

# MBS Real Studio DataTypes Plugin Documentation

Christian Schmitz

May 15, 2012

## 0.1 Introduction

This is the PDF version of the documentation for the Real Studio Plug-in from Monkeybread Software Germany. Plugin part: MBS Real Studio DataTypes Plugin

## 0.2 Content

- 1 List of all topics 3
- 2 All items in this plugin 23
- 6 List of all classes 305
- 7 List of all global methods 309

# Chapter 1

## List of Topics

• 2 Basic	23
– 2.1 class StringHandleMBS	23
* 2.1.1 Add(data as string)	24
* 2.1.1 clone as StringHandleMBS	24
* 2.1.1 Constructor	24
* 2.1.1 Constructor(initvalue as string)	25
* 2.1.1 Copy as string	25
* 2.1.1 Delete(start as integer, length as integer)	25
* 2.1.1 Extract(start as integer, length as integer) as string	26
* 2.1.1 Insert(data as string, position as integer)	26
* 2.1.1 InStr(srcOfs as Integer, target as String) as Integer	26
* 2.1.1 InStr(target as String) as Integer	27
* 2.1.1 Left(length as integer) as string	27
* 2.1.1 Mid(start as integer, length as integer) as string	28
* 2.1.1 Replace(a as String, b as string)	28
* 2.1.1 Replace(startpos as Integer, a as String, b as string)	29
* 2.1.1 ReplaceAll(a as String, b as string)	29
* 2.1.1 ReplaceAll(startpos as Integer, a as String, b as string)	30
* 2.1.1 Right(length as integer) as string	30
* 2.1.1 Truncate(length as integer)	31
* 2.1.2 BlockLen as Int64	32
* 2.1.2 BlockSize as Int64	32
* 2.1.2 Encoding as Int64	32
* 2.1.2 Len as Int64	33
* 2.1.2 ReplaceCount as Int64	33

– 2.2 class WeakRefMBS	33
* 2.2.1 close	34
* 2.2.2 Ref as StrongRefMBS	34
* 2.2.2 Value as object	34
– 2.3 class StrongRefMBS	35
* 2.3.1 close	36
* 2.3.1 NewWeakRef as WeakRefMBS	36
* 2.3.2 RefCount as integer	37
* 2.3.2 Value as object	37
• 3 Data Types	39
– 3.41 class StringToVariantOrderedMapMBS	247
* 3.41.1 Clear	248
* 3.41.1 Constructor	248
* 3.41.1 CountKey(key as string) as integer	249
* 3.41.1 find(key as string) as StringToVariantOrderedMapIteratorMBS	249
* 3.41.1 first as StringToVariantOrderedMapIteratorMBS	249
* 3.41.1 hasKey(key as string) as boolean	250
* 3.41.1 Key(index as integer) as string	250
* 3.41.1 Keys as string()	250
* 3.41.1 last as StringToVariantOrderedMapIteratorMBS	250
* 3.41.1 lookup(key as string, defaultvalue as variant) as variant	251
* 3.41.1 LowerBound(key as string) as StringToVariantOrderedMapIteratorMBS	252
* 3.41.1 Remove(first as StringToVariantOrderedMapIteratorMBS, last as StringToVariantOrderedMapIteratorMBS)	252
* 3.41.1 Remove(key as string) as integer	252
* 3.41.1 Remove(pos as StringToVariantOrderedMapIteratorMBS)	252
* 3.41.1 UpperBound(key as string) as StringToVariantOrderedMapIteratorMBS	253
* 3.41.1 value(key as string) as variant	253
* 3.41.1 ValueAtIndex(index as integer) as variant	253
* 3.41.1 Values as variant()	253
* 3.41.2 Count as Integer	254
* 3.41.2 Empty as Boolean	254
* 3.41.2 MaxSize as Integer	254
– 3.35 class StackSingleMBS	219
* 3.35.1 Bottom as single	219
* 3.35.1 clear	219
* 3.35.1 close	220
* 3.35.1 Contains(o as single) as boolean	220
* 3.35.1 Deep as integer	220
* 3.35.1 Pop as single	221

