**

newState            new state of running remote

**Value**

oldState

**Examples**

```
runningRmoteCheck()
```

---

sandboxGetFileInfo     *sandboxGetFileInfo*

---

**Description**

Get metadata on file in sandbox (or entire sandbox). If the current sandbox is the entire file system on a Cytoscape workstation, trying to delete it is an error. Otherwise, deleting the current sandbox results in the default sandbox becoming the new current sandbox. When running standalone on the same workstation as Cytoscape, the default sandbox is the entire file system on the Cytoscape workstation. When running in a Notebook or remote server, the default sandbox is the 'default\_sandbox' created automatically under the filetransfer directory in the CytoscapeConfiguration directory. If that sandbox is deleted, it will be re-created so that subsequent file operations can complete successfully. Note that this function can be used to query either a file or a directory.

**Usage**

```
sandboxGetFileInfo(fileName, sandboxName = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

fileName	Name of file whose metadata to return ... can be sandbox-relative path ... “.” returns metadata on sandbox itself
sandboxName	Name of sandbox containing file. None means "the current sandbox".
base.url	Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://127.0.0.1:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

dict: 'filePath': <full path on Cytoscape workstation>, 'modifiedTime': <last changed time, " if file doesn't exist>, 'isFile': <True if file, False if directory>

**Examples**

```
sandboxGetFileInfo()
```

---

sandboxGetFrom	<i>sandboxGetFrom</i>
----------------	-----------------------

---

**Description**

Transfer a file from a sandbox. The source file is transferred from the named (or current) sandbox to the R workflow's file system, overwriting an existing file if one already exists. The sourceFile can be an absolute path if the sandbox is the entire file system (i.e., for standalone R execution) or a path relative to the sandbox (i.e., for Notebook or remote execution or if a sandbox was explicitly created).

**Usage**

```
sandboxGetFrom(
  sourceFile,
  destFile = NULL,
  overwrite = TRUE,
  sandboxName = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

sourceFile	Name of file to read (as absolute path or sandbox-relative path)
destFile	Name of file in the R workflow's file system ... if None, use file name in source_file
overwrite	Name of sandbox containing file. None means "the current sandbox".
sandboxName	Name of sandbox containing file. None means "the current sandbox".

`base.url` Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://127.0.0.1:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

`sandboxGetFrom`

**Examples**

`sandboxGetFrom()`

---

`sandboxInitializer`     *sandboxInitializer*

---

**Description**

Start with a sandbox template and update properties using whatever is found in the `new_sandbox`.

**Usage**

`sandboxInitializer(newSandbox = NULL, ...)`

**Arguments**

<code>newSandbox</code>	<code>newSandbox</code>
<code>...</code>	<code>...</code>

**Value**

`sandbox`

**Examples**

`sandboxInitializer()`

---

sandboxRemove	<i>sandboxRemove</i>
---------------	----------------------

---

### Description

Delete sandbox contents and remove its directory. If the current sandbox is the entire file system on a Cytoscape workstation, trying to delete it is an error. Otherwise, deleting the current sandbox results in the default sandbox becoming the new current sandbox. When running standalone on the same workstation as Cytoscape, the default sandbox is the entire file system on the Cytoscape workstation. When running in a Notebook or remote server, the default sandbox is the 'default\_sandbox' created automatically under the filetransfer directory in the CytoscapeConfiguration directory. If that sandbox is deleted, it will be re-created so that subsequent file operations can complete successfully.

### Usage

```
sandboxRemove(sandboxName = NULL, base.url = .defaultBaseUrl)
```

### Arguments

sandboxName	Name of sandbox to delete. None means to delete the current sandbox. If that sandbox is the default sandbox, it is automatically re-created.
base.url	Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://127.0.0.1:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Value

dict: 'sandboxPath': <directory on Cytoscape workstation>, 'existed': <True if sandbox existed>

### Examples

```
sandboxRemove()
```

---

sandboxRemoveFile	<i>sandboxRemoveFile</i>
-------------------	--------------------------

---

### Description

Remove a file from a sandbox. The named file is removed from the named sandbox. If the sandbox is the entire file system (i.e., for standalone R execution), the file name can be an absolute path. Otherwise, it is a path relative to the named sandbox. Note that there is no function that deletes a directory, except for `sandboxRemove()`, which deletes a sandbox and all of its contents.

**Usage**

```
sandboxRemoveFile(fileName, sandboxName = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

fileName	Name of file to delete (as absolute path or sandbox-relative path)
sandboxName	Name of sandbox containing file. None means "the current sandbox".
base.url	Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://127.0.0.1:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

dict: 'filePath': <file's absolute path in Cytoscape workstation>, 'existed': True if file existed before being deleted

**Examples**

```
sandboxRemoveFile()
```

---

sandboxSendTo	<i>sandboxSendTo</i>
---------------	----------------------

---

**Description**

Transfer a file to a sandbox. The source file is transferred to the named (or current) sandbox, overwriting an existing file if one already exists. The destFile can be an absolute path if the sandbox is the entire file system (i.e., for standalone R execution) or a path relative to the sandbox (i.e., for Notebook or remote execution or if a sandbox was explicitly created). Note that there is no function that transfers an entire directory. Note, though, that when using `sandboxSet()` to make a sandbox current, it is possible to copy the Cytoscape sample data directories into to the sandbox at the same time.

**Usage**

```
sandboxSendTo(
  sourceFile,
  destFile = NULL,
  overwrite = TRUE,
  sandboxName = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

sourceFile	Name of file to read (as absolute path or sandbox-relative path)
destFile	Name of file in the R workflow's file system ... if None, use file name in source_file
overwrite	Name of sandbox containing file. None means "the current sandbox".
sandboxName	Name of sandbox containing file. None means "the current sandbox".
base.url	Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://127.0.0.1:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

sandboxSendTo

**Examples**

```
sandboxSendTo()
```

---

sandboxSet

*sandboxSet*

---

**Description**

Set a new default sandbox, creating it if necessary. A sandbox is the root for the file system used for all file operations. When running standalone on the same workstation as Cytoscape, the default sandbox is the directory that's current for the R kernel. When running in a Notebook or remote server, the default sandbox is the 'default\_sandbox' created automatically under the filetransfer directory in the CytoscapeConfiguration directory. Naming a sandbox with this function creates a new sub-directory as a sibling to 'default\_sandbox' and uses it for subsequent file operations. Setting a None sandbox uses the default sandbox instead. Sandboxes are highly recommended as an aid to creating workflows that can be shared with others.

**Usage**

```
sandboxSet(  
  sandboxName,  
  copySamples = TRUE,  
  reinitialize = TRUE,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

sandboxName	Name of new default sandbox. None means to use the original default sandbox
copySamples	True to copy the Cytoscape sampleData into the sandbox
reinitialize	True to delete sandbox contents (if any) if sandbox already exists
base.url	Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://127.0.0.1:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

sandbox path in Cytoscape workstation's file system

**Examples**

```
sandboxSet()
```

---

sandboxUrlTo	<i>sandboxUrlTo</i>
--------------	---------------------

---

**Description**

Transfer a cloud-based file to a sandbox. The source URL identifies a file to be transferred to the named (or current) sandbox, overwriting an existing file if one already exists. The destFile can be an absolute path if the sandbox is the entire file system (i.e., for standalone R execution), or it can be a path relative to the sandbox (i.e., for Notebook or remote execution or if a sandbox was explicitly created).

**Usage**

```
sandboxUrlTo(
  sourceURL,
  destFile,
  overwrite = TRUE,
  sandboxName = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

sourceURL	URL addressing cloud file to download
destFile	Name of file in the R workflow's file system ... if None, use file name in source_file
overwrite	Name of sandbox containing file. None means "the current sandbox".
sandboxName	Name of sandbox containing file. None means "the current sandbox".
base.url	Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://127.0.0.1:1234 and the latest version of the CyREST API supported by this version of RCy3.



**Value**

dict: 'filePath': <new file's absolute path in Cytoscape workstation>, 'fileByteCount': number of bytes read

**Examples**

```
sandboxUrlTo()
```

---

 saveSession

*Save Session to File*


---

**Description**

Saves the current Cytoscape session as a CYS file.

**Usage**

```
saveSession(filename = NULL, base.url = .defaultBaseUrl, overwriteFile = TRUE)
```

**Arguments**

filename	Full path or path relative to current working directory, in addition to the name of the file. The .cys extension is automatically added. Leave blank to update previously saved session file.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.
overwriteFile	(optional) FALSE allows an error to be generated if the file already exists; TRUE allows Cytoscape to overwrite it without asking. Default value is TRUE.

**Details**

If no filename is provided, then it attempts to save to an existing CYS file associated with the session. If filename already exists, then it is overwritten.

Unlike most export functions in RCy3, Cytoscape will automatically overwrite CYS session files with the same name. You will not be prompted to confirm or reject overwrite. Use carefully!

**Value**

server response

**Examples**

```
saveSession('/fullpath/mySession')
saveSession()
```

selectAllEdges      *Select all edges*

---

**Description**

Selects all edges in a Cytoscape Network

**Usage**

```
selectAllEdges(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Selects all edges in a specified network.

**Author(s)**

Alexander Pico, Julia Gustavsen

**Examples**

```
cw <- CytoscapeWindow('new.demo', new('graphNEL'))
selectAllEdges(cw)
```

---

selectAllNodes      *Select all nodes*

---

**Description**

Selects all nodes in a Cytoscape Network

**Usage**

```
selectAllNodes(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network into which you want to select; default is "current" network
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Selects all nodes in a specified network.

**Author(s)**

Alexander Pico, Julia Gustavsen

**See Also**

[selectNodes](#)

**Examples**

```
selectAllNodes()
```

---

selectEdges

*Select Edges*

---

**Description**

Select edges in the network by SUID, name or other column values.

**Usage**

```
selectEdges(  
  edges,  
  by.col = "SUID",  
  preserve.current.selection = TRUE,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

<code>edges</code>	List of edge SUIDs, names or other column values
<code>by.col</code>	Edge table column to lookup up provide edge values. Default is 'SUID'.
<code>preserve.current.selection</code>	boolean Whether to maintain previously selected edges.
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

list of newly selected edge SUIDs

**Author(s)**

AlexanderPico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
selectEdges()
```

---

```
selectEdgesAdjacentToSelectedNodes
```

*Select Edges Adjacent To Selected Nodes*

---

**Description**

Takes currently selected nodes and adds to the selection all edges connected to those nodes, regardless of directionality.

**Usage**

```
selectEdgesAdjacentToSelectedNodes(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Lists of SUIDs for selected nodes and edges

**Examples**

```
selectEdgesAdjacentToSelectedNodes()
```

---

```
selectEdgesConnectingSelectedNodes
```

*Select the edges connecting selected nodes in Cytoscape Network*

---

**Description**

Selects edges in a Cytoscape Network connecting the selected nodes, including self loops connecting single nodes.

**Usage**

```
selectEdgesConnectingSelectedNodes(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Lists of SUIDs for selected nodes and edges

**Author(s)**

Alexander Pico, Julia Gustavsen

**Examples**

```
selectEdgesConnectingSelectedNodes()
```

---

selectFirstNeighbors    *Select first neighbor nodes*

---

### Description

Select nodes directly connected to currently selected nodes. Can specify connection directionality using the direction param.

### Usage

```
selectFirstNeighbors(
  direction = "any",
  network = NULL,
  base.url = .defaultBaseUrl
)
```

### Arguments

direction	direction of connections to neighbors to follow, e.g., incoming, outgoing, undirected, or any (default)
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

### Value

list of suids of selected nodes, including original selection

### Examples

```
selectFirstNeighbors()
selectFirstNeighbors('outgoing')
selectFirstNeighbors('incoming')
```

---

selectNodes    *Select Nodes*

---

### Description

Select nodes in the network by SUID, name or other column values.

**Usage**

```
selectNodes(  
  nodes,  
  by.col = "SUID",  
  preserve.current.selection = TRUE,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

nodes	List of node SUIDs, names or other column values
by.col	Node table column to lookup up provide node values. Default is 'SUID'.
preserve.current.selection	boolean Whether to maintain previously selected nodes.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

list of newly selected node SUIDs

**Author(s)**

AlexanderPico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

**Examples**

```
selectNodes()
```

---

```
selectNodesConnectedBySelectedEdges
```

*Select Nodes Connected By Selected Edges*

---

**Description**

Takes currently selected edges and extends the selection to connected nodes, regardless of directionality.

**Usage**

```
selectNodesConnectedBySelectedEdges(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

Lists of SUIDs for currently selected nodes and edges

**Examples**

```
selectNodesConnectedBySelectedEdges()
```

---

```
setBackgroundColorDefault
    Set Background Color Default
```

---

**Description**

Set the default background color.

**Usage**

```
setBackgroundColorDefault(
    new.color,
    style.name = NULL,
    base.url = .defaultBaseUrl
)
```

**Arguments**

new.color	Color as hex code, e.g., #FD5903
style.name	Name of style; default is "default" style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setBackgroundColorDefault('#888888')
```



---

setCatchupFilterSecs *Set Catchup Filter Delay*

---

**Description**

This function sets an internal delay variable that allows Cytoscape to "catchup" prior to subsequent functions. Call without specifying secs to restore default value.

**Usage**

```
setCatchupFilterSecs(secs = 1)
```

**Arguments**

secs                      Number of seconds to delay.

**Details**

This delay is only necessary while concurrency bugs exist in the Cytoscape application. This delay may need to be increased from the default value in certain use cases, e.g., larger networks.

**Value**

None

**See Also**

setModelPropagationSecs, setCatchupNetworkSecs

**Examples**

```
{
  setCatchupFilterSecs(2)
  setCatchupFilterSecs() #restores default delay
}
```

---

setCatchupNetworkSecs *Set Catchup Network Delay*

---

**Description**

This function sets an internal delay variable that allows Cytoscape to "catchup" prior to subsequent functions. Call without specifying secs to restore default value.

**Usage**

```
setCatchupNetworkSecs(secs = 2)
```

**Arguments**

secs                      Number of seconds to delay.

**Details**

This delay is only necessary while concurrency bugs exist in the Cytoscape application. This delay may need to be increased from the default value in certain use cases, e.g., larger networks.

**Value**

None

**See Also**

setModelPropagationSecs, setCatchupFilterSecs

**Examples**

```
{
  setCatchupNetworkSecs(2)
  setCatchupNetworkSecs() #restores default delay
}
```

---

setCurrentNetwork            *Set current network*

---

**Description**

Selects the given network as "current"

**Usage**

```
setCurrentNetwork(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network                    (optional) Name or suid of the network that you want set as current  
base.url                    (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

server response

**Examples**

```
setCurrentNetwork('MyNetwork')
```

---

setCurrentSandbox	<i>setCurrentSandbox</i>
-------------------	--------------------------

---

**Description**

Set and return the current sandbox name and path.

**Usage**

```
setCurrentSandbox(sandboxName, sandboxPath)
```

**Arguments**

sandboxName	sandboxName
sandboxPath	sandboxPath

**Value**

current sandbox

**Examples**

```
setCurrentSandbox()
```

---

setCurrentView	<i>Set Current View</i>
----------------	-------------------------

---

**Description**

Set which network view is "current".

