

Object						
new = \= == \== <> >< "" (abuttal) " " (blank) class copy defaultName hashCode hasMethod identityHash init instanceMethod instanceMethods isA, isInstanceOf objectName[=] request run setMethod start string unsetMethod	Alarm cancel init	ArgUtil validateClass validateLength validateNonNegative validateNumber validateNumberRange validatePosition validatePositive validateWhole validateWholeRange (no instance methods)	Array +OrderedCollection new of allIndexes allItems append at, [] dimension empty first hasIndex hasItem index isEmpty items last makeArray makeString next previous put, []= remove removeItem section size sort sortWith stableSort stableSortWith supplier toString	Class = \= == \== <> >< baseClass defaultName define delete enhanced id inherit isSubclassOf metaClass method methods mixinClass new queryMixinClass subclass subclasses superClass superClasses unInherit	Collection (M) allIndexes * allItems * at *, [] * difference hasIndex hasItem index * interSection items makeArray put *, []= * subSet supplier union xor MapCollection (M) makeArray putAll OrderedCollection (M) append * appendAll difference interSection subSet union xor SetCollection (M)	Comparable (M) compareTo *

Object (continued)									
Comparator (M) compare	CaseLessColumnComparator (M) compare init	CaseLessComparator (M) compare	CaseLessDescendingComparator (M) compare	ColumnComparator (M) compare init	DescendingComparator (M) compare	InvertingComparator (M) compare init	DateTime +Comparable fromBaseDate fromCivilTime fromEuropeanDate fromISOdate fromLongTime fromNormalDate fromNormalTime fromOrderedDate fromStandardDate fromTicks fromUSAdate maxDate minDate today + - = \= <> >< == \== > >= \> < <= \>> >>= << <=< \<< \>> addDays addHours addMicroSeconds addMinutes addSeconds addWeeks addYears baseDate civilTime compareTo convertDate convertTime date day dayMicroSeconds dayMinutes dayName daySeconds daysInMonth daysInYear elapsed europeanDate fullDate hashCode hours init isLeapYear ISOdate languageDate longTime microSeconds minutes month monthName normalDate normalTime orderedDate seconds standardDate string ticks timeOfDay USAdate weekDay year yearDay	Directory +MapCollection new allIndexes allItems at, [] empty entry hasEntry hasIndex hasItem index isEmpty items makeArray put, []= remove removeItem setEntry setMethod supplier unknown unSetMethod Properties load getLogical getProperty getWhole load put, []= save setLogical setProperty setWhole	InputStream (M) +InputStream +OutputStream Stream (M) arrayIn arrayOut charIn charOut chars close command description flush init lineIn lineOut lines makeArray open position qualify query say seek state string supplier unInit

Object (continued) ...							
new = \= == \== <> >< "" (abuttal) " " (blank) class copy defaultName hashCode hasMethod identityHash init instanceMethod instanceMethods isA, isInstanceOf objectName[=] request run setMethod start string unsetMethod	InputStream (M) arrayIn charIn * charOut (o) chars * close lineIn * lineOut (o) lines * open position (o)	List +OrderedCollection new of allIndexes allItems append at, [] empty first firstItem hasIndex hasItem index insert isEmpty items last lastItem makeArray next previous put, []= remove removeItem section supplier	Message new arguments completed errorCondition hasError messageName notify result send start target	Method new newFile isGuarded isPrivate isProtected setGuarded setPrivate setProtected setSecurityManager setUnguarded source	Monitor current destination init unknown	MutableBuffer new append delete getBufferSize insert lastPos length overlay pos request setBufferSize string subChar subStr unInit	OutputStream (M) arrayOut charIn (o) charOut * chars (o) close lineIn (o) lineOut * lines (o) open position (o)

Object (continued) ...							
Queue +OrderedCollection new of allIndexes allItems append at, [] empty first hasIndex hasItem index insert isEmpty items last makeArray next peek previous pull push put, []= queue remove removeItem supplier	RegularExpression <i>::requires rxregexp.cls</i> new init match parse pos pos unInit	Relation +MapCollection new allAt allIndex allIndexes allItems at, [] difference empty hasIndex hasItem index interSection isEmpty items makeArray put, []= remove removeItem subSet supplier union xor	RexxQueue create delete delete get init lineIn lineOut pull push queue queued say set	Stem +MapCollection new = \= == \== <> >< allIndexes allItems at, [] empty hasIndex hasItem index isEmpty items makeArray put, []= remove removeItem request supplier unknown	CircularQueue of init makeArray push queue resize size string supplier	Bag +SetCollection of difference interSection put, []= putAll subSet union xor	

(M) ... class created as a mixin class * ... abstract method (o) ... method is optional; if not implemented, raises syntax error (93.963)

(2007-11-05 09:59:38)

Note: Bold methods denote class methods and are listed first (*create, delete, new, newFile, of*)