* 3.35.1 PopBottom as single	221
* 3.35.1 Push(o as single) as boolean	222
* 3.35.1 Top as single	222
* 3.35.2 IsEmpty as Boolean	222
– 3.2 class IntegerToVariantOrderedMapMBS	46
* 3.2.1 Clear	47
* 3.2.1 Constructor	47
* 3.2.1 CountKey(key as integer) as integer	47
* 3.2.1 find(key as integer) as IntegerToVariantOrderedMapIteratorMBS	47
* 3.2.1 first as IntegerToVariantOrderedMapIteratorMBS	47
* 3.2.1 hasKey(key as integer) as boolean	48
* 3.2.1 Key(index as integer) as integer	48
* 3.2.1 Keys as integer()	48
* 3.2.1 last as IntegerToVariantOrderedMapIteratorMBS	49
* 3.2.1 lookup(key as integer, defaultvalue as variant) as variant	49
* 3.2.1 LowerBound(key as integer) as IntegerToVariantOrderedMapIteratorMBS	50
* 3.2.1 Remove(first as IntegerToVariantOrderedMapIteratorMBS, last as IntegerToVariantOrderedMapIteratorMBS)	50
* 3.2.1 Remove(key as integer) as integer	50
* 3.2.1 Remove(pos as IntegerToVariantOrderedMapIteratorMBS)	51
* 3.2.1 UpperBound(key as integer) as IntegerToVariantOrderedMapIteratorMBS	51
* 3.2.1 value(key as integer) as variant	51
* 3.2.1 ValueAtIndex(index as integer) as variant	51
* 3.2.1 Values as variant()	51
* 3.2.2 Count as Integer	52
* 3.2.2 Empty as Boolean	52
* 3.2.2 MaxSize as Integer	53
– 3.37 class IntegerToIntegerHashMapMBS	230
* 3.37.1 Clear	231
* 3.37.1 Constructor	231
* 3.37.1 CountKey(key as integer) as integer	231
* 3.37.1 find(key as integer) as IntegerToIntegerHashMapIteratorMBS	231
* 3.37.1 first as IntegerToIntegerHashMapIteratorMBS	232
* 3.37.1 hasKey(key as integer) as boolean	232
* 3.37.1 Key(index as integer) as integer	232
* 3.37.1 Keys as integer()	233
* 3.37.1 last as IntegerToIntegerHashMapIteratorMBS	233
* 3.37.1 lookup(key as integer, defaultvalue as integer) as integer	234
* 3.37.1 Remove(first as IntegerToIntegerHashMapIteratorMBS, last as IntegerToIntegerHashMapIteratorMBS)	234
* 3.37.1 Remove(key as integer) as integer	234

* 3.37.1 Remove(pos as IntegerToIntegerHashMapIteratorMBS)	235
* 3.37.1 value(key as integer) as integer	235
* 3.37.1 ValueAtIndex(index as integer) as integer	235
* 3.37.1 Values as integer()	235
* 3.37.2 BinCount as Integer	236
* 3.37.2 Count as Integer	236
* 3.37.2 Empty as Boolean	237
* 3.37.2 MaxSize as Integer	237
– 3.36 class StringToStringHashMapMBS	223
* 3.36.1 Clear	224
* 3.36.1 Constructor	224
* 3.36.1 CountKey(key as string) as integer	224
* 3.36.1 find(key as string) as StringToStringHashMapIteratorMBS	224
* 3.36.1 first as StringToStringHashMapIteratorMBS	225
* 3.36.1 hasKey(key as string) as boolean	225
* 3.36.1 Key(index as integer) as string	225
* 3.36.1 Keys as string()	226
* 3.36.1 last as StringToStringHashMapIteratorMBS	226
* 3.36.1 lookup(key as string, defaultvalue as string) as string	227
* 3.36.1 Remove(first as StringToStringHashMapIteratorMBS, last as StringToStringHashMapIteratorMBS)	227
* 3.36.1 Remove(key as string) as integer	227
* 3.36.1 Remove(pos as StringToStringHashMapIteratorMBS)	228
* 3.36.1 value(key as string) as string	228
* 3.36.1 ValueAtIndex(index as integer) as string	228
* 3.36.1 Values as string()	228
* 3.36.2 BinCount as Integer	229
* 3.36.2 Count as Integer	229
* 3.36.2 Empty as Boolean	230
* 3.36.2 MaxSize as Integer	230
– 3.9 class IntegerOrderedSetMBS	75
* 3.9.1 Clear	76
* 3.9.1 Constructor	76
* 3.9.1 CountKey(key as integer) as integer	76
* 3.9.1 find(key as integer) as IntegerOrderedSetIteratorMBS	76
* 3.9.1 first as IntegerOrderedSetIteratorMBS	77
* 3.9.1 insert(key as integer)	77
* 3.9.1 Key(index as integer) as integer	77
* 3.9.1 Keys as integer()	78
* 3.9.1 last as IntegerOrderedSetIteratorMBS	78
* 3.9.1 lookup(key as integer) as boolean	79