**Usage**

```
setCurrentView(network = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

network	(optional) Name or SUID of a network or view. The first (presumably only) view associated a network is used if a specific view SUID is not provided.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Takes first (presumably only) view associated with provided network

**Value**

None

**Examples**

```
setCurrentView()
```

---

```
setDefaultSandbox    setDefaultSandbox
```

---

**Description**

Set and return the sandbox properties to be used as a default, probably based on whether a Notebook is running.

**Usage**

```
setDefaultSandbox(newSandbox = NULL, ...)
```

**Arguments**

<code>newSandbox</code>	<code>newSandbox</code>
<code>...</code>	<code>...</code>

**Value**

default sandbox

**Examples**

```
setDefaultSandbox()
```

---

```
setDefaultSandboxPath setDefaultSandboxPath
```

---

**Description**

Set and return the default path, which isn't one of the properties tracked in the `default_sandbox`.

**Usage**

```
setDefaultSandboxPath(newPath)
```

**Arguments**

newPath            new path of default sandbox

**Value**

default sandbox path

**Examples**

```
setDefaultSandboxPath()
```

---

setEdgeColorBypass     *Set Edge Color Bypass*

---

**Description**

Override the color for particular edges.

**Usage**

```
setEdgeColorBypass(
  edge.names,
  new.colors,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

edge.names	List of edge names or SUIDs
new.colors	List of hex colors, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

This method permanently overrides any default values or mappings defined for this visual property of the edge or edges specified. This method ultimately calls the generic function, [setEdgePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearEdgePropertyBypass](#).

**Value**

None

**See Also**

[setEdgePropertyBypass](#), [clearEdgePropertyBypass](#)

**Examples**

```
setEdgeColorBypass()
```

---

setEdgeColorDefault    *Set Edge Color Default*

---

**Description**

Set the default edge color.

**Usage**

```
setEdgeColorDefault(new.color, style.name = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>new.color</code>	Color as hex code, e.g., #FD5903
<code>style.name</code>	Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setEdgeColorDefault('#FD5903')
```

---

setEdgeColorMapping    *Set Edge Color Mapping*

---

### Description

Map table column values to colors to set the edge color.

### Usage

```
setEdgeColorMapping(
  table.column,
  table.column.values = NULL,
  colors = NULL,
  mapping.type = "c",
  default.color = NULL,
  style.name = NULL,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

### Arguments

table.column	Name of Cytoscape table column to map values from
table.column.values	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
colors	List of hex colors to map to table.column.values or a color palette function, e.g., paletteColorBrewerSet3 (without quotes). See RColorBrewer::display.brewer.all()
mapping.type	(char) continuous, discrete or passthrough (c,d,p); default is continuous
default.color	Hex color to set as default
style.name	Name of style; default is "default" style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

### Value

None

### Examples

```
setEdgeColorMapping('score', c(0,5), c('#FFFFFF', '#FF7755'))
setEdgeColorMapping('score', colors=paletteColorBrewerRdBu)
setEdgeColorMapping('score', colors=paletteColorBrewerSet3, mapping.type='d')
```

---

setEdgeFontFaceBypass *Set Edge Font Face Bypass*

---

### Description

Override the font face for particular edges.

### Usage

```
setEdgeFontFaceBypass(  
    edge.names,  
    new.fonts,  
    network = NULL,  
    base.url = .defaultBaseUrl  
)
```

### Arguments

edge.names	List of edge names or SUIDs
new.fonts	List of font faces, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method permanently overrides any default values or mappings defined for this visual property of the edge or edges specified. This method ultimately calls the generic function, [setEdgePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearEdgePropertyBypass](#).

### Value

None

### See Also

[setEdgePropertyBypass](#), [clearEdgePropertyBypass](#)

### Examples

```
setEdgeFontFaceBypass()
```



---

`setEdgeFontFaceDefault`*Set Edge Font Face Default*

---

**Description**

Set the default edge font.

**Usage**

```
setEdgeFontFaceDefault(new.font, style.name = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>new.font</code>	String specification of font face, style and size, e.g., "SansSerif,plain,12" or "Dialog,plain,10"
<code>style.name</code>	Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setEdgeFontFaceDefault("Dialog,plain,10")
```

---

`setEdgeFontFaceMapping`*Set Edge Font Face Mapping*

---

**Description**

Sets font face for edge labels.

**Usage**

```

setEdgeFontFaceMapping(
  table.column,
  table.column.values,
  fonts,
  mapping.type = "d",
  default.font = NULL,
  style.name = NULL,
  network = NULL,
  base.url = .defaultBaseUrl
)

```

**Arguments**

<code>table.column</code>	Name of Cytoscape table column to map values from
<code>table.column.values</code>	List of values from Cytoscape table to be used in mapping
<code>fonts</code>	List of string specifications of font face, style and size, e.g., <code>c("SansSerif,plain,12", "Dialog,plain,10")</code>
<code>mapping.type</code>	(char) discrete or passthrough (d,p); default is discrete
<code>default.font</code>	String specification of font face, style and size, e.g., <code>"SansSerif,plain,12"</code> or <code>"Dialog,plain,10"</code>
<code>style.name</code>	Name of style; default is "default" style
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```

setEdgeFontFaceMapping("myfonts", c("normal", "small"),
  c("SansSerif,plain,12", "Dialog,plain,10"))

```

---

setEdgeFontSizeBypass *Set Edge Font Size Bypass*

---

### Description

Override the font size for particular edges.

### Usage

```
setEdgeFontSizeBypass(  
    edge.names,  
    new.sizes,  
    network = NULL,  
    base.url = .defaultBaseUrl  
)
```

### Arguments

edge.names	List of edge names or SUIDs
new.sizes	List of size values, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method permanently overrides any default values or mappings defined for this visual property of the edge or edges specified. This method ultimately calls the generic function, [setEdgePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearEdgePropertyBypass](#).

### Value

None

### See Also

[setEdgePropertyBypass](#), [clearEdgePropertyBypass](#)

### Examples

```
setEdgeFontSizeBypass()
```

---

`setEdgeFontSizeDefault`*Set Edge Font Size Default*

---

**Description**

Set the default edge font size.

**Usage**

```
setEdgeFontSizeDefault(new.size, style.name = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>new.size</code>	Numeric value for size
<code>style.name</code>	Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setEdgeFontSizeDefault(12)
```

---

`setEdgeFontSizeMapping`*Set Edge Font Size Mapping*

---

**Description**

Map table column values to sizes to set the edge size.

**Usage**

```

setEdgeFontSizeMapping(
  table.column,
  table.column.values = NULL,
  sizes = NULL,
  mapping.type = "c",
  default.size = NULL,
  style.name = NULL,
  network = NULL,
  base.url = .defaultBaseUrl
)

```

**Arguments**

table.column	Name of Cytoscape table column to map values from
table.column.values	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
sizes	List of sizes to map to table.column.values. A range of 10 to 100 is used by default for automatic mapping.
mapping.type	(char) continuous, discrete or passthrough (c,d,p); default is continuous
default.size	Size value to set as default
style.name	Name of style; default is "default" style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```

setEdgeFontSizeMapping('score')
setEdgeFontSizeMapping('score', sizes=c(6,24))
setEdgeFontSizeMapping('score', c(0,30), c(35,55))

```

---

setEdgeLabelBypass      *Set Edge Label Bypass*

---

### Description

Override the label for particular edges.

### Usage

```
setEdgeLabelBypass(  
    edge.names,  
    new.labels,  
    network = NULL,  
    base.url = .defaultBaseUrl  
)
```

### Arguments

edge.names	List of edge names or SUIDs
new.labels	List of labels, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method permanently overrides any default values or mappings defined for this visual property of the edge or edges specified. This method ultimately calls the generic function, [setEdgePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearEdgePropertyBypass](#).

### Value

None

### See Also

[setEdgePropertyBypass](#), [clearEdgePropertyBypass](#)

### Examples

```
setEdgeLabelBypass()
```

---

`setEdgeLabelColorBypass`*Set Edge Label Color Bypass*

---

## Description

Override the label color for particular edges.

## Usage

```
setEdgeLabelColorBypass(  
  edge.names,  
  new.colors,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

## Arguments

<code>edge.names</code>	List of edge names or SUIDs
<code>new.colors</code>	List of hex colors, or single value
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

## Details

This method permanently overrides any default values or mappings defined for this visual property of the edge or edges specified. This method ultimately calls the generic function, [setEdgePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearEdgePropertyBypass](#).

## Value

None

## See Also

[setEdgePropertyBypass](#), [clearEdgePropertyBypass](#)

## Examples

```
setEdgeLabelColorBypass()
```

setEdgeLabelColorDefault

*Set Edge Label Color Default*

---

### Description

Set the default edge label color.

### Usage

```
setEdgeLabelColorDefault(  
    new.color,  
    style.name = NULL,  
    base.url = .defaultBaseUrl  
)
```

### Arguments

new.color	Color as hex code, e.g., #FD5903
style.name	Name of style; default is "default" style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

### Value

None

### Examples

```
setEdgeLabelColorDefault("#FD5903")
```

---

setEdgeLabelColorMapping

*Set Edge Label Color Mapping*

---

### Description

Map table column values to colors to set the edge border color.



**Usage**

```
setEdgeLabelColorMapping(
  table.column,
  table.column.values = NULL,
  colors = NULL,
  mapping.type = "c",
  default.color = NULL,
  style.name = NULL,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

<code>table.column</code>	Name of Cytoscape table column to map values from
<code>table.column.values</code>	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
<code>colors</code>	List of hex colors to map to <code>table.column.values</code> or a color palette function, e.g., <code>paletteColorBrewerSet3</code> (without quotes). See <code>RColorBrewer::display.brewer.all()</code>
<code>mapping.type</code>	(char) continuous, discrete or passthrough (c,d,p); default is continuous
<code>default.color</code>	Hex color to set as default
<code>style.name</code>	Name of style; default is "default" style
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setEdgeLabelColorMapping('score', c(0,5), c('#FFFFFF', '#FF7755'))
setEdgeLabelColorMapping('score', colors=paletteColorBrewerRdBu)
setEdgeLabelColorMapping('score', colors=paletteColorBrewerSet3, mapping.type='d')
```

---

setEdgeLabelDefault    *Set Edge Label Default*

---

**Description**

Set the default edge label.

**Usage**

```
setEdgeLabelDefault(new.label, style.name = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

new.label	String label for unmapped edges.
style.name	Name of style; default is "default" style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setEdgeLabelDefault('unknown')
```

---

setEdgeLabelMapping    *Set Edge Label Mapping*

---

**Description**

Pass the values from a table column to display as edge labels.

**Usage**

```
setEdgeLabelMapping(  
  table.column,  
  style.name = NULL,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

table.column	Name of Cytoscape table column to map values from
style.name	Name of style; default is "default" style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setEdgeLabelMapping('label')
```

---

```
setEdgeLabelOpacityBypass
```

*Set Edge Label Opacity Bypass*

---

**Description**

Override the label opacity for particular edges.

**Usage**

```
setEdgeLabelOpacityBypass(
  edge.names,
  new.value,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

edge.names	List of edge names or SUIDs
new.value	List of opacity values, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

## Details

This method permanently overrides any default values or mappings defined for this visual property of the edge or edges specified. This method ultimately calls the generic function, [setEdgePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearEdgePropertyBypass](#).

## Value

None

## See Also

[setEdgePropertyBypass](#), [clearEdgePropertyBypass](#)

## Examples

```
setEdgeLabelOpacityBypass()
```

---

setEdgeLabelOpacityDefault  
*Set Edge Label Opacity Default*

---

## Description

Set default opacity value for all unmapped edge labels.

## Usage

```
setEdgeLabelOpacityDefault(  
    new.opacity,  
    style.name = NULL,  
    base.url = .defaultBaseUrl  
)
```

## Arguments

<code>new.opacity</code>	Numeric values between 0 and 255; 0 is invisible.
<code>style.name</code>	Name of style; default is "default" style.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

## Value

None

**Examples**

```
setEdgeLabelOpacityDefault(50)
```

---

```
setEdgeLabelOpacityMapping
    Set Edge Label Opacity Mapping
```

---

**Description**

Sets opacity for edge label only.

**Usage**

```
setEdgeLabelOpacityMapping(
    table.column,
    table.column.values = NULL,
    opacities = NULL,
    mapping.type = "c",
    default.opacity = NULL,
    style.name = NULL,
    network = NULL,
    base.url = .defaultBaseUrl
)
```

**Arguments**

table.column	Name of Cytoscape table column to map values from
table.column.values	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
opacities	(integer) values between 0 and 255; 0 is invisible. A range of 50 to 255 is used by default for automatic mapping.
mapping.type	(char) continuous, discrete or passthrough (c,d,p); default is continuous
default.opacity	Opacity value to set as default for all unmapped values
style.name	Name of style; default is "default" style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

## Examples

```
setEdgeLabelOpacityMapping('weight')
setEdgeLabelOpacityMapping('weight', opacities=c(0,100))
setEdgeLabelOpacityMapping('weight', c(1,10), c(50,255))
```

---

setEdgeLineStyleBypass

*Set Edge Line Style Bypass*

---

## Description

Override the style for particular edges.

## Usage

```
setEdgeLineStyleBypass(
  edge.names,
  new.styles,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

## Arguments

edge.names	List of edge names or SUIDs
new.styles	List of style values, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

## Details

This method permanently overrides any default values or mappings defined for this visual property of the edge or edges specified. This method ultimately calls the generic function, [setEdgePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearEdgePropertyBypass](#).

## Value

None

## See Also

[setEdgePropertyBypass](#), [clearEdgePropertyBypass](#)

**Examples**

```
setEdgeLineStyleBypass()
```

---

```
setEdgeLineStyleDefault
```

*Set Edge Line Style Default*

---

**Description**

Set the default edge style.