Blue: new in 3.1.1 Red: new in 3.1.2 Green: 3.2.0

© 2007 Rony G. Flatscher

Object (continued) ...				
new = \= == \== <> >< "" (abuttal) " " (blank) class copy defaultName hashCode hasMethod identityHash init instanceMethod instanceMethods isA, isInstanceOf objectName[=] request run setMethod start string unsetMethod	String +Comparable			
	new + - * ** / // % \ = \= <> >< == \== > >= \> < <= \< >> >>= << <<= \<< \>> & && "" (abuttal) " " (blank) abbrev abs b2x bitAnd bitOr bitXor c2d c2x caselessAbbrev caselessChangeStr caselessCompare caselessCompareTo caselessCountStr caselessEquals	caselessLastPos caselessMatch caselessMatchChar caselessPos caselessWordPos center, centre changeStr compare compareTo copies countStr d2c d2x dataType decodeBase64 delStr delWord encodeBase64 equals format insert	lastPos left length lower makeArray makeString match matchChar max min overlay pos reverse right sign space strip subChar subStr subWord translate	trunc upper verify word wordIndex wordLength wordPos words x2b x2c x2d
	Supplier new allIndexes allItems available getArrays index init item next supplier StreamSupplier available index init item next			
	Table +MapCollection new allIndexes allItems at, [] empty hasIndex hasItem index isEmpty items makeArray put, []= remove removeItem supplier Set +SetCollection of interSection put, []= putAll next subSet union xor			

Object (continued)	
TimeSpan +Comparable	
fromCivilTime fromDays fromHours fromLongTime fromMicroSeconds fromMinutes fromNormalTime fromSeconds fromStringFormat + - * / // % \ = \= <> >< == \== > >= \> < <= \< >> >>= << <<= \<< \>> addDays addHours addMicroSeconds addMinutes addSeconds	addWeeks compareTo days duration hashCode hours init microSeconds minutes seconds sign string totalDays totalHours totalMicroSeconds totalMinutes totalSeconds

(M) ... class created as a mixin class

* ... abstract method

(2007-11-05 09:59:38)

Note: Bold methods denote class methods and are listed first (**new**, **of**)

Blue: new in 3.1.1 Red: new in 3.1.2 Green: 3.2.0

© 2007 Rony G. Flatscher

```

Object
Alarm
Array+OrderedCollection
ArgUtil
Class
Collection (M)
    MapCollection (M)
    OrderedCollection (M)
    SetCollection (M)
Comparable (M)
Comparator (M)
    CaselessColumnComparator (M)
    CaselessComparator (M)
    CaselessDescendingComparator (M)
    ColumnComparator (M)
    DescendingComparator (M)
    InvertingComparator (M)
DateTime+Comparable
Directory+MapCollection
    Properties
InputStream+OutputStream (M)+InputStream+OutputStream
    Stream (M)
InputStream (M)
List+OrderedCollection
Message
Method
Monitor
MutableBuffer
OutputStream (M)
Queue+OrderedCollection
    CircularQueue
Relation+MapCollection
    Bag+SetCollection
RexxQueue
Stem+MapCollection
String+Comparable
Supplier
    StreamSupplier
Table+MapCollection
    Set+SetCollection
TimeSpan+Comparable

```

Class Name (Alphabetically Sorted)	Immediate Superclass
Alarm	Object
Array	Object
ArgUtil	Object
Bag	Relation
CaselessColumnComparator (M)	Comparator
CaselessComparator (M)	Comparator
CaselessDescendingComparator (M)	Comparator
CircularQueue	Queue
Class	Object
Collection (M)	Object
ColumnComparator (M)	Comparator
Comparable (M)	Object
Comparator (M)	Object
DateTime	Object
DescendingComparator (M)	Comparator
Directory	Object
InputStream+OutputStream (M)	Object
InputStream (M)	Object
InvertingComparator (M)	Comparator
List	Object
MapCollection (M)	Collection
Message	Object
Method	Object
Monitor	Object
MutableBuffer	Object
Object	
OrderedCollection (M)	Collection
OutputStream (M)	Object
Properties	Directory
Queue	Object
Relation	Object
RexxQueue	Object
Set	Table
SetCollection (M)	Collection
Stem	Object
Stream (M)	InputStream+OutputStream
StreamSupplier	Supplier
String	Object
Supplier	Object
Table	Object
TimeSpan	Object