* 3.9.1 LowerBound(key as integer) as IntegerOrderedSetIteratorMBS	79
* 3.9.1 Remove(first as IntegerOrderedSetIteratorMBS, last as IntegerOrderedSetIteratorMBS)	79
* 3.9.1 Remove(key as integer) as integer	80
* 3.9.1 Remove(pos as IntegerOrderedSetIteratorMBS)	80
* 3.9.1 UpperBound(key as integer) as IntegerOrderedSetIteratorMBS	80
* 3.9.2 Count as Integer	80
* 3.9.2 Empty as Boolean	81
* 3.9.2 MaxSize as Integer	81
– 3.30 class VariantToVariantOrderedMapMBS	193
* 3.30.1 Clear	193
* 3.30.1 Constructor	194
* 3.30.1 CountKey(key as variant) as integer	194
* 3.30.1 find(key as variant) as VariantToVariantMapIteratorMBS	194
* 3.30.1 first as VariantToVariantMapIteratorMBS	194
* 3.30.1 hasKey(key as variant) as boolean	195
* 3.30.1 Key(index as integer) as variant	195
* 3.30.1 Keys as variant()	195
* 3.30.1 last as VariantToVariantMapIteratorMBS	196
* 3.30.1 lookup(key as variant, defaultvalue as variant) as variant	196
* 3.30.1 LowerBound(key as variant) as VariantToVariantMapIteratorMBS	197
* 3.30.1 Remove(first as VariantToVariantMapIteratorMBS, last as VariantToVariantMapIteratorMBS)	197
* 3.30.1 Remove(key as variant) as integer	197
* 3.30.1 Remove(pos as VariantToVariantMapIteratorMBS)	197
* 3.30.1 UpperBound(key as variant) as VariantToVariantMapIteratorMBS	198
* 3.30.1 value(key as variant) as variant	198
* 3.30.1 ValueAtIndex(index as integer) as variant	198
* 3.30.1 Values as variant()	198
* 3.30.2 Count as Integer	199
* 3.30.2 Empty as Boolean	199
* 3.30.2 MaxSize as Integer	199
– 3.40 class VariantToVariantHashMapIteratorMBS	245
* 3.40.1 isEqual(other as VariantToVariantHashMapIteratorMBS) as boolean	245
* 3.40.1 isNotEqual(other as VariantToVariantHashMapIteratorMBS) as boolean	246
* 3.40.1 Key as variant	246
* 3.40.1 MoveNext	247
* 3.40.1 Value as variant	247
– 3.23 class ComplexSingleMBS	155
* 3.23.1 abs as single	155
* 3.23.1 Add(c as ComplexSingleMBS)	156