**Usage**

```
setEdgeLineStyleDefault(  
    new.line.style,  
    style.name = NULL,  
    base.url = .defaultBaseUrl  
)
```

**Arguments**

`new.line.style` Name of line style, e.g., SOLID, LONG\_DASH, etc (see [getLineStylees](#))

`style.name` Name of style; default is "default" style

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setEdgeLineStyleDefault('LONG_DASH')
```

---

```
setEdgeLineStyleMapping
```

*Set Edge Line Style Mapping*

---

### Description

Map table column values to styles to set the edge style.

### Usage

```
setEdgeLineStyleMapping(
    table.column,
    table.column.values = NULL,
    line.styles = NULL,
    default.line.style = "SOLID",
    style.name = NULL,
    network = NULL,
    base.url = .defaultBaseUrl
)
```

### Arguments

<code>table.column</code>	Name of Cytoscape table column to map values from
<code>table.column.values</code>	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
<code>line.styles</code>	List of line styles. Leave NULL to perform an automatic mapping to available line styles. See <a href="#">getLineStyle</a> .
<code>default.line.style</code>	Style to set as default. See <a href="#">getLineStyle</a> .
<code>style.name</code>	Name of style; default is "default" style
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Value

None

### Examples

```
setEdgeLineStyleMapping('type')
setEdgeLineStyleMapping('type',c('pp','pd'),c('SOLID','LONG_DASH'))
```



---

`setEdgeLineWidthBypass`*Set Edge Line Width Bypass*

---

## Description

Override the width for particular edges.

## Usage

```
setEdgeLineWidthBypass(  
    edge.names,  
    new.widths,  
    network = NULL,  
    base.url = .defaultBaseUrl  
)
```

## Arguments

<code>edge.names</code>	List of edge names or SUIDs
<code>new.widths</code>	List of width values, or single value
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

## Details

This method permanently overrides any default values or mappings defined for this visual property of the edge or edges specified. This method ultimately calls the generic function, [setEdgePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearEdgePropertyBypass](#).

## Value

None

## See Also

[setEdgePropertyBypass](#), [clearEdgePropertyBypass](#)

## Examples

```
setEdgeLineWidthBypass()
```

setEdgeLineWidthDefault

*Set Edge Line Width Default*

---

### Description

Set the default edge width.

### Usage

```
setEdgeLineWidthDefault(  
    new.width,  
    style.name = NULL,  
    base.url = .defaultBaseUrl  
)
```

### Arguments

new.width	Numeric value for width
style.name	Name of style; default is "default" style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

### Value

None

### Examples

```
setEdgeLineWidthDefault(3)
```

---

setEdgeLineWidthMapping

*Set Edge Line Width Mapping*

---

### Description

Map table column values to widths to set the edge line width.

**Usage**

```

setEdgeLineWidthMapping(
  table.column,
  table.column.values = NULL,
  widths = NULL,
  mapping.type = "c",
  default.width = NULL,
  style.name = NULL,
  network = NULL,
  base.url = .defaultBaseUrl
)

```

**Arguments**

table.column	Name of Cytoscape table column to map values from
table.column.values	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
widths	List of width values to map to table.column.values. A range of 10 to 100 is used by default for automatic mapping.
mapping.type	(char) continuous, discrete or passthrough (c,d,p); default is continuous
default.width	Width value to set as default for all unmapped values for all unmapped values.
style.name	Name of style; default is "default" style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```

setEdgeLineWidthMapping('score')
setEdgeLineWidthMapping('score', widths=c(1,10))
setEdgeLineWidthMapping('score', c(0,30), c(1,5))

```

---

setEdgeOpacityBypass *Set Edge Opacity Bypass*

---

### Description

Override the opacity for particular edges.

### Usage

```
setEdgeOpacityBypass(  
    edge.names,  
    new.values,  
    network = NULL,  
    base.url = .defaultBaseUrl  
)
```

### Arguments

edge.names	List of edge names or SUIDs
new.values	List of values to set, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method permanently overrides any default values or mappings defined for this visual property of the edge or edges specified. This method ultimately calls the generic function, [setEdgePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearEdgePropertyBypass](#).

### Value

None

### See Also

[setEdgePropertyBypass](#), [clearEdgePropertyBypass](#)

### Examples

```
setEdgeOpacityBypass()
```

---

setEdgeOpacityDefault *Set Edge Opacity Default*

---

### Description

Set default opacity value for all unmapped edges.

### Usage

```
setEdgeOpacityDefault(  
    new.opacity,  
    style.name = NULL,  
    base.url = .defaultBaseUrl  
)
```

### Arguments

new.opacity	Numeric values between 0 and 255; 0 is invisible.
style.name	Name of style; default is "default" style.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

### Value

None

### Examples

```
setEdgeOpacityDefault(50)
```

---

setEdgeOpacityMapping *Set Edge Opacity Mapping*

---

### Description

Map table column values to opacities to set the edge opacity.

**Usage**

```

setEdgeOpacityMapping(
  table.column,
  table.column.values = NULL,
  opacities = NULL,
  mapping.type = "c",
  default.opacity = NULL,
  style.name = NULL,
  network = NULL,
  base.url = .defaultBaseUrl
)

```

**Arguments**

<code>table.column</code>	Name of Cytoscape table column to map values from
<code>table.column.values</code>	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
<code>opacities</code>	(integer) values between 0 and 255; 0 is invisible. A range of 50 to 255 is used by default for automatic mapping.
<code>mapping.type</code>	(char) continuous, discrete or passthrough (c,d,p); default is continuous
<code>default.opacity</code>	Opacity value to set as default for all unmapped values
<code>style.name</code>	Name of style; default is "default" style
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```

setEdgeOpacityMapping('weight')
setEdgeOpacityMapping('weight', opacities=c(0,100))
setEdgeOpacityMapping('weight', c(1,10), c(50,255))

```

---

setEdgePropertyBypass *Set Edge Property Bypass*

---

### Description

Set bypass values for any edge property of the specified edges, overriding default values and mappings defined by any visual style.

### Usage

```
setEdgePropertyBypass(  
  edge.names,  
  new.values,  
  visual.property,  
  bypass = TRUE,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

edge.names	List of edge names or SUIDs
new.values	List of values to set, or single value
visual.property	Name of a visual property. See <a href="#">getVisualPropertyNames</a> .
bypass	Whether to set permanent bypass value. Default is TRUE.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method permanently overrides any default values or mappings defined for the visual properties of the edge or edges specified. To restore defaults and mappings, use [clearEdgePropertyBypass](#).

### Value

None

### See Also

[clearEdgePropertyBypass](#)

## Examples

```
setEdgePropertyBypass()
```

---

```
setEdgeSelectionColorDefault  
Set Edge Selection Color Default
```

---

## Description

Set the default selected edge color.

## Usage

```
setEdgeSelectionColorDefault(  
    new.color,  
    style.name = NULL,  
    base.url = .defaultBaseUrl  
)
```

## Arguments

<code>new.color</code>	Color as hex code, e.g., #FD5903
<code>style.name</code>	Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

## Value

None

## Examples

```
setEdgeSelectionColorDefault('#FD5903')
```



---

setEdgeSourceArrowColorBypass  
*Set Edge Source Arrow Color Bypass*

---

## Description

Override the source arrow color for particular edges.

## Usage

```
setEdgeSourceArrowColorBypass(  
    edge.names,  
    new.colors,  
    network = NULL,  
    base.url = .defaultBaseUrl  
)
```

## Arguments

edge.names	List of edge names or SUIDs
new.colors	List of hex colors, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

## Details

This method permanently overrides any default values or mappings defined for this visual property of the edge or edges specified. This method ultimately calls the generic function, [setEdgePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearEdgePropertyBypass](#).

## Value

None

## See Also

[setEdgePropertyBypass](#), [clearEdgePropertyBypass](#)

## Examples

```
setEdgeSourceArrowColorBypass()
```

setEdgeSourceArrowColorDefault

*Set Edge Source Arrow Color Default*

---

### Description

Set the default edge source arrow color.

### Usage

```
setEdgeSourceArrowColorDefault(  
  new.color,  
  style.name = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

<code>new.color</code>	Color as hex code, e.g., #FD5903
<code>style.name</code>	Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Value

None

### Examples

```
setEdgeSourceArrowColorDefault('#FD5903')
```

---

setEdgeSourceArrowColorMapping

*Set Edge Source Arrow Color Mapping*

---

### Description

Map table column values to colors to set the source arrow color.

**Usage**

```
setEdgeSourceArrowColorMapping(
  table.column,
  table.column.values = NULL,
  colors = NULL,
  mapping.type = "c",
  default.color = NULL,
  style.name = NULL,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

<code>table.column</code>	Name of Cytoscape table column to map values from
<code>table.column.values</code>	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
<code>colors</code>	List of hex colors to map to <code>table.column.values</code> or a color palette function, e.g., <code>paletteColorBrewerSet3</code> (without quotes). See <code>RColorBrewer::display.brewer.all()</code>
<code>mapping.type</code>	(char) continuous, discrete or passthrough (c,d,p); default is continuous
<code>default.color</code>	Hex color to set as default
<code>style.name</code>	Name of style; default is "default" style
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setEdgeSourceArrowColorMapping('score', c(0,5), c('#FFFFFF', '#FF7755'))
setEdgeSourceArrowColorMapping('score', colors=paletteColorBrewerRdBu)
setEdgeSourceArrowColorMapping('score', colors=paletteColorBrewerSet3, mapping.type='d')
```

---

```
setEdgeSourceArrowMapping
```

*Set Edge Source Arrow Mapping*

---

### Description

Map table column values to shapes to set the source arrow shape.

### Usage

```
setEdgeSourceArrowMapping(  
  table.column,  
  table.column.values = NULL,  
  shapes = NULL,  
  default.shape = "ARROW",  
  style.name = NULL,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

<code>table.column</code>	Name of Cytoscape table column to map values from
<code>table.column.values</code>	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
<code>shapes</code>	List of shapes to map to <code>table.column.values</code> . Leave NULL to perform an automatic mapping to available shapes. See <a href="#">getArrowShapes</a>
<code>default.shape</code>	Shape to set as default. See <a href="#">getArrowShapes</a>
<code>style.name</code>	Name of style; default is "default" style
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Value

None

### Examples

```
setEdgeSourceArrowMapping('type')  
setEdgeSourceArrowMapping('type',c('activation','inhibition'),c('ARROW','T'))
```

---

setEdgeSourceArrowShapeBypass  
*Set Edge Source Arrow Shape Bypass*

---

## Description

Override the source arrow shape for particular edges.

## Usage

```
setEdgeSourceArrowShapeBypass(  
  edge.names,  
  new.shapes,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

## Arguments

edge.names	List of edge names or SUIDs
new.shapes	List of shapes, or single value. See <a href="#">getArrowShapes</a> .
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

## Details

This method permanently overrides any default values or mappings defined for this visual property of the edge or edges specified. This method ultimately calls the generic function, [setEdgePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearEdgePropertyBypass](#).

## Value

None

## See Also

[setEdgePropertyBypass](#), [clearEdgePropertyBypass](#)

## Examples

```
setEdgeSourceArrowShapeBypass()
```

---

`setEdgeSourceArrowShapeDefault`*Set Edge Source Arrow Shape Default*

---

**Description**

Set the default edge source arrow shape.

**Usage**

```
setEdgeSourceArrowShapeDefault(  
    new.shape,  
    style.name = NULL,  
    base.url = .defaultBaseUrl  
)
```

**Arguments**

<code>new.shape</code>	Name of shape, e.g., ARROW, T, etc (see <a href="#">getArrowShapes</a> )
<code>style.name</code>	Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setEdgeSourceArrowShapeDefault('ARROW')
```

---

`setEdgeSourceArrowShapeMapping`*Set Edge Source Arrow Shape Mapping*

---

**Description**

Map table column values to shapes to set the source arrow shape.

**Usage**

```
setEdgeSourceArrowShapeMapping(  
  table.column,  
  table.column.values = NULL,  
  shapes = NULL,  
  default.shape = NULL,  
  style.name = NULL,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

table.column	Name of Cytoscape table column to map values from
table.column.values	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
shapes	List of shapes to map to table.column.values. Leave NULL to perform an automatic mapping to available shapes. See <a href="#">getArrowShapes</a>
default.shape	Shape to set as default. See <a href="#">getArrowShapes</a>
style.name	Name of style; default is "default" style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setEdgeSourceArrowShapeMapping('type')  
setEdgeSourceArrowShapeMapping('type',c('activation','inhibition'),  
  c('ARROW','T'))
```

---

setEdgeTargetArrowColorBypass

*Set Edge Target Arrow Color Bypass*

---

**Description**

Override the target arrow color for particular edges.

**Usage**

```
setEdgeTargetArrowColorBypass(  
    edge.names,  
    new.colors,  
    network = NULL,  
    base.url = .defaultBaseUrl  
)
```

**Arguments**

edge.names	List of edge names or SUIDs
new.colors	List of hex colors, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Details**

This method permanently overrides any default values or mappings defined for this visual property of the edge or edges specified. This method ultimately calls the generic function, [setEdgePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearEdgePropertyBypass](#).

**Value**

None

**See Also**

[setEdgePropertyBypass](#), [clearEdgePropertyBypass](#)

**Examples**

```
setEdgeTargetArrowColorBypass()
```

---

setEdgeTargetArrowColorDefault

*Set Edge Target Arrow Color Default*

---

**Description**

Set the default edge target arrow color.



**Usage**

```
setEdgeTargetArrowColorDefault(  
    new.color,  
    style.name = NULL,  
    base.url = .defaultBaseUrl  
)
```

**Arguments**

new.color	Color as hex code, e.g., #FD5903
style.name	Name of style; default is "default" style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setEdgeTargetArrowColorDefault('#FD5903')
```

---

```
setEdgeTargetArrowColorMapping  
    Set Edge Target Arrow Color Mapping
```

---

**Description**

Map table column values to colors to set the target arrow color.

**Usage**

```
setEdgeTargetArrowColorMapping(  
    table.column,  
    table.column.values = NULL,  
    colors = NULL,  
    mapping.type = "c",  
    default.color = NULL,  
    style.name = NULL,  
    network = NULL,  
    base.url = .defaultBaseUrl  
)
```

**Arguments**

table.column	Name of Cytoscape table column to map values from
table.column.values	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
colors	List of hex colors to map to table.column.values or a color palette function, e.g., paletteColorBrewerSet3 (without quotes). See RColorBrewer::display.brewer.all()
mapping.type	(char) continuous, discrete or passthrough (c,d,p); default is continuous
default.color	Hex color to set as default
style.name	Name of style; default is "default" style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setEdgeTargetArrowColorMapping('score', c(0,5), c('#FFFFFF', '#FF7755'))
setEdgeTargetArrowColorMapping('score', colors=paletteColorBrewerRdBu)
setEdgeTargetArrowColorMapping('score', colors=paletteColorBrewerSet3, mapping.type='d')
```

---

```
setEdgeTargetArrowMapping
```

*Set Edge Target Arrow Mapping*

---

**Description**

Map table column values to shapes to set the target arrow shape.

**Usage**

```
setEdgeTargetArrowMapping(
  table.column,
  table.column.values = NULL,
  shapes = NULL,
  default.shape = "ARROW",
  style.name = NULL,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

### Arguments

<code>table.column</code>	Name of Cytoscape table column to map values from
<code>table.column.values</code>	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
<code>shapes</code>	List of shapes to map to <code>table.column.values</code> . Leave NULL to perform an automatic mapping to available shapes. See <a href="#">getArrowShapes</a>
<code>default.shape</code>	Shape to set as default. See <a href="#">getArrowShapes</a>
<code>style.name</code>	Name of style; default is "default" style
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Value

None

### Examples

```
setEdgeTargetArrowMapping('type')
setEdgeTargetArrowMapping('type',c('activation','inhibition'),c('ARROW','T'))
```

---

setEdgeTargetArrowShapeBypass

*Set Edge Target Arrow Shape Bypass*

---

### Description

Override the target arrow shape for particular edges.

### Usage

```
setEdgeTargetArrowShapeBypass(
  edge.names,
  new.shapes,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

### Arguments

edge.names	List of edge names or SUIDs
new.shapes	List of values to set, or single value. See <a href="#">getArrowShapes</a> .
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method permanently overrides any default values or mappings defined for this visual property of the edge or edges specified. This method ultimately calls the generic function, [setEdgePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearEdgePropertyBypass](#).

### Value

None

### See Also

[setEdgePropertyBypass](#), [clearEdgePropertyBypass](#)

### Examples

```
setEdgeTargetArrowShapeBypass()
```

---

```
setEdgeTargetArrowShapeDefault  
Set Edge Target Arrow Shape Default
```

---

### Description

Set the default edge target arrow shape.

### Usage

```
setEdgeTargetArrowShapeDefault(  
  new.shape,  
  style.name = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

<code>new.shape</code>	Name of shape, e.g., ARROW, T, etc (see <a href="#">getArrowShapes</a> )
<code>style.name</code>	Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setEdgeTargetArrowShapeDefault('ARROW')
```

---

```
setEdgeTargetArrowShapeMapping
```

*Set Edge Target Arrow Shape Mapping*

---

**Description**

Map table column values to shapes to set the target arrow shape.