BIF: new option	DataType(val,"o")	returns .true, if val is a logical (Boolean) value, .false else
OLEObject: new info	getKnownMethods	returns additional information encoded as "!CoClassName" and "!CoClassDoc" ("!LibName" and "!LibDoc" now contain the correct information)
PARSE: new option	PARSE VAR x "<"tag">" content ("</"tag">")	allow any expression in parenthesis
PARSE: new option	parse value "01A0002CC" with len1 +2 val1 >(len1) len2 +2 val2 >(len2)>	allows true length match and usage of 0 length; len1=[01], val1=[A], len2=[00], val2=[]
PARSE: new option	parse value "1234567890" with "4" prefix <2	< extract chars between match position and previous characters (parses "23" from the string)
PARSE: new option	parse value "1234567890" with "4" prefix >2	> extract chars from match position up on (parses "45" from the string) length '0' would be allowed, in which case the empty string is assigned ('var <0' or 'var >0': var='')
CALL: new behaviour	CALL (any [expression] [allowed] [now])	allows any expression within parenthesis
PARSE: new option	PARSE VAR v a[3], o-attribute	allow assigning values to collections and object attributes
USE: new option	USE ARG a[3], o-attribute	allow assigning values to collections and object attributes
new BIF	lower(string[, [start] [, length])	new BIF (compatibility to Regina)
new BIF	upper(string[, [start] [, length])	new BIF (compatibility to Regina)
USE: new option	USE ARG var[=defaultValue] [, var[=defaultValue]...]	allow supplying a default value after the assignment operator "="
USE: new options	USE STRICT ARG [var[=defaultValue] [, var[=defaultValue] ...] [, ...]	strictly check [non-]existence of listed arguments, allow supplying default values (can be result of an expression, optional ellipsis (verbatimly!) at the very end to indicate possible optional values one implementation was: USE STRICT ARG var[=(default value)] [ASSERT (cond1 [, ...])][...] [, ...]
New keyword	LOOP	NetRexx like LOOP instruction, syntactically the same as the DO loop instructions
New feature	IF cond1, cond2[, ...] THEN WHEN cond1, cond2[, ...] THEN DO WHILE cond1, cond2[, ...] DO UNTIL cond1, cond2[, ...] LOOP WHILE cond1, cond2[, ...] LOOP UNTIL cond1, cond2[, ...]	evaluates conditions one by one, stopping evaluation whenever one condition yields .false evaluates conditions one by one, stopping evaluation whenever one condition yields .false evaluates conditions one by one, stopping evaluation whenever one condition yields .false evaluates conditions one by one, stopping evaluation whenever one condition yields .false evaluates conditions one by one, stopping evaluation whenever one condition yields .false evaluates conditions one by one, stopping evaluation whenever one condition yields .true
New labeling options	DO LABEL label1 LOOP LABEL label1 SELECT LABEL label1 ... END [label1]	optionally supply a label for a block, which then can be used to leave the block (e.g. LEAVE label1, or ITERATE label1 on DO/LOOP) END may optionally list the label for easier block matching
New directive option	::METHOD someName ABSTRACT	ABSTRACT option on method directive, documents that method must be implemented in subclass
Array's "PUT" and "[]=": new behaviour for multiple dimensional arrays: one can supply the index		values in a single array object
New "rexx.exe" startup switch '-e': allows to supply Rexx statements with optional arguments to be executed by the interpreter, e.g.:	rexx -e "say 'hi!' date('s') time()"	
New entry in .environment	.endOfLine	Operating system dependent line-end character sequence; 0a on Unix, 0d0a on Windows
TAB as whitespace	the TAB character ("09"x) is now treated like the space (blank) character	
New short-hand assignments: += -= /= //%= *= **= = = &= &&=		This is "syntax sugar", i.e., the interpreter will reformat it as a normal assignment.
New keywords on directives	"PRIVATE" on ::class and ::routine, "PUBLIC" and "UNPROTECTED" on ::method	
New directive "ATTRIBUTE"	::ATTRIBUTE someName [GET SET] [PUBLIC PRIVATE] [GUARDED UNGUARDED] [PROTECTED]	
New arg (caseless)changeStrchangeStr(old,new[,count]) caselessChangeStr(old,new[,count])		Optional third argument allows to determine up to how many changes are allowed to take place
New arg in changeStr()-BIF:changestr(old,string,new[,count])		Optional fourth argument allows to determine up to how many changes are allowed to take place
VALUE: new feature	value(".local")	Allows to evaluate environment symbols
New environment symbol	.LINE	Contains the line number currently being executed in the Rexx program.
DATE(), TIME(): new option 'F'ull time		returns the number of microseconds since "0001-01-01 00:00:00.000000" (0 microseconds); the date/time of "9999-12-31 23:59:59.999999" in microseconds: 31553625686399999999 (18 digits) returns number of seconds since 1970-01-01 (Unix epoch); compatible to Regina
TIME(): new option 'T'		
new error code „88“		allows to supply non-positional argument error messages by expecting the argument's name that is in error
improved handling of '.nil'.nil is allowed on the RHS of a comparison with a string object		
improved handling of concatenation of non-string objects		.Object now has the operator methods '', " " and ; requests string value from .object if necessary