* 3.23.1 Add(x as single)	156
* 3.23.1 arg as single	156
* 3.23.1 conj as ComplexSingleMBS	156
* 3.23.1 Constructor(other as ComplexSingleMBS)	157
* 3.23.1 Constructor(x as single = 0.0, y as single = 0.0)	157
* 3.23.1 cos as ComplexSingleMBS	158
* 3.23.1 cosh as ComplexSingleMBS	158
* 3.23.1 Divide(c as ComplexSingleMBS)	158
* 3.23.1 Divide(x as single)	158
* 3.23.1 exp as ComplexSingleMBS	159
* 3.23.1 log as ComplexSingleMBS	159
* 3.23.1 log10 as ComplexSingleMBS	159
* 3.23.1 Multiply(c as ComplexSingleMBS)	159
* 3.23.1 Multiply(x as single)	160
* 3.23.1 norm as single	160
* 3.23.1 Operator_ Add(c as ComplexSingleMBS) as ComplexSingleMBS	160
* 3.23.1 Operator_ Add(x as single) as ComplexSingleMBS	161
* 3.23.1 Operator_ Compare(c as ComplexSingleMBS) as integer	161
* 3.23.1 Operator_ Divide(c as ComplexSingleMBS) as ComplexSingleMBS	161
* 3.23.1 Operator_ Divide(x as single) as ComplexSingleMBS	162
* 3.23.1 Operator_ Multiply(c as ComplexSingleMBS) as ComplexSingleMBS	162
* 3.23.1 Operator_ Multiply(x as single) as ComplexSingleMBS	163
* 3.23.1 Operator_ Power(x as ComplexSingleMBS) as ComplexSingleMBS	163
* 3.23.1 Operator_ Subtract(c as ComplexSingleMBS) as ComplexSingleMBS	163
* 3.23.1 Operator_ Subtract(x as single) as ComplexSingleMBS	164
* 3.23.1 PI as double	164
* 3.23.1 polar(rho as single, theta as single) as ComplexSingleMBS	164
* 3.23.1 pow(x as ComplexSingleMBS) as ComplexSingleMBS	165
* 3.23.1 pow(x as single) as ComplexSingleMBS	165
* 3.23.1 pow(x as single, y as ComplexSingleMBS) as ComplexSingleMBS	166
* 3.23.1 sin as ComplexSingleMBS	166
* 3.23.1 sinh as ComplexSingleMBS	166
* 3.23.1 sqrt as ComplexSingleMBS	167
* 3.23.1 str as string	167
* 3.23.1 Subtract(c as ComplexSingleMBS)	167
* 3.23.1 Subtract(x as single)	167
* 3.23.1 tan as ComplexSingleMBS	168
* 3.23.1 tanh as ComplexSingleMBS	168
* 3.23.2 Imag as single	168
* 3.23.2 Real as single	169
– 3.1 class IntegerToVariantHashMapMBS	39

* 3.1.1 Clear	40
* 3.1.1 Constructor	40
* 3.1.1 CountKey(key as integer) as integer	40
* 3.1.1 find(key as integer) as IntegerToVariantHashMapIteratorMBS	40
* 3.1.1 first as IntegerToVariantHashMapIteratorMBS	40
* 3.1.1 hasKey(key as integer) as boolean	41
* 3.1.1 Key(index as integer) as integer	41
* 3.1.1 Keys as integer()	41
* 3.1.1 last as IntegerToVariantHashMapIteratorMBS	42
* 3.1.1 lookup(key as integer, defaultvalue as variant) as variant	42
* 3.1.1 Remove(first as IntegerToVariantHashMapIteratorMBS, last as IntegerToVariantHashMapIteratorMBS)	43
* 3.1.1 Remove(key as integer) as integer	43
* 3.1.1 Remove(pos as IntegerToVariantHashMapIteratorMBS)	43
* 3.1.1 value(key as integer) as variant	44
* 3.1.1 ValueAtIndex(index as integer) as variant	44
* 3.1.1 Values as variant()	44
* 3.1.2 BinCount as Integer	45
* 3.1.2 Count as Integer	45
* 3.1.2 Empty as Boolean	46
* 3.1.2 MaxSize as Integer	46
– 3.18 class VariantToVariantHashMapMBS	128
* 3.18.1 Clear	128
* 3.18.1 Constructor	129
* 3.18.1 CountKey(key as variant) as integer	129
* 3.18.1 find(key as variant) as VariantToVariantHashMapIteratorMBS	129
* 3.18.1 first as VariantToVariantHashMapIteratorMBS	129
* 3.18.1 hasKey(key as variant) as boolean	130
* 3.18.1 Key(index as integer) as variant	130
* 3.18.1 Keys as variant()	130
* 3.18.1 last as VariantToVariantHashMapIteratorMBS	131
* 3.18.1 lookup(key as variant, defaultvalue as variant) as variant	131
* 3.18.1 Remove(first as VariantToVariantHashMapIteratorMBS, last as VariantToVariantHashMapIteratorMBS)	132
* 3.18.1 Remove(key as variant) as integer	132
* 3.18.1 Remove(pos as VariantToVariantHashMapIteratorMBS)	132
* 3.18.1 value(key as variant) as variant	132
* 3.18.1 ValueAtIndex(index as integer) as variant	133
* 3.18.1 Values as variant()	133
* 3.18.2 BinCount as Integer	133
* 3.18.2 Count as Integer	134
* 3.18.2 Empty as Boolean	134