**Usage**

```
setEdgeTargetArrowShapeMapping(
  table.column,
  table.column.values = NULL,
  shapes = NULL,
  default.shape = NULL,
  style.name = NULL,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

<code>table.column</code>	Name of Cytoscape table column to map values from
<code>table.column.values</code>	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
<code>shapes</code>	List of shapes to map to <code>table.column.values</code> . Leave NULL to perform an automatic mapping to available shapes. See <a href="#">getArrowShapes</a>
<code>default.shape</code>	Shape to set as default. See <a href="#">getArrowShapes</a>

style.name	Name of style; default is "default" style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setEdgeTargetArrowShapeMapping('type')
setEdgeTargetArrowShapeMapping('type',c('activation','inhibition'),
  c('ARROW','T'))
```

---

setEdgeTooltipBypass    *Set Edge Tooltip Bypass*

---

**Description**

Override the tooltip for particular edges.

**Usage**

```
setEdgeTooltipBypass(
  edge.names,
  new.values,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

edge.names	List of edge names or SUIDs
new.values	List of tooltip values, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

This method permanently overrides any default values or mappings defined for this visual property of the edge or edges specified. This method ultimately calls the generic function, [setEdgePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearEdgePropertyBypass](#).

**Value**

None

**See Also**

[setEdgePropertyBypass](#), [clearEdgePropertyBypass](#)

**Examples**

```
setEdgeTooltipBypass()
```

---

setEdgeTooltipDefault *Set Edge Tooltip Default*

---

**Description**

Set the default edge tooltip

**Usage**

```
setEdgeTooltipDefault(  
    new.tooltip,  
    style.name = NULL,  
    base.url = .defaultBaseUrl  
)
```

**Arguments**

<code>new.tooltip</code>	String tooltip for unmapped edges.
<code>style.name</code>	Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

## Examples

```
setEdgeTooltipDefault('unknown')
```

---

setEdgeTooltipMapping *Set Edge Tooltip Mapping*

---

## Description

Pass the values from a table column to display as edge tooltips.

## Usage

```
setEdgeTooltipMapping(  
  table.column,  
  style.name = NULL,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

## Arguments

table.column	Name of Cytoscape table column to map values from
style.name	Name of style; default is "default" style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

## Value

None

## Examples

```
setEdgeTooltipMapping('description')
```



---

setLayoutProperties    *Set Layout Properties*

---

### Description

Sets the specified properties for the specified layout. Unmentioned properties are left unchanged.

### Usage

```
setLayoutProperties(layout.name, properties.list, base.url = .defaultBaseUrl)
```

### Arguments

layout.name	(character) Name of the layout
properties.list	(list) List of one or more property=value pairs
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Details

Run [getLayoutNames](#) to list available layouts. Run [getLayoutPropertyNames](#) to list properties per layout.

### Value

None

### Author(s)

Alexander Pico, Tanja Muetze, Georgi Kolishovski, Paul Shannon

### Examples

```
setLayoutProperties('force-directed', list(defaultSpringLength=50, defaultSpringCoefficient=6E-04))  
# Successfully updated the property 'defaultSpringLength'.  
# Successfully updated the property 'defaultSpringCoefficient'.
```

---

`setModelPropagationSecs`*Set Model Propagation Delay*

---

**Description**

This function sets an internal delay variable that allows Cytoscape to "catchup" prior to subsequent functions. Call without specifying secs to restore default value.

**Usage**

```
setModelPropagationSecs(secs = 5)
```

**Arguments**

secs            Number of seconds to delay.

**Details**

This delay is only necessary while concurrency bugs exist in the Cytoscape application. This delay may need to be increased from the default value in certain use cases, e.g., larger networks.

**Value**

None

**See Also**

setCatchupFilterSecs, setCatchupNetworkSecs

**Examples**

```
{  
  setModelPropagationSecs(2)  
  setModelPropagationSecs() #restores default delay  
}
```

---

`setNetworkCenterBypass`*Set Network Center Bypass*

---

### Description

Set the bypass value for center x and y for the network. This function could be used to pan and scroll the Cytoscape canvas.

### Usage

```
setNetworkCenterBypass(  
  x,  
  y,  
  bypass = FALSE,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

x	Coordinate value, increases going to the right.
y	Coordinate value, increase going down.
bypass	Whether to set permanent bypass value. Default is FALSE per common use of temporary center settings.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method permanently overrides any default values for this visual property. This method ultimately calls the generic function, [setNetworkPropertyBypass](#), which can be used to set any visual property. To restore defaults, use [clearNetworkPropertyBypass](#).

### Value

None

### See Also

[setNetworkPropertyBypass](#), [clearNetworkPropertyBypass](#)

## Examples

```
setNetworkCenterBypass()
```

---

```
setNetworkPropertyBypass
```

*Set Network Property Bypass*

---

## Description

Set bypass values for any network property, overriding default values defined by any visual style.

## Usage

```
setNetworkPropertyBypass(  
    new.value,  
    visual.property,  
    bypass = TRUE,  
    network = NULL,  
    base.url = .defaultBaseUrl  
)
```

## Arguments

<code>new.value</code>	Value to set
<code>visual.property</code>	Name of a visual property. See <a href="#">getVisualPropertyNames</a> .
<code>bypass</code>	Whether to set permanent bypass value. Default is TRUE.
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

## Details

This method permanently overrides any default values or mappings defined for the visual properties of the node or nodes specified. To restore defaults and mappings, use [clearNodePropertyBypass](#).

## Value

None

## See Also

[clearNetworkPropertyBypass](#)

## Examples

```
setNetworkPropertyBypass()
```

---

```
setNetworkZoomBypass Set Network Zoom Bypass
```

---

## Description

Set the bypass value for scale factor for the network.

## Usage

```
setNetworkZoomBypass(  
    new.value,  
    bypass = FALSE,  
    network = NULL,  
    base.url = .defaultBaseUrl  
)
```

## Arguments

new.value	Zoom factor
bypass	Whether to set permanent bypass value. Default is FALSE per common use of temporary zoom settings.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

## Details

This method permanently overrides any default values for this visual property. This method ultimately calls the generic function, [setNetworkPropertyBypass](#), which can be used to set any visual property. To restore defaults, use [clearNetworkPropertyBypass](#).

## Value

None

## See Also

[setNetworkPropertyBypass](#), [clearNetworkPropertyBypass](#)

## Examples

```
setNetworkZoomBypass()
```

setNodeBorderColorBypass

*Set Node Border Color Bypass*

---

### Description

Override the border color for particular nodes.

### Usage

```
setNodeBorderColorBypass(  
  node.names,  
  new.colors,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

node.names	List of node names or SUIDs
new.colors	List of hex colors, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method permanently overrides any default values or mappings defined for this visual property of the node or nodes specified. This method ultimately calls the generic function, [setNodePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearNodePropertyBypass](#), see examples.

### Value

None

### See Also

[setNodePropertyBypass](#), [clearNodePropertyBypass](#)

### Examples

```
setNodeBorderColorBypass('Node 1', '#FF55AA')  
setNodeBorderColorBypass(c('Node 1', 'Node 2'), '#FF55AA')  
clearNodePropertyBypass(c('Node 1', 'Node 2'), 'NODE_BORDER_PAINT')
```

---

`setNodeBorderColorDefault`*Set Node Border Color Default*

---

**Description**

Set the default node border color.

**Usage**

```
setNodeBorderColorDefault(  
  new.color,  
  style.name = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

<code>new.color</code>	Color as hex code, e.g., #FD5903
<code>style.name</code>	Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeBorderColorDefault('#FD5903')
```

---

`setNodeBorderColorMapping`*Set Node Border Color Mapping*

---

**Description**

Map table column values to colors to set the node border color.

**Usage**

```
setNodeBorderColorMapping(
  table.column,
  table.column.values = NULL,
  colors = NULL,
  mapping.type = "c",
  default.color = NULL,
  style.name = NULL,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

<code>table.column</code>	Name of Cytoscape table column to map values from
<code>table.column.values</code>	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
<code>colors</code>	List of hex colors to map to <code>table.column.values</code> or a color palette function, e.g., <code>paletteColorBrewerSet3</code> (without quotes). See <code>RColorBrewer::display.brewer.all()</code>
<code>mapping.type</code>	(char) continuous, discrete or passthrough (c,d,p); default is continuous
<code>default.color</code>	Hex color to set as default
<code>style.name</code>	Name of style; default is "default" style
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeBorderColorMapping('score', c(0,5), c('#FFFFFF', '#FF7755'))
setNodeBorderColorMapping('score', colors=paletteColorBrewerRdBu)
setNodeBorderColorMapping('score', colors=paletteColorBrewerSet3, mapping.type='d')
```



---

```
setNodeBorderOpacityBypass  
    Set Node Border Opacity Bypass
```

---

## Description

Override the border opacity for particular nodes.

## Usage

```
setNodeBorderOpacityBypass(  
    node.names,  
    new.values,  
    network = NULL,  
    base.url = .defaultBaseUrl  
)
```

## Arguments

<code>node.names</code>	List of node names or SUIDs
<code>new.values</code>	List of values to set, or single value
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

## Details

This method permanently overrides any default values or mappings defined for this visual property of the node or nodes specified. This method ultimately calls the generic function, [setNodePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearNodePropertyBypass](#).

## Value

None

## See Also

[setNodePropertyBypass](#), [clearNodePropertyBypass](#)

## Examples

```
setNodeBorderOpacityBypass()
```

setNodeBorderOpacityDefault

*Set Node Border Opacity Default*

---

### Description

Set defaults opacity value for all unmapped node borders.

### Usage

```
setNodeBorderOpacityDefault(  
    new.opacity,  
    style.name = NULL,  
    base.url = .defaultBaseUrl  
)
```

### Arguments

<code>new.opacity</code>	Numeric values between 0 and 255; 0 is invisible.
<code>style.name</code>	Name of style; default is "default" style.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Value

None

### Examples

```
setNodeBorderOpacityDefault(50)
```

---

setNodeBorderOpacityMapping

*Set Node Border Opacity Mapping*

---

### Description

Sets opacity for node border only.

**Usage**

```
setNodeBorderOpacityMapping(
  table.column,
  table.column.values = NULL,
  opacities = NULL,
  mapping.type = "c",
  default.opacity = NULL,
  style.name = NULL,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

<code>table.column</code>	Name of Cytoscape table column to map values from
<code>table.column.values</code>	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
<code>opacities</code>	(integer) values between 0 and 255; 0 is invisible. A range of 50 to 255 is used by default for automatic mapping.
<code>mapping.type</code>	(char) continuous, discrete or passthrough (c,d,p); default is continuous
<code>default.opacity</code>	Opacity value to set as default for all unmapped values
<code>style.name</code>	Name of style; default is "default" style
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeBorderOpacityMapping('score')
setNodeBorderOpacityMapping('score', opacities=c(0,100))
setNodeBorderOpacityMapping('score', c(-5,5), c(50,255))
```

---

`setNodeBorderWidthBypass`*Set Node Border Width Bypass*

---

### Description

Override the border width for particular nodes.

### Usage

```
setNodeBorderWidthBypass(  
  node.names,  
  new.sizes,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

<code>node.names</code>	List of node names or SUIDs
<code>new.sizes</code>	List of size values, or single value
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method permanently overrides any default values or mappings defined for this visual property of the node or nodes specified. This method ultimately calls the generic function, [setNodePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearNodePropertyBypass](#), see examples.

### Value

None

### See Also

[setNodePropertyBypass](#), [clearNodePropertyBypass](#)

### Examples

```
setNodeBorderWidthBypass('Node 1', 5)  
setNodeBorderWidthBypass(c('Node 1','Node 2'), 5)  
clearNodePropertyBypass(c('Node 1','Node 2'), 'NODE_BORDER_WIDTH')
```

---

`setNodeBorderWidthDefault`*Set Node Border Width Default*

---

**Description**

Set the default node border width.

**Usage**

```
setNodeBorderWidthDefault(  
    new.width,  
    style.name = NULL,  
    base.url = .defaultBaseUrl  
)
```

**Arguments**

<code>new.width</code>	Numeric value for width
<code>style.name</code>	Name of style; default is "default" style.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeBorderWidthDefault(2)
```

---

`setNodeBorderWidthMapping`*Set Node Border Width Mapping*

---

**Description**

Map table column values to widths to set the node border width.

**Usage**

```

setNodeBorderWidthMapping(
  table.column,
  table.column.values = NULL,
  widths = NULL,
  mapping.type = "c",
  default.width = NULL,
  style.name = NULL,
  network = NULL,
  base.url = .defaultBaseUrl
)

```

**Arguments**

<code>table.column</code>	Name of Cytoscape table column to map values from
<code>table.column.values</code>	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
<code>widths</code>	List of width values to map to <code>table.column.values</code> . A range of 10 to 100 is used by default for automatic mapping.
<code>mapping.type</code>	(char) continuous, discrete or passthrough (c,d,p); default is continuous
<code>default.width</code>	Width value to set as default for all unmapped values
<code>style.name</code>	Name of style; default is "default" style
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```

setNodeBorderWidthMapping('score')
setNodeBorderWidthMapping('score', widths=c(1,10))
setNodeBorderWidthMapping('score', c(0,30), c(1,5))

```

---

setNodeColorBypass      *Set Node Color Bypass*

---

### Description

Set the bypass value for fill color for the specified node or nodes.

### Usage

```
setNodeColorBypass(  
  node.names,  
  new.colors,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

node.names	List of node names or SUIDs
new.colors	List of hex colors, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method permanently overrides any default values or mappings defined for this visual property of the node or nodes specified. This method ultimately calls the generic function, [setNodePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearNodePropertyBypass](#), see examples.

### Value

None

### See Also

[setNodePropertyBypass](#), [clearNodePropertyBypass](#)

### Examples

```
setNodeColorBypass ('node1', '#FF0088')  
setNodeColorBypass (c('node1', 'node2'), c('#88FF88', '#FF0088'))  
clearNodePropertyBypass(c('Node 1', 'Node 2'), 'NODE_FILL_COLOR')
```

---

setNodeColorDefault    *Set Node Color Default*

---

### Description

Set the default node color.

### Usage

```
setNodeColorDefault(new.color, style.name = NULL, base.url = .defaultBaseUrl)
```

### Arguments

new.color	Color as hex code, e.g., #FD5903
style.name	Name of style; default is "default" style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

### Value

None

### Examples

```
setNodeColorDefault('#FD5903')
```

---

setNodeColorMapping    *Set Node Color Mapping*

---

### Description

Map table column values to colors to set the node fill color.

### Usage

```
setNodeColorMapping(  
  table.column,  
  table.column.values = NULL,  
  colors = NULL,  
  mapping.type = "c",  
  default.color = NULL,  
  style.name = NULL,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```



**Arguments**

table.column	Name of Cytoscape table column to map values from
table.column.values	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
colors	List of hex colors to map to table.column.values or a color palette function, e.g., paletteColorBrewerSet3 (without quotes). See RColorBrewer::display.brewer.all()
mapping.type	(char) continuous, discrete or passthrough (c,d,p); default is continuous
default.color	Hex color to set as default
style.name	Name of style; default is "default" style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeColorMapping('score', c(-5,0,5), c('#5577FF', '#FFFFFF', '#FF7755'))
setNodeColorMapping('score', colors=paletteColorBrewerRdBu)
setNodeColorMapping('score', colors=paletteColorBrewerSet3, mapping.type='d')
```

---

```
setNodeComboOpacityMapping
```

*Set Node Combo Opacity Mapping*

---

**Description**

Sets opacity for node fill, border and label all together.

**Usage**

```
setNodeComboOpacityMapping(
  table.column,
  table.column.values = NULL,
  opacities = NULL,
  mapping.type = "c",
  default.opacity = NULL,
  style.name = NULL,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

table.column	Name of Cytoscape table column to map values from
table.column.values	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
opacities	(integer) values between 0 and 255; 0 is invisible. A range of 50 to 255 is used by default for automatic mapping.
mapping.type	(char) continuous, discrete or passthrough (c,d,p); default is continuous
default.opacity	Opacity value to set as default for all unmapped values
style.name	Name of style; default is "default" style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeComboOpacityMapping('score')
setNodeComboOpacityMapping('score', opacities=c(0,100))
setNodeComboOpacityMapping('score', c(-5,5), c(50,255))
```

---

setNodeCustomBarChart *Set Node Custom Bar Chart*

---

**Description**

Makes a bar chart per node using specified node table columns by setting a default custom graphic style.

**Usage**

```
setNodeCustomBarChart(
  columns,
  type = "GROUPED",
  colors = NULL,
  range = NULL,
  orientation = "VERTICAL",
  colAxis = FALSE,
```

```

    rangeAxis = FALSE,
    zeroLine = FALSE,
    axisWidth = 0.25,
    axisColor = "#000000",
    axisFontSize = 1,
    separation = 0,
    slot = 1,
    style.name = NULL,
    base.url = .defaultBaseUrl
  )

```

### Arguments

columns	List of node column names to be displayed, in order.
type	Type of bar chart: GROUPED (default), STACKED, HEAT_STRIPS, or UP_DOWN
colors	(optional) List of colors to be matched with columns or with range, depending on type. Default is a set of colors from an appropriate Brewer palette.
range	(optional) Min and max values of chart. Default is to use min and max from specified data columns.
orientation	(optional) HORIZONTAL or VERTICAL (default).
colAxis	(optional) Show axis with column labels. Default is FALSE.
rangeAxis	(optional) Show axis with range of values. Default is FALSE.
zeroLine	(optional) Show a line at zero. Default is FALSE.
axisWidth	(optional) Width of axis lines, if shown. Default is 0.25.
axisColor	(optional) Color of axis lines, if shown. Default is black.
axisFontSize	(optional) Font size of axis labels, if shown. Default is 1.
separation	(optional) Distance between bars. Default is 0.0.
slot	(optional) Which custom graphics slot to modify. Slots 1-9 are available for independent charts, gradients and images. Default is 1.
style.name	(optional) Name of style; default is "default" style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

### Value

None

### See Also

setNodeCustomPosition, removeNodeCustomGraphics

### Examples

```
setNodeCustomBarChart(c("data1", "data2", "data3"))
```

---

setNodeCustomBoxChart *Set Node Custom Box Chart*

---

### Description

Makes a box chart per node using specified node table columns by setting a default custom graphic style.

### Usage

```
setNodeCustomBoxChart(
    columns,
    colors = NULL,
    range = NULL,
    orientation = "VERTICAL",
    rangeAxis = FALSE,
    zeroLine = FALSE,
    axisWidth = 0.25,
    axisColor = "#000000",
    axisFontSize = 1,
    slot = 1,
    style.name = NULL,
    base.url = .defaultBaseUrl
)
```

### Arguments

columns	List of node column names to be displayed.
colors	(optional) List of colors to be matched with columns or with range, depending on type. Default is a set of colors from an appropriate Brewer palette.
range	(optional) Min and max values of chart. Default is to use min and max from specified data columns.
orientation	(optional) HORIZONTAL or VERTICAL (default).
rangeAxis	(optional) Show axis with range of values. Default is FALSE.
zeroLine	(optional) Show a line at zero. Default is FALSE.
axisWidth	(optional) Width of axis lines, if shown. Default is 0.25.
axisColor	(optional) Color of axis lines, if shown. Default is black.
axisFontSize	(optional) Font size of axis labels, if shown. Default is 1.
slot	(optional) Which custom graphics slot to modify. Slots 1-9 are available for independent charts, gradients and images. Default is 1.
style.name	(optional) Name of style; default is "default" style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**See Also**

setNodeCustomPosition, removeNodeCustomGraphics

**Examples**

```
setNodeCustomBoxChart(c("data1", "data2", "data3"))
```

---

```
setNodeCustomHeatMapChart
```

*Set Node Custom HeatMap Chart*

---

**Description**

Makes a heatmap chart per node using specified node table columns by setting a default custom graphic style.

**Usage**

```
setNodeCustomHeatMapChart(
  columns,
  colors = NULL,
  range = NULL,
  orientation = "HORIZONTAL",
  rangeAxis = FALSE,
  zeroLine = FALSE,
  axisWidth = 0.25,
  axisColor = "#000000",
  axisFontSize = 1,
  slot = 1,
  style.name = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

columns	List of node column names to be displayed.
colors	(optional) List of colors to be matched with columns or with range, depending on type. Default is a set of colors from an appropriate Brewer palette.
range	(optional) Min and max values of chart. Default is to use min and max from specified data columns.
orientation	(optional) VERTICAL or HORIZONTAL (default).

rangeAxis	(optional) Show axis with range of values. Default is FALSE.
zeroLine	(optional) Show a line at zero. Default is FALSE.
axisWidth	(optional) Width of axis lines, if shown. Default is 0.25.
axisColor	(optional) Color of axis lines, if shown. Default is black.
axisFontSize	(optional) Font size of axis labels, if shown. Default is 1.
slot	(optional) Which custom graphics slot to modify. Slots 1-9 are available for independent charts, gradients and images. Default is 1.
style.name	(optional) Name of style; default is "default" style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**See Also**

setNodeCustomPosition, removeNodeCustomGraphics

**Examples**

```
setNodeCustomHeatMapChart(c("data1", "data2", "data3"))
```

---

```
setNodeCustomLinearGradient
```

*Set Node Custom Linear Gradient*

---

**Description**

Makes a gradient fill per node by setting a default custom graphic style.

**Usage**

```
setNodeCustomLinearGradient(
  colors = c("#DDDDDD", "#888888"),
  anchors = c(0, 1),
  angle = 0,
  slot = 1,
  style.name = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

colors	(optional) List of colors to define gradient
anchors	(optional) Position of colors from 0.0 to 1.0.
angle	(optional) Angle of gradient. Default is 0 (left-to-right).
slot	(optional) Which custom graphics slot to modify. Slots 1-9 are available for independent charts, gradients and images. Default is 1.
style.name	(optional) Name of style; default is "default" style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeCustomLinearGradient()
```

---

```
setNodeCustomLineChart
```

*Set Node Custom Line Chart*

---

**Description**

Makes a line chart per node using specified node table columns by setting a default custom graphic style.

**Usage**

```
setNodeCustomLineChart(
  columns,
  colors = NULL,
  range = NULL,
  lineWidth = 1,
  rangeAxis = FALSE,
  zeroLine = FALSE,
  axisWidth = 0.25,
  axisColor = "#000000",
  axisFontSize = 1,
  slot = 1,
  style.name = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

columns	List of node column names to be displayed.
colors	(optional) List of colors to be matched with columns or with range, depending on type. Default is a set of colors from an appropriate Brewer palette.
range	(optional) Min and max values of chart. Default is to use min and max from specified data columns.
lineWidth	(optional) Width of chart line. Default is 1.0.
rangeAxis	(optional) Show axis with range of values. Default is FALSE.
zeroLine	(optional) Show a line at zero. Default is FALSE.
axisWidth	(optional) Width of axis lines, if shown. Default is 0.25.
axisColor	(optional) Color of axis lines, if shown. Default is black.
axisFontSize	(optional) Font size of axis labels, if shown. Default is 1.
slot	(optional) Which custom graphics slot to modify. Slots 1-9 are available for independent charts, gradients and images. Default is 1.
style.name	(optional) Name of style; default is "default" style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**See Also**

setNodeCustomPosition, removeNodeCustomGraphics

**Examples**

```
setNodeCustomLineChart(c("data1", "data2", "data3"))
```

---

setNodeCustomPieChart *Set Node Custom Pie Chart*

---

**Description**

Makes a pie chart per node using specified node table columns by setting a default custom graphic style.



**Usage**

```
setNodeCustomPieChart(  
  columns,  
  colors = NULL,  
  startAngle = 0,  
  slot = 1,  
  style.name = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

columns	List of node column names to be displayed.
colors	(optional) List of colors to be matched with columns or with range, depending on type. Default is a set of colors from an appropriate Brewer palette.
startAngle	(optional) Angle to start filling pie. Default is 0.0.
slot	(optional) Which custom graphics slot to modify. Slots 1-9 are available for independent charts, gradients and images. Default is 1.
style.name	(optional) Name of style; default is "default" style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**See Also**

setNodeCustomPosition, removeNodeCustomGraphics

**Examples**

```
setNodeCustomPieChart(c("data1", "data2", "data3"))
```

---

setNodeCustomPosition *Set Node Custom Position*

---

**Description**

Adjust the position of a custom graphic relative to its node.

**Usage**

```

setNodeCustomPosition(
  nodeAnchor = "C",
  graphicAnchor = "C",
  justification = "c",
  xOffset = 0,
  yOffset = 0,
  slot = 1,
  style.name = NULL,
  base.url = .defaultBaseUrl
)

```

**Arguments**

nodeAnchor	Position on node to place the graphic: NW,N,NE,E,SE,S,SW,W or C for center (default)
graphicAnchor	Position on graphic to place on node: NW,N,NE,E,SE,S,SW,W or C for center (default)
justification	Positioning of content within graphic: l,r,c (default)
xOffset	Additional offset in the x direction
yOffset	Additional offset in the y direction
slot	(optional) Which custom graphics slot to modify. Slots 1-9 are available for independent charts, gradients and images. Default is 1.
style.name	(optional) Name of style; default is "default" style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeCustomPosition()
```

---

setNodeCustomRadialGradient

*Set Node Custom Radial Gradient*

---

**Description**

Makes a gradient fill per node by setting a default custom graphic style.

**Usage**

```
setNodeCustomRadialGradient(
  colors = c("#DDDDDD", "#888888"),
  anchors = c(0, 1),
  xCenter = 0.5,
  yCenter = 0.5,
  slot = 1,
  style.name = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

colors	(optional) List of colors to define gradient
anchors	(optional) Position of colors from 0.0 to 1.0.
xCenter	(optional) X position for center of radial effect from 0.0 to 1.0. Default is 0.5.
yCenter	(optional) Y position for center of radial effect from 0.0 to 1.0. Default is 0.5.
slot	(optional) Which custom graphics slot to modify. Slots 1-9 are available for independent charts, gradients and images. Default is 1.
style.name	(optional) Name of style; default is "default" style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeCustomRadialGradient()
```

---

setNodeCustomRingChart

*Set Node Custom Ring Chart*

---

**Description**

Makes a ring chart per node using specified node table columns by setting a default custom graphic style.

**Usage**

```

setNodeCustomRingChart(
  columns,
  colors = NULL,
  startAngle = 0,
  holeSize = 0.5,
  slot = 1,
  style.name = NULL,
  base.url = .defaultBaseUrl
)

```

**Arguments**

columns	List of node column names to be displayed.
colors	(optional) List of colors to be matched with columns or with range, depending on type. Default is a set of colors from an appropriate Brewer palette.
startAngle	(optional) Angle to start filling ring Default is 0.0.
holeSize	(optional) Size of hole in ring. Ranges 0-1. Default is 0.5.
slot	(optional) Which custom graphics slot to modify. Slots 1-9 are available for independent charts, gradients and images. Default is 1.
style.name	(optional) Name of style; default is "default" style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**See Also**

setNodeCustomPosition, removeNodeCustomGraphics

**Examples**

```
setNodeCustomRingChart(c("data1", "data2", "data3"))
```

---

setNodeFillOpacityBypass

*Set Node Fill Opacity Bypass*

---

**Description**

Override the fill opacity for particular nodes.

## Usage

```
setNodeFillOpacityBypass(  
    node.names,  
    new.values,  
    network = NULL,  
    base.url = .defaultBaseUrl  
)
```

## Arguments

node.names	List of node names or SUIDs
new.values	List of values to set, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

## Details

This method permanently overrides any default values or mappings defined for this visual property of the node or nodes specified. This method ultimately calls the generic function, [setNodePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearNodePropertyBypass](#), see examples.

## Value

None

## See Also

[setNodePropertyBypass](#), [clearNodePropertyBypass](#)

## Examples

```
setNodeFillOpacityBypass('Node 1', 100)  
setNodeFillOpacityBypass(c('Node 1', 'Node 2'), 100)  
clearNodePropertyBypass(c('Node 1', 'Node 2'), 'NODE_TRANSPARENCY')
```

setNodeFillOpacityDefault

*Set Node Fill Opacity Default*

---

### Description

Set default opacity value for all unmapped nodes.

### Usage

```
setNodeFillOpacityDefault(  
  new.opacity,  
  style.name = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

<code>new.opacity</code>	Numeric values between 0 and 255; 0 is invisible.
<code>style.name</code>	Name of style; default is "default" style.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Value

None

### Examples

```
setNodeFillOpacityDefault(50)
```

---

setNodeFillOpacityMapping

*Set Node Fill Opacity Mapping*

---

### Description

Sets opacity for node fill only.

**Usage**

```

setNodeFillOpacityMapping(
  table.column,
  table.column.values = NULL,
  opacities = NULL,
  mapping.type = "c",
  default.opacity = NULL,
  style.name = NULL,
  network = NULL,
  base.url = .defaultBaseUrl
)

```

**Arguments**

<code>table.column</code>	Name of Cytoscape table column to map values from
<code>table.column.values</code>	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
<code>opacities</code>	(integer) values between 0 and 255; 0 is invisible. A range of 50 to 255 is used by default for automatic mapping.
<code>mapping.type</code>	(char) continuous, discrete or passthrough (c,d,p); default is continuous
<code>default.opacity</code>	Opacity value to set as default for all unmapped values
<code>style.name</code>	Name of style; default is "default" style
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```

setNodeFillOpacityMapping('score')
setNodeFillOpacityMapping('score', opacities=c(0,100))
setNodeFillOpacityMapping('score', c(-5,5), c(50,255))

```

---

setNodeFontFaceBypass *Set Node Font Face Bypass*

---

### Description

Override the font face for particular nodes.