* 3.18.2 MaxSize as Integer	135
– 3.8 class StringToVariantHashMapMBS	68
* 3.8.1 Clear	69
* 3.8.1 Constructor	69
* 3.8.1 CountKey(key as string) as integer	69
* 3.8.1 find(key as string) as StringToVariantHashMapIteratorMBS	69
* 3.8.1 first as StringToVariantHashMapIteratorMBS	70
* 3.8.1 hasKey(key as string) as boolean	70
* 3.8.1 Key(index as integer) as string	70
* 3.8.1 Keys as string()	71
* 3.8.1 last as StringToVariantHashMapIteratorMBS	71
* 3.8.1 lookup(key as string, defaultvalue as variant) as variant	72
* 3.8.1 Remove(first as StringToVariantHashMapIteratorMBS, last as StringToVariantHashMapIteratorMBS)	72
* 3.8.1 Remove(key as string) as integer	72
* 3.8.1 Remove(pos as StringToVariantHashMapIteratorMBS)	73
* 3.8.1 value(key as string) as variant	73
* 3.8.1 ValueAtIndex(index as integer) as variant	73
* 3.8.1 Values as variant()	73
* 3.8.2 BinCount as Integer	74
* 3.8.2 Count as Integer	74
* 3.8.2 Empty as Boolean	75
* 3.8.2 MaxSize as Integer	75
– 3.33 class IntegerToStringOrderedMapMBS	209
* 3.33.1 Clear	209
* 3.33.1 Constructor	209
* 3.33.1 CountKey(key as integer) as integer	210
* 3.33.1 find(key as integer) as IntegerToStringOrderedMapIteratorMBS	210
* 3.33.1 first as IntegerToStringOrderedMapIteratorMBS	210
* 3.33.1 hasKey(key as integer) as boolean	211
* 3.33.1 Key(index as integer) as integer	211
* 3.33.1 Keys as integer()	211
* 3.33.1 last as IntegerToStringOrderedMapIteratorMBS	211
* 3.33.1 lookup(key as integer, defaultvalue as string) as string	212
* 3.33.1 LowerBound(key as integer) as IntegerToStringOrderedMapIteratorMBS	213
* 3.33.1 Remove(first as IntegerToStringOrderedMapIteratorMBS, last as IntegerToStringOrderedMapIteratorMBS)	213
* 3.33.1 Remove(key as integer) as integer	213
* 3.33.1 Remove(pos as IntegerToStringOrderedMapIteratorMBS)	213
* 3.33.1 UpperBound(key as integer) as IntegerToStringOrderedMapIteratorMBS	214
* 3.33.1 value(key as integer) as string	214