### Usage

```
setNodeFontFaceBypass(  
  node.names,  
  new.fonts,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

node.names	List of node names or SUIDs
new.fonts	List of font faces, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method permanently overrides any default values or mappings defined for this visual property of the node or nodes specified. This method ultimately calls the generic function, [setNodePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearNodePropertyBypass](#).

### Value

None

### See Also

[setNodePropertyBypass](#), [clearNodePropertyBypass](#)

### Examples

```
setNodeFontFaceBypass()
```



---

`setNodeFontFaceDefault`*Set Node Font Face Default*

---

**Description**

Set the default node font.

**Usage**

```
setNodeFontFaceDefault(new.font, style.name = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>new.font</code>	String specification of font face, style and size, e.g., "SansSerif,plain,12" or "Dialog,plain,10"
<code>style.name</code>	Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeFontFaceDefault("Dialog,plain,10")
```

---

`setNodeFontFaceMapping`*Set Node Font Face Mapping*

---

**Description**

Sets font face for node labels.

**Usage**

```

setNodeFontFaceMapping(
  table.column,
  table.column.values,
  fonts,
  mapping.type = "d",
  default.font = NULL,
  style.name = NULL,
  network = NULL,
  base.url = .defaultBaseUrl
)

```

**Arguments**

<code>table.column</code>	Name of Cytoscape table column to map values from
<code>table.column.values</code>	List of values from Cytoscape table to be used in mapping
<code>fonts</code>	List of string specifications of font face, style and size, e.g., <code>c("SansSerif,plain,12", "Dialog,plain,10")</code>
<code>mapping.type</code>	(char) discrete or passthrough (d,p); default is discrete
<code>default.font</code>	String specification of font face, style and size, e.g., <code>"SansSerif,plain,12"</code> or <code>"Dialog,plain,10"</code>
<code>style.name</code>	Name of style; default is "default" style
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```

setNodeFontFaceMapping("myfonts", c("normal", "small"),
  c("SansSerif,plain,12", "Dialog,plain,10"))

```

---

setNodeFontSizeBypass *Set Node Font Size Bypass*

---

### Description

Override the font size for particular nodes.

### Usage

```
setNodeFontSizeBypass(  
  node.names,  
  new.sizes,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

node.names	List of node names or SUIDs
new.sizes	List of size values, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method permanently overrides any default values or mappings defined for this visual property of the node or nodes specified. This method ultimately calls the generic function, [setNodePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearNodePropertyBypass](#), see examples.

### Value

None

### See Also

[setNodePropertyBypass](#), [clearNodePropertyBypass](#)

### Examples

```
setNodeFontSizeBypass('Node 1', 5)  
setNodeFontSizeBypass(c('Node 1', 'Node 2'), 5)  
clearNodePropertyBypass(c('Node 1', 'Node 2'), 'NODE_LABEL_FONT_SIZE')
```

setNodeFontSizeDefault

*Set Node Font Size Default*

---

### **Description**

Set the default node font size.

### **Usage**

```
setNodeFontSizeDefault(new.size, style.name = NULL, base.url = .defaultBaseUrl)
```

### **Arguments**

<code>new.size</code>	Numeric value for size
<code>style.name</code>	Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### **Value**

None

### **Examples**

```
setNodeFontSizeDefault(12)
```

---

setNodeFontSizeMapping

*Set Node Font Size Mapping*

---

### **Description**

Map table column values to sizes to set the node size.

**Usage**

```
setNodeFontSizeMapping(  
  table.column,  
  table.column.values = NULL,  
  sizes = NULL,  
  mapping.type = "c",  
  default.size = NULL,  
  style.name = NULL,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

table.column	Name of Cytoscape table column to map values from
table.column.values	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
sizes	List of sizes to map to table.column.values. A range of 10 to 100 is used by default for automatic mapping.
mapping.type	(char) continuous, discrete or passthrough (c,d,p); default is continuous
default.size	Size value to set as default
style.name	Name of style; default is "default" style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeFontSizeMapping('score')  
setNodeFontSizeMapping('score', sizes=c(6,24))  
setNodeFontSizeMapping('score', c(0,30), c(35,55))
```

---

setNodeHeightBypass    *Set Node Height Bypass*

---

### Description

Override the height for particular nodes.

### Usage

```
setNodeHeightBypass(  
  node.names,  
  new.heights,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

node.names	List of node names or SUIDs
new.heights	List of height values, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method permanently overrides any default values or mappings defined for this visual property of the node or nodes specified. This method ultimately calls the generic function, [setNodePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearNodePropertyBypass](#), see examples.

### Value

None

### See Also

[setNodePropertyBypass](#), [clearNodePropertyBypass](#)

### Examples

```
setNodeHeightBypass('Node 1', 35)  
setNodeHeightBypass(c('Node 1', 'Node 2'), 35)  
clearNodePropertyBypass(c('Node 1', 'Node 2'), 'NODE_HEIGHT')
```

---

setNodeHeightDefault *Set Node Height Default*

---

### Description

Set the default node height.

### Usage

```
setNodeHeightDefault(new.height, style.name = NULL, base.url = .defaultBaseUrl)
```

### Arguments

new.height	Numeric value for height.
style.name	Name of style; default is "default" style.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

### Value

None

### Examples

```
setNodeHeightDefault(35)
```

---

setNodeHeightMapping *Set Node Height Mapping*

---

### Description

Map table column values to the node heights.

### Usage

```
setNodeHeightMapping(  
  table.column,  
  table.column.values = NULL,  
  heights = NULL,  
  mapping.type = "c",  
  default.height = NULL,  
  style.name = NULL,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

table.column	Name of Cytoscape table column to map values from
table.column.values	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
heights	List of height values to map to table.column.values. A range of 10 to 100 is used by default for automatic mapping.
mapping.type	(char) continuous, discrete or passthrough (c,d,p); default is continuous
default.height	Size value to set as default
style.name	Name of style; default is "default" style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Using this function will unlock node width and height to use separate values.

**Value**

None

**Examples**

```
setNodeHeightMapping('score')
setNodeHeightMapping('score', heights=c(30,80))
setNodeHeightMapping('score', c(0,30), c(35,55))
```

---

setNodeLabelBypass      *Set Node Label Bypass*

---

**Description**

Override the label for particular nodes.

**Usage**

```
setNodeLabelBypass(
  node.names,
  new.labels,
  network = NULL,
  base.url = .defaultBaseUrl
)
```



### Arguments

node.names	List of node names or SUIDs
new.labels	List of labels, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method permanently overrides any default values or mappings defined for this visual property of the node or nodes specified. This method ultimately calls the generic function, [setNodePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearNodePropertyBypass](#), see examples.

### Value

None

### See Also

[setNodePropertyBypass](#), [clearNodePropertyBypass](#)

### Examples

```
setNodeLabelBypass('Node 1', 'Custom Label')
setNodeLabelBypass(c('Node 1', 'Node 2'), 'Custom Label')
clearNodePropertyBypass(c('Node 1', 'Node 2'), 'NODE_LABEL')
```

---

```
setNodeLabelColorBypass
```

*Set Node Label Color Bypass*

---

### Description

Override the label color for particular nodes.

### Usage

```
setNodeLabelColorBypass(  
  node.names,  
  new.colors,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

node.names	List of node names or SUIDs
new.colors	List of hex colors, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method permanently overrides any default values or mappings defined for this visual property of the node or nodes specified. This method ultimately calls the generic function, [setNodePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearNodePropertyBypass](#), see examples.

### Value

None

### See Also

[setNodePropertyBypass](#), [clearNodePropertyBypass](#)

### Examples

```
setNodeLabelColorBypass('Node 1', '#FF55AA')
setNodeLabelColorBypass(c('Node 1', 'Node 2'), '#FF55AA')
clearNodePropertyBypass(c('Node 1', 'Node 2'), 'NODE_LABEL_COLOR')
```

---

setNodeLabelColorDefault

*Set Node Label Color Default*

---

### Description

Set the default node label color.

### Usage

```
setNodeLabelColorDefault(  
  new.color,  
  style.name = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

<code>new.color</code>	Color as hex code, e.g., #FD5903
<code>style.name</code>	Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeLabelColorDefault('#FD5903')
```

---

```
setNodeLabelColorMapping
```

*Set Node Label Color Mapping*

---

**Description**

Map table column values to colors to set the node border color.

**Usage**

```
setNodeLabelColorMapping(  
  table.column,  
  table.column.values = NULL,  
  colors = NULL,  
  mapping.type = "c",  
  default.color = NULL,  
  style.name = NULL,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

<code>table.column</code>	Name of Cytoscape table column to map values from
<code>table.column.values</code>	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
<code>colors</code>	List of hex colors to map to <code>table.column.values</code> or a color palette function, e.g., <code>paletteColorBrewerSet3</code> (without quotes). See <code>RColorBrewer::display.brewer.all()</code>
<code>mapping.type</code>	(char) continuous, discrete or passthrough (c,d,p); default is continuous

default.color	Hex color to set as default
style.name	Name of style; default is "default" style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeLabelColorMapping('score', c(0,5), c('#FFFFFF', '#FF7755'))
setNodeLabelColorMapping('score', colors=paletteColorBrewerRdBu)
setNodeLabelColorMapping('score', colors=paletteColorBrewerSet3, mapping.type='d')
```

---

setNodeLabelDefault    *Set Node Label Default*

---

**Description**

Set the default node label.

**Usage**

```
setNodeLabelDefault(new.label, style.name = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

new.label	String label for unmapped nodes.
style.name	Name of style; default is "default" style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeLabelDefault('unknown')
```

---

setNodeLabelMapping    *Set Node Label Mapping*

---

### Description

Pass the values from a table column to display as node labels.

### Usage

```
setNodeLabelMapping(  
  table.column,  
  style.name = NULL,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

table.column	Name of Cytoscape table column to map values from
style.name	Name of style; default is "default" style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

### Value

None

### Examples

```
setNodeLabelMapping('name')
```

---

setNodeLabelOpacityBypass  
*Set Node Label Opacity Bypass*

---

### Description

Override the label opacity for particular nodes.

## Usage

```
setNodeLabelOpacityBypass(  
  node.names,  
  new.values,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

## Arguments

node.names	List of node names or SUIDs
new.values	List of values to set, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

## Details

This method permanently overrides any default values or mappings defined for this visual property of the node or nodes specified. This method ultimately calls the generic function, [setNodePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearNodePropertyBypass](#).

## Value

None

## See Also

[setNodePropertyBypass](#), [clearNodePropertyBypass](#)

## Examples

```
setNodeLabelOpacityBypass()
```

---

setNodeLabelOpacityDefault

*Set Node Label Opacity Default*

---

## Description

Set default opacity value for all unmapped node labels.

**Usage**

```
setNodeLabelOpacityDefault(  
    new.opacity,  
    style.name = NULL,  
    base.url = .defaultBaseUrl  
)
```

**Arguments**

<code>new.opacity</code>	Numeric values between 0 and 255; 0 is invisible.
<code>style.name</code>	Name of style; default is "default" style.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeLabelOpacityDefault(50)
```

---

setNodeLabelOpacityMapping  
*Set Node Label Opacity Mapping*

---

**Description**

Sets opacity for node label only.

**Usage**

```
setNodeLabelOpacityMapping(  
    table.column,  
    table.column.values = NULL,  
    opacities = NULL,  
    mapping.type = "c",  
    default.opacity = NULL,  
    style.name = NULL,  
    network = NULL,  
    base.url = .defaultBaseUrl  
)
```

**Arguments**

table.column	Name of Cytoscape table column to map values from
table.column.values	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
opacities	(integer) values between 0 and 255; 0 is invisible. A range of 50 to 255 is used by default for automatic mapping.
mapping.type	(char) continuous, discrete or passthrough (c,d,p); default is continuous
default.opacity	Opacity value to set as default for all unmapped values
style.name	Name of style; default is "default" style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeLabelOpacityMapping('score')
setNodeLabelOpacityMapping('score', opacities=c(0,100))
setNodeLabelOpacityMapping('score', c(-5,5), c(50,255))
```

---

setNodeOpacityBypass    *Set Node Opacity Bypass*

---

**Description**

Set the bypass value for node fill, label and border opacity for the specified node or nodes.

**Usage**

```
setNodeOpacityBypass(
  node.names,
  new.values,
  network = NULL,
  base.url = .defaultBaseUrl
)
```



### Arguments

node.names	List of node names or SUIDs
new.values	List of values to set, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method permanently overrides any default values or mappings defined for this visual property of the node or nodes specified. This method ultimately calls the generic function, [setNodePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearNodeOpacityBypass](#), see examples.

### Value

None

### See Also

[setNodePropertyBypass](#), [clearNodeOpacityBypass](#)

### Examples

```
setNodeOpacityBypass('Node 1', 100)
setNodeOpacityBypass(c('Node 1', 'Node 2'), 100)
clearNodeOpacityBypass(c('Node 1', 'Node 2'))
```

---

setNodePropertyBypass *Set Node Property Bypass*

---

### Description

Set bypass values for any node property of the specified nodes, overriding default values and mappings defined by any visual style.

### Usage

```
setNodePropertyBypass(
  node.names,
  new.values,
  visual.property,
  bypass = TRUE,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

### Arguments

node.names	List of node names or SUIDs
new.values	List of values to set, or single value
visual.property	Name of a visual property. See <a href="#">getVisualPropertyNames</a> .
bypass	Whether to set permanent bypass value. Default is TRUE.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method permanently overrides any default values or mappings defined for the visual properties of the node or nodes specified. To restore defaults and mappings, use [clearNodePropertyBypass](#).

### Value

None

### See Also

[clearNodePropertyBypass](#)

### Examples

```
setNodePropertyBypass()
```

---

setNodeSelectionColorDefault  
*Set Node Selection Color Default*

---

### Description

Set the default selection node color.

### Usage

```
setNodeSelectionColorDefault(  
  new.color,  
  style.name = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

<code>new.color</code>	Color as hex code, e.g., #FD5903
<code>style.name</code>	(optional) Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeSelectionColorDefault('#FD5903')
```

---

```
setNodeShapeBypass      Set Node Shape Bypass
```

---

**Description**

Override the shape for particular nodes.

**Usage**

```
setNodeShapeBypass(
  node.names,
  new.shapes,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

<code>node.names</code>	List of node names or SUIDs
<code>new.shapes</code>	List of shapes, or single value. See <a href="#">getNodeShapes</a> .
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Details**

This method permanently overrides any default values or mappings defined for this visual property of the node or nodes specified. This method ultimately calls the generic function, [setNodePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearNodePropertyBypass](#), see examples.

**Value**

None

**See Also**[setNodePropertyBypass](#), [clearNodePropertyBypass](#)**Examples**

```
setNodeShapeBypass('Node 1', 'ROUND_RECTANGLE')
setNodeShapeBypass(c('Node 1', 'Node 2'), 'ROUND_RECTANGLE')
clearNodePropertyBypass(c('Node 1', 'Node 2'), 'NODE_SHAPE')
```

---

setNodeShapeDefault    *Set Node Shape Default*

---

**Description**

Set the default node shape.

**Usage**

```
setNodeShapeDefault(new.shape, style.name = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

<code>new.shape</code>	Name of shape, e.g., ELLIPSE, RECTANGLE, etc (see <a href="#">getNodeShapes</a> )
<code>style.name</code>	Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeShapeDefault('ELLIPSE')
```

---

setNodeShapeMapping    *Set Node Shape Mapping*

---

### Description

Map table column values to shapes to set the node shape.

### Usage

```
setNodeShapeMapping(  
  table.column,  
  table.column.values = NULL,  
  shapes = NULL,  
  default.shape = NULL,  
  style.name = NULL,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

table.column	Name of Cytoscape table column to map values from
table.column.values	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
shapes	List of shapes to map to table.column.values. Leave NULL to perform an automatic mapping to available shapes. See <a href="#">getNodeShapes</a>
default.shape	Shape to set as default. See <a href="#">getNodeShapes</a>
style.name	Name of style; default is "default" style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

### Value

None

### Examples

```
setNodeShapeMapping('type')  
setNodeShapeMapping('type',c('protein','dna'),c('ELLIPSE','RECTANGLE'))
```

---

setNodeSizeBypass      *Set Node Size Bypass*

---

### Description

Sets the bypass value of node size for one or more nodes. Only applicable if node dimensions are locked. See `lockNodeDimensions`.

### Usage

```
setNodeSizeBypass(  
  node.names,  
  new.sizes,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

### Arguments

<code>node.names</code>	List of node names or SUIDs
<code>new.sizes</code>	List of size values, or single value
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method permanently overrides any default values or mappings defined for this visual property of the node or nodes specified. This method ultimately calls the generic function, [setNodePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearNodePropertyBypass](#), see examples.

### Value

None

### See Also

[setNodePropertyBypass](#), [clearNodePropertyBypass](#)

### Examples

```
setNodeSizeBypass('Node 1', 35)  
setNodeSizeBypass(c('Node 1', 'Node 2'), 35)  
clearNodePropertyBypass(c('Node 1', 'Node 2'), 'NODE_SIZE')
```

---

setNodeSizeDefault      *Set Node Size Default*

---

### Description

Set the default node size.

### Usage

```
setNodeSizeDefault(new.size, style.name = NULL, base.url = .defaultBaseUrl)
```

### Arguments

new.size	Numeric value for size
style.name	Name of style; default is "default" style.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

### Value

None

### Examples

```
setNodeSizeDefault(35)
```

---

setNodeSizeMapping      *Set Node Size Mapping*

---

### Description

Map table column values to node font sizes.

### Usage

```
setNodeSizeMapping(  
  table.column,  
  table.column.values = NULL,  
  sizes = NULL,  
  mapping.type = "c",  
  default.size = NULL,  
  style.name = NULL,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

table.column	Name of Cytoscape table column to map values from
table.column.values	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
sizes	List of sizes to map to table.column.values. A range of 10 to 100 is used by default for automatic mapping.
mapping.type	(char) continuous, discrete or passthrough (c,d,p); default is continuous
default.size	Size value to set as default
style.name	Name of style; default is "default" style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Using this function will lock node width and height to use a singular "size" value.

**Value**

None

**Examples**

```
setNodeSizeMapping('score')
setNodeSizeMapping('score', sizes=c(30,80))
setNodeSizeMapping('score', c(0,30), c(35,55))
```

---

setNodeTooltipBypass *Set Node Tooltip Bypass*

---

**Description**

Sets a bypass tooltip for one or more nodes

**Usage**

```
setNodeTooltipBypass(
  node.names,
  new.tooltip,
  network = NULL,
  base.url = .defaultBaseUrl
)
```



**Arguments**

node.names	List of node names or SUIDs
new.tooltip	List of tooltips, or a single tooltip
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Details**

This method permanently overrides any default values or mappings defined for this visual property of the node or nodes specified. This method ultimately calls the generic function, [setNodePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearNodePropertyBypass](#), see examples.

**Value**

None

**See Also**

[setNodePropertyBypass](#), [clearNodePropertyBypass](#)

**Examples**

```
setNodeTooltipBypass('Node 1', 'This is an important node.')
```

---

setNodeTooltipDefault *Set Node Tooltip Default*

---

**Description**

Set the default node tooltip

**Usage**

```
setNodeTooltipDefault(  
  new.tooltip,  
  style.name = NULL,  
  base.url = .defaultBaseUrl  
)
```

**Arguments**

<code>new.tooltip</code>	String tooltip for unmapped nodes.
<code>style.name</code>	Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeTooltipDefault('unknown')
```

---

```
setNodeTooltipMapping Set Node Tooltip Mapping
```

---

**Description**

Pass the values from a table column to display as node tooltips.

**Usage**

```
setNodeTooltipMapping(
  table.column,
  style.name = NULL,
  network = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

<code>table.column</code>	Name of Cytoscape table column to map values from
<code>style.name</code>	Name of style; default is "default" style
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

## Examples

```
setNodeTooltipMapping('description')
```

---

setNodeWidthBypass	<i>Set Node Width Bypass</i>
--------------------	------------------------------

---

## Description

Override the width for particular nodes.

## Usage

```
setNodeWidthBypass(  
  node.names,  
  new.widths,  
  network = NULL,  
  base.url = .defaultBaseUrl  
)
```

## Arguments

node.names	List of node names or SUIDs
new.widths	List of width values, or single value
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

## Details

This method permanently overrides any default values or mappings defined for this visual property of the node or nodes specified. This method ultimately calls the generic function, [setNodePropertyBypass](#), which can be used to set any visual property. To restore defaults and mappings, use [clearNodePropertyBypass](#), see examples.

## Value

None

## See Also

[setNodePropertyBypass](#), [clearNodePropertyBypass](#)

**Examples**

```
setNodeWidthBypass('Node 1', 35)
setNodeWidthBypass(c('Node 1', 'Node 2'), 35)
clearNodePropertyBypass(c('Node 1', 'Node 2'), 'NODE_WIDTH')
```

---

setNodeWidthDefault    *Set Node Width Default*

---

**Description**

Set the default node width.

**Usage**

```
setNodeWidthDefault(new.width, style.name = NULL, base.url = .defaultBaseUrl)
```

**Arguments**

new.width	Numeric value for width.
style.name	Name of style; default is "default" style.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setNodeWidthDefault(35)
```

---

setNodeWidthMapping    *Set Node Width Mapping*

---

**Description**

Map table column values to the node widths.

**Usage**

```

setNodeWidthMapping(
  table.column,
  table.column.values = NULL,
  widths = NULL,
  mapping.type = "c",
  default.width = NULL,
  style.name = NULL,
  network = NULL,
  base.url = .defaultBaseUrl
)

```

**Arguments**

table.column	Name of Cytoscape table column to map values from
table.column.values	List of values from Cytoscape table to be used in mapping. Leave NULL to perform an automatic mapping to all column values.
widths	List of width values to map to table.column.values. A range of 10 to 100 is used by default for automatic mapping.
mapping.type	(char) continuous, discrete or passthrough (c,d,p); default is continuous
default.width	Size value to set as default
style.name	Name of style; default is "default" style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <a href="http://localhost:1234">http://localhost:1234</a> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None #' @details Using this function will unlock node width and height to use separate values.

**Examples**

```

setNodeWidthMapping('score')
setNodeWidthMapping('score', widths=c(30,80))
setNodeWidthMapping('score', c(0,30), c(35,55))

```

---

setNotebookIsRunning    *setNotebookIsRunning*

---

**Description**

setNotebookIsRunning

**Usage**

setNotebookIsRunning(newState = NULL)

**Arguments**

newState            new state of running remote

**Value**

oldState

**Examples**

setNotebookIsRunning()

---

setSandboxReinitialize  
                          *setSandboxReinitialize*

---

**Description**

Set and return flag indicating that next command should reinitialize the sandbox according to the default\_sandbox.

**Usage**

setSandboxReinitialize(doReinitialize = TRUE)

**Arguments**

doReinitialize    default is TRUE

**Value**

sandbox reinitialize

**Examples**

setCurrentSandbox()

---

`setStyleDependencies` *Set Style Dependencies*

---

**Description**

Sets the values of dependencies in a style, overriding any prior settings.

**Usage**

```
setStyleDependencies(  
    style.name = NULL,  
    dependencies,  
    base.url = .defaultBaseUrl  
)
```

**Arguments**

<code>style.name</code>	Name of style; default is "default" style
<code>dependencies</code>	A list of style dependencies, see Available Dependencies below. Note: each dependency is set by a boolean, TRUE or FALSE (T or F)
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

server response

**Available Dependencies**

`arrowColorMatchesEdge` `nodeCustomGraphicsSizeSync` `nodeSizeLocked`

**Examples**

```
setStyleDependencies("myStyle", list(nodeSizeLocked=TRUE))
```

---

`setVisualPropertyDefault`*Set Visual Property Default*

---

**Description**

Set the default value for a visual property.

**Usage**

```
setVisualPropertyDefault(  
    style.string,  
    style.name = NULL,  
    base.url = .defaultBaseUrl  
)
```

**Arguments**

<code>style.string</code>	A named list including "visualProperty" and "value"
<code>style.name</code>	Name of style; default is "default" style
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setVisualPropertyDefault(list(visualProperty = "NODE_SIZE", value = 35))
```

---

`setVisualStyle`*Set Visual Style*

---

**Description**

Apply a visual style to a network.

**Usage**

```
setVisualStyle(style.name, network = NULL, base.url = .defaultBaseUrl)
```



**Arguments**

style.name	Name of a visual style
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
setVisualStyle()
```

---

```
spoofResponse-class  spoofResponse
```

---

**Description**

Call CyREST as a remote service via Jupyter-bridge

---

```
syncNodeCustomGraphicsSize
    Sync Node Custom Graphics Size
```

---

**Description**

Set a boolean value to have the size of custom graphics match that of the node.

**Usage**

```
syncNodeCustomGraphicsSize(
  new.state,
  style.name = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

new.state	(Boolean) Whether to sync node custom graphics size
style.name	Name of style; default is "default" style
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
syncNodeCustomGraphicsSize(TRUE)
```

---

toggleGraphicsDetails *Toggle Graphics Details*

---

**Description**

Regardless of the current zoom level and network size, show (or hide) graphics details, e.g., node labels.

**Usage**

```
toggleGraphicsDetails(base.url = .defaultBaseUrl)
```

**Arguments**

`base.url` (optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is `http://localhost:1234` and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Displaying graphics details on a very large network will affect pan and zoom performance, depending on your available RAM. See [cytoscapeMemoryStatus](#).

**Value**

None

**Examples**

```
showGraphicsDetails(TRUE)
```

---

`unhideAll`*Unhide All*

---

### Description

Unhide all previously hidden nodes and edges, by clearing the Visible property bypass value.

### Usage

```
unhideAll(network = NULL, base.url = .defaultBaseUrl)
```

### Arguments

<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method sets node and edge visibility bypass to true, overriding any defaults or mappings. Pending CyREST updates, this method will ultimately call the generic function, [clearEdgePropertyBypass](#), which can be used to clear any visual property.

### Value

None

### See Also

[clearEdgePropertyBypass](#), [unhideNodes](#) [unhideEdges](#)

### Examples

```
unhideAll()
```

---

`unhideEdges`*Unhide Edges*

---

### Description

Unhide specified edges that were previously hidden, by clearing the Visible property bypass value.

### Usage

```
unhideEdges(edge.names, network = NULL, base.url = .defaultBaseUrl)
```

### Arguments

<code>edge.names</code>	List of edge names or SUIDs
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method ultimately calls the generic function, [clearEdgePropertyBypass](#), which can be used to clear any visual property.

### Value

None

### See Also

[clearEdgePropertyBypass](#), [unhideAll](#)

### Examples

```
unhideEdges()
```

---

`unhideNodes`*Unhide Nodes*

---

### Description

Unhide specified nodes that were previously hidden, by clearing the Node Visible property bypass value.

### Usage

```
unhideNodes(node.names, network = NULL, base.url = .defaultBaseUrl)
```

### Arguments

<code>node.names</code>	List of node names or SUIDs
<code>network</code>	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

### Details

This method ultimately calls the generic function, [clearNodePropertyBypass](#), which can be used to clear any visual property.

### Value

None

### See Also

[clearNodePropertyBypass](#), [unhideAll](#)

### Examples

```
unhideNodes()
```

---

 uninstallApp

*Uninstall App*


---

**Description**

Uninstall an app from Cytoscape.

**Usage**

```
uninstallApp(app, base.url = .defaultBaseUrl)
```

**Arguments**

app	Name of app
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

```
uninstallApp()
```

---

updateApp

*Update App*


---

**Description**

Update a Cytoscape app to the latest available version.

**Usage**

```
updateApp(app, base.url = .defaultBaseUrl)
```

**Arguments**

app	Name of app
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Value**

None

**Examples**

updateApp()

---

 updateNetworkInNDEx     *Update Network In NDEx*


---

**Description**

Update an existing network in NDEx, given a previously associated Cytoscape network, e.g., previously exported to NDEx or imported from NDEx.

**Usage**

```
updateNetworkInNDEx(
  username,
  password,
  isPublic,
  network = NULL,
  metadata = NULL,
  base.url = .defaultBaseUrl
)
```

**Arguments**

username	NDEx account username
password	NDEx account password
isPublic	(Boolean) Whether to make the network publicly accessible at NDEx.
network	(optional) Name or SUID of the network. Default is the "current" network active in Cytoscape.
metadata	(optional) A list of structured information describing the network
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

NDEx identifier (externalId) for the updated submission

**Examples**

```
updateNetworkInNDEx("user", "pass", TRUE)
```

---

updateStyleDefaults     *Updates the default values of visual properties in a style*

---

**Description**

Updates visual property defaults, overriding any prior settings. See mapVisualProperty for the list of visual properties.

**Usage**

```
updateStyleDefaults(style.name, defaults, base.url = .defaultBaseUrl)
```

**Arguments**

style.name	(char) name for style
defaults	(list) a list of visual property default settings
base.url	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is http://localhost:1234 and the latest version of the CyREST API supported by this version of RCy3.

**Value**

server response

**See Also**

mapVisualProperty

**Examples**

```
updateStyleDefaults('myStyle',list('node fill color'='#0000FF','node size'=50))
```

---

updateStyleMapping     *Updates a visual property mapping in a style*

---

**Description**

Updates the visual property mapping, overriding any prior mapping. Creates a visual property mapping if it doesn't already exist in the style.

**Usage**

```
updateStyleMapping(style.name, mapping, base.url = .defaultBaseUrl)
```



**Arguments**

<code>style.name</code>	(char) name for style
<code>mapping</code>	a single visual property mapping, see <code>mapVisualProperty</code>
<code>base.url</code>	(optional) Ignore unless you need to specify a custom domain, port or version to connect to the CyREST API. Default is <code>http://localhost:1234</code> and the latest version of the CyREST API supported by this version of RCy3.

**Details**

Requires visual property mappings to be previously created, see `mapVisualProperty`.

**Value**

server response

**See Also**

`mapVisualProperty`

**Examples**

```
updateStyleMapping('myStyle',mapVisualProperty('node label','name','p'))
```

# Index

- [.getDefaultSandbox, 10](#)
- [.getRequester, 10](#)
  
- [addAnnotationText, 11](#)
- [addCyEdges, 12](#)
- [addCyNodes, 13](#)
- [AddToGroup, 14](#)
- [analyzeNetwork, 15](#)
- [applyFilter, 15](#)
  
- [bundleEdges, 17](#)
  
- [checkNotebookIsRunning, 17](#)
- [checkRunningRemote, 18](#)
- [clearEdgeBends, 18](#)
- [clearEdgePropertyBypass, 19, 197, 198, 200, 203, 206, 207, 212, 214, 217, 220, 223, 225, 229, 232, 236, 239, 307, 308](#)
- [clearNetworkCenterBypass, 20](#)
- [clearNetworkPropertyBypass, 20, 243–245](#)
- [clearNetworkZoomBypass, 21](#)
- [clearNodeOpacityBypass, 22, 289](#)
- [clearNodePropertyBypass, 23, 244, 246, 249, 252, 255, 269, 272, 275, 278, 281, 282, 286, 290–292, 294, 297, 299, 309](#)
- [clearSelection, 24](#)
- [cloneNetwork, 24](#)
- [closeSession, 25](#)
- [collapseGroup, 26](#)
- [commandEcho, 27](#)
- [commandOpenDialog, 27](#)
- [commandPause, 28](#)
- [commandQuit, 29](#)
- [commandRunFile, 29](#)
- [commandsAPI, 30](#)
- [commandsGET, 30](#)
- [commandsHelp, 31](#)
- [commandSleep, 32](#)
  
- [commandsPOST, 32](#)
- [commandsRun, 33](#)
- [copyVisualStyle, 34](#)
- [createColumnFilter, 34](#)
- [createCompositeFilter, 36](#)
- [createDegreeFilter, 37](#)
- [createGraphFromNetwork, 38](#)
- [createGroup, 39](#)
- [createGroupByColumn, 40](#)
- [createIgraphFromNetwork, 41](#)
- [createNetworkFromDataFrames, 42](#)
- [createNetworkFromGraph, 43](#)
- [createNetworkFromIgraph, 44](#)
- [createSubnetwork, 45](#)
- [createVisualStyle, 46](#)
- [cybrowserClose, 47](#)
- [cybrowserDialog, 48, 49, 51](#)
- [cybrowserHide, 49, 49, 51](#)
- [cybrowserList, 50](#)
- [cybrowserSend, 50](#)
- [cybrowserShow, 49, 51](#)
- [cybrowserVersion, 52](#)
- [cyrestAPI, 52](#)
- [cyrestDELETE, 53](#)
- [cyrestGET, 54](#)
- [cyrestPOST, 54](#)
- [cyrestPUT, 55](#)
- [cytoscapeApiVersions, 56](#)
- [cytoscapeFreeMemory, 57](#)
- [cytoscapeMemoryStatus, 57, 306](#)
- [cytoscapeNumberOfCores, 58](#)
- [cytoscapePing, 59](#)
- [cytoscapeVersionInfo, 59](#)
  
- [deleteAllNetworks, 60](#)
- [deleteAnnotation, 61](#)
- [deleteDuplicateEdges, 61](#)
- [deleteGroup, 62](#)
- [deleteNetwork, 63](#)
- [deleteSelectedEdges, 64](#)

- deleteSelectedNodes, 64
- deleteSelfLoops, 65
- deleteStyleMapping, 66
- deleteTableColumn, 66
- deleteVisualStyle, 67
- diffusionAdvanced, 68
- diffusionBasic, 69
- disableApp, 69
- dockPanel, 70
- doInitializeSandbox, 71
- doRequestRemote, 71
- doSetSandbox, 72
  
- enableApp, 73
- expandGroup, 73
- exportFilters, 74
- exportImage, 75
- exportNetwork, 76
- exportNetworkToNDEx, 77
- exportVisualStyles, 78
  
- findRemoteCytoscape, 79
- fitContent, 76, 79
- floatPanel, 80
  
- getAbsSandboxPath, 81
- getAllEdges, 81
- getAllNodes, 82
- getAllStyleMappings, 83
- getAnnotationList, 83
- getAppInformation, 84
- getAppStatus, 85
- getAppUpdates, 85
- getArrowShapes, 86, 228–231, 235–237
- getAvailableApps, 86
- getBackgroundColorDefault, 87
- getBrowserClientChannel, 88
- getBrowserClientJs, 88
- getCollectionList, 89
- getCollectionName, 89
- getCollectionNetworks, 90
- getCollectionSuid, 91
- getCurrentSandbox, 91
- getCurrentSandboxName, 92
- getCurrentSandboxPath, 92
- getCurrentStyle, 93
- getDefaultSandbox, 93
- getDefaultSandboxPath, 94
- getDisabledApps, 94
  
- getEdgeColor, 95
- getEdgeCount, 95
- getEdgeInfo, 96
- getEdgeLineStyle, 97
- getEdgeLineWidth, 97
- getEdgeProperty, 98
- getEdgeSelectionColorDefault, 99
- getEdgeTargetArrowShape, 100
- getFilterList, 100
- getFirstNeighbors, 101
- getGroupInfo, 102
- getInstalledApps, 103
- getJupyterBridgeURL, 103
- getLayoutNameMapping, 104
- getLayoutNames, 104, 105–107, 144, 241
- getLayoutPropertyNames, 105, 106, 107, 241
- getLayoutPropertyType, 106
- getLayoutPropertyValue, 107
- getLineStyle, 108, 215, 216
- getNetworkCenter, 108
- getNetworkCount, 109
- getNetworkList, 110
- getNetworkName, 110
- getNetworkNDExId, 111
- getNetworkProperty, 112
- getNetworkSuid, 112
- getNetworkViews, 113
- getNetworkViewSuid, 114
- getNetworkZoom, 114
- getNodeColor, 115
- getNodeCount, 116
- getNodeHeight, 116
- getNodePosition, 117
- getNodeProperty, 118
- getNodeSelectionColorDefault, 119
- getNodeShapes, 119, 291–293
- getNodeSize, 120
- getNodeWidth, 121
- getNotebookIsRunning, 121
- getSandboxReinitialize, 122
- getSelectedEdgeCount, 122
- getSelectedEdges, 123
- getSelectedNodeCount, 124
- getSelectedNodes, 124
- getStyleDependencies, 125
- getStyleMapping, 126
- getTableColumnNames, 127

- getTableColumns, 128
- getTableColumnTypes, 129
- getTableValue, 130
- getUninstalledApps, 131
- getVisualPropertyDefault, 131
- getVisualPropertyNames, 19, 21, 23, 98, 112, 118, 132, 132, 223, 244, 290
- getVisualStyleNames, 133
  
- hideAllPanels, 133
- hideEdges, 134, 137
- hideNodes, 135, 138
- hidePanel, 136
- hideSelectedEdges, 134, 136
- hideSelectedNodes, 135, 137
  
- importFilters, 138
- importNetworkFromFile, 139
- importNetworkFromNDEx, 139
- importVisualStyles, 140
- installApp, 141
- invertEdgeSelection, 141
- invertNodeSelection, 142
  
- layoutCopycat, 143
- layoutNetwork, 144
- listGroups, 145
- loadTableData, 145
- lockNodeDimensions, 146
  
- makeSimpleGraph, 147
- makeSimpleIgraph, 147
- mapTableColumn, 148
- mapVisualProperty, 149
- matchArrowColorToEdge, 151
- mergeNetworks, 152
  
- openAppStore, 153
- openSession, 154
  
- paletteColorBrewerAccent, 155
- paletteColorBrewerBlues, 155
- paletteColorBrewerBrBG, 156
- paletteColorBrewerBuGn, 156
- paletteColorBrewerBuPu, 157
- paletteColorBrewerDark2, 157
- paletteColorBrewerGnBu, 158
- paletteColorBrewerGreens, 159
- paletteColorBrewerGreys, 159
- paletteColorBrewerOranges, 160
- paletteColorBrewerOrRd, 160
- paletteColorBrewerPaired, 161
- paletteColorBrewerPastel1, 161
- paletteColorBrewerPastel2, 162
- paletteColorBrewerPiYG, 163
- paletteColorBrewerPRGn, 163
- paletteColorBrewerPuBu, 164
- paletteColorBrewerPuBuGn, 164
- paletteColorBrewerPuOr, 165
- paletteColorBrewerPuRd, 165
- paletteColorBrewerPurples, 166
- paletteColorBrewerRdBu, 167
- paletteColorBrewerRdPu, 167
- paletteColorBrewerRdYlBu, 168
- paletteColorBrewerReds, 168
- paletteColorBrewerSet1, 169
- paletteColorBrewerSet2, 169
- paletteColorBrewerSet3, 170
- paletteColorBrewerYlGn, 171
- paletteColorBrewerYlGnBu, 171
- paletteColorBrewerYlOrBr, 172
- paletteColorBrewerYlOrRd, 172
- paletteColorRandom, 173
  
- RCy3, 173
- RemoveFromGroup, 174
- removeNodeCustomGraphics, 175
- renameNetwork, 175
- renameTableColumn, 176
- resetDefaultSandbox, 177
- runningRemoteCheck, 178
  
- sandboxGetFileInfo, 178
- sandboxGetFrom, 179
- sandboxInitializer, 180
- sandboxRemove, 181
- sandboxRemoveFile, 181
- sandboxSendTo, 182
- sandboxSet, 183
- sandboxUrlTo, 184
- saveSession, 154, 185
- selectAllEdges, 186
- selectAllNodes, 186
- selectEdges, 187
- selectEdgesAdjacentToSelectedNodes, 188
- selectEdgesConnectingSelectedNodes, 189
- selectFirstNeighbors, 190

- selectNodes, [187](#), [190](#)
- selectNodesConnectedBySelectedEdges, [191](#)
- setBackground-color-default, [192](#)
- setCatchupFilterSecs, [193](#)
- setCatchupNetworkSecs, [193](#)
- setCurrentSandbox, [194](#)
- setCurrentView, [195](#)
- setDefaultSandbox, [196](#)
- setDefaultSandboxPath, [196](#)
- setEdgeColorBypass, [197](#)
- setEdgeColorDefault, [198](#)
- setEdgeColorMapping, [199](#)
- setEdgeFontFaceBypass, [200](#)
- setEdgeFontFaceDefault, [201](#)
- setEdgeFontFaceMapping, [201](#)
- setEdgeFontSizeBypass, [203](#)
- setEdgeFontSizeDefault, [204](#)
- setEdgeFontSizeMapping, [204](#)
- setEdgeLabelBypass, [206](#)
- setEdgeLabelColorBypass, [207](#)
- setEdgeLabelColorDefault, [208](#)
- setEdgeLabelColorMapping, [208](#)
- setEdgeLabelDefault, [210](#)
- setEdgeLabelMapping, [210](#)
- setEdgeLabelOpacityBypass, [211](#)
- setEdgeLabelOpacityDefault, [212](#)
- setEdgeLabelOpacityMapping, [213](#)
- setEdgeLineStyleBypass, [214](#)
- setEdgeLineStyleDefault, [215](#)
- setEdgeLineStyleMapping, [216](#)
- setEdgeLineWidthBypass, [217](#)
- setEdgeLineWidthDefault, [218](#)
- setEdgeLineWidthMapping, [218](#)
- setEdgeOpacityBypass, [220](#)
- setEdgeOpacityDefault, [221](#)
- setEdgeOpacityMapping, [221](#)
- setEdgePropertyBypass, [19](#), [134](#), [137](#), [197](#), [198](#), [200](#), [203](#), [206](#), [207](#), [212](#), [214](#), [217](#), [220](#), [223](#), [225](#), [229](#), [232](#), [236](#), [239](#)
- setEdgeSelectionColorDefault, [224](#)
- setEdgeSourceArrowColorBypass, [225](#)
- setEdgeSourceArrowColorDefault, [226](#)
- setEdgeSourceArrowColorMapping, [226](#)
- setEdgeSourceArrowMapping, [228](#)
- setEdgeSourceArrowShapeBypass, [229](#)
- setEdgeSourceArrowShapeDefault, [230](#)
- setEdgeSourceArrowShapeMapping, [230](#)
- setEdgeTargetArrowColorBypass, [231](#)
- setEdgeTargetArrowColorDefault, [232](#)
- setEdgeTargetArrowColorMapping, [233](#)
- setEdgeTargetArrowMapping, [234](#)
- setEdgeTargetArrowShapeBypass, [235](#)
- setEdgeTargetArrowShapeDefault, [236](#)
- setEdgeTargetArrowShapeMapping, [237](#)
- setEdgeTooltipBypass, [238](#)
- setEdgeTooltipDefault, [239](#)
- setEdgeTooltipMapping, [240](#)
- setLayoutProperties, [241](#)
- setModelPropagationSecs, [242](#)
- setNetworkCenterBypass, [243](#)
- setNetworkPropertyBypass, [243](#), [244](#), [245](#)
- setNetworkZoomBypass, [245](#)
- setNodeBorderColorBypass, [246](#)
- setNodeBorderColorDefault, [247](#)
- setNodeBorderColorMapping, [247](#)
- setNodeBorderOpacityBypass, [249](#)
- setNodeBorderOpacityDefault, [250](#)
- setNodeBorderOpacityMapping, [250](#)
- setNodeBorderWidthBypass, [252](#)
- setNodeBorderWidthDefault, [253](#)
- setNodeBorderWidthMapping, [253](#)
- setNodeColorBypass, [255](#)
- setNodeColorDefault, [256](#)
- setNodeColorMapping, [256](#)
- setNodeComboOpacityMapping, [257](#)
- setNodeCustomBarChart, [258](#)
- setNodeCustomBoxChart, [260](#)
- setNodeCustomHeatMapChart, [261](#)
- setNodeCustomLinearGradient, [262](#)
- setNodeCustomLineChart, [263](#)
- setNodeCustomPieChart, [264](#)
- setNodeCustomPosition, [265](#)
- setNodeCustomRadialGradient, [266](#)
- setNodeCustomRingChart, [267](#)
- setNodeFillOpacityBypass, [268](#)
- setNodeFillOpacityDefault, [270](#)
- setNodeFillOpacityMapping, [270](#)
- setNodeFontFaceBypass, [272](#)
- setNodeFontFaceDefault, [273](#)
- setNodeFontFaceMapping, [273](#)
- setNodeFontSizeBypass, [275](#)
- setNodeFontSizeDefault, [276](#)
- setNodeFontSizeMapping, [276](#)

setNodeHeightBypass, 278  
setNodeHeightDefault, 279  
setNodeHeightMapping, 279  
setNodeLabelBypass, 280  
setNodeLabelColorBypass, 281  
setNodeLabelColorDefault, 282  
setNodeLabelColorMapping, 283  
setNodeLabelDefault, 284  
setNodeLabelMapping, 285  
setNodeLabelOpacityBypass, 285  
setNodeLabelOpacityDefault, 286  
setNodeLabelOpacityMapping, 287  
setNodeOpacityBypass, 22, 288  
setNodePropertyBypass, 21, 23, 135, 138,  
246, 249, 252, 255, 269, 272, 275,  
278, 281, 282, 286, 289, 289, 291,  
292, 294, 297, 299  
setNodeSelectionColorDefault, 290  
setNodeShapeBypass, 291  
setNodeShapeDefault, 292  
setNodeShapeMapping, 293  
setNodeSizeBypass, 294  
setNodeSizeDefault, 295  
setNodeSizeMapping, 295  
setNodeTooltipBypass, 296  
setNodeTooltipDefault, 297  
setNodeTooltipMapping, 298  
setNodeWidthBypass, 299  
setNodeWidthDefault, 300  
setNodeWidthMapping, 300  
setNotebookIsRunning, 302  
setSandboxReinitialize, 302  
setStyleDependencies, 303  
setVisualPropertyDefault, 304  
setVisualStyle, 304  
spoofResponse (spoofResponse-class), 305  
spoofResponse-class, 305  
syncNodeCustomGraphicsSize, 305  
  
toggleGraphicsDetails, 306  
  
unhideAll, 134, 135, 137, 138, 307, 308, 309  
unhideEdges, 134, 137, 307, 308  
unhideNodes, 135, 138, 307, 309  
uninstallApp, 310  
updateApp, 310  
updateNetworkInNDEx, 311  
updateStyleDefaults, 312  
updateStyleMapping, 312