* 3.33.1 ValueAtIndex(index as integer) as string	214
* 3.33.1 Values as string()	214
* 3.33.2 Count as Integer	215
* 3.33.2 Empty as Boolean	215
* 3.33.2 MaxSize as Integer	215
– 3.5 class StackStringMBS	59
* 3.5.1 Bottom as string	59
* 3.5.1 clear	60
* 3.5.1 close	60
* 3.5.1 Contains(o as string) as boolean	60
* 3.5.1 Deep as integer	61
* 3.5.1 Pop as string	61
* 3.5.1 PopBottom as string	61
* 3.5.1 Push(o as string) as boolean	62
* 3.5.1 Top as string	62
* 3.5.2 IsEmpty as Boolean	63
– 3.10 class IntegerHashSetMBS	81
* 3.10.1 Clear	82
* 3.10.1 Constructor	82
* 3.10.1 CountKey(key as integer) as integer	82
* 3.10.1 find(key as integer) as IntegerHashSetIteratorMBS	82
* 3.10.1 first as IntegerHashSetIteratorMBS	82
* 3.10.1 insert(key as integer)	83
* 3.10.1 Key(index as integer) as integer	83
* 3.10.1 Keys as integer()	83
* 3.10.1 last as IntegerHashSetIteratorMBS	84
* 3.10.1 lookup(key as integer) as boolean	85
* 3.10.1 Remove(first as IntegerHashSetIteratorMBS, last as IntegerHashSetIteratorMBS)	85
* 3.10.1 Remove(key as integer) as integer	85
* 3.10.1 Remove(pos as IntegerHashSetIteratorMBS)	86
* 3.10.2 BinCount as Integer	86
* 3.10.2 Count as Integer	86
* 3.10.2 Empty as Boolean	87
* 3.10.2 MaxSize as Integer	87
– 3.38 class StackDoubleMBS	237
* 3.38.1 Bottom as double	238
* 3.38.1 clear	238
* 3.38.1 close	239
* 3.38.1 Contains(o as double) as boolean	239
* 3.38.1 Deep as integer	239
* 3.38.1 Pop as double	240

* 3.38.1 PopBottom as double	240
* 3.38.1 Push(o as double) as boolean	240
* 3.38.1 Top as double	241
* 3.38.2 IsEmpty as Boolean	241
– 3.13 class VariantOrderedSetMBS	95
* 3.13.1 Clear	96
* 3.13.1 Constructor	96
* 3.13.1 CountKey(key as variant) as integer	96
* 3.13.1 find(key as variant) as VariantOrderedSetIteratorMBS	96
* 3.13.1 first as VariantOrderedSetIteratorMBS	96
* 3.13.1 insert(key as variant)	97
* 3.13.1 Key(index as integer) as variant	97
* 3.13.1 Keys as variant()	97
* 3.13.1 last as VariantOrderedSetIteratorMBS	98
* 3.13.1 lookup(key as variant) as boolean	98
* 3.13.1 LowerBound(key as variant) as VariantOrderedSetIteratorMBS	99
* 3.13.1 Remove(first as VariantOrderedSetIteratorMBS, last as VariantOrderedSetIteratorMBS)	99
* 3.13.1 Remove(key as variant) as integer	99
* 3.13.1 Remove(pos as VariantOrderedSetIteratorMBS)	99
* 3.13.1 UpperBound(key as variant) as VariantOrderedSetIteratorMBS	100
* 3.13.2 Count as Integer	100
* 3.13.2 Empty as Boolean	100
* 3.13.2 MaxSize as Integer	101
– 3.43 class StringToVariantHashMapIteratorMBS	258
* 3.43.1 isEqual(other as StringToVariantHashMapIteratorMBS) as boolean	258
* 3.43.1 isNotEqual(other as StringToVariantHashMapIteratorMBS) as boolean	259
* 3.43.1 Key as string	259
* 3.43.1 MoveNext	260
* 3.43.1 Value as variant	260
– 3.25 class IntegerToVariantOrderedMapIteratorMBS	172
* 3.25.1 isEqual(other as IntegerToVariantOrderedMapIteratorMBS) as boolean	173
* 3.25.1 isNotEqual(other as IntegerToVariantOrderedMapIteratorMBS) as boolean	173
* 3.25.1 Key as integer	174
* 3.25.1 MoveNext	174
* 3.25.1 MovePrev	175
* 3.25.1 Value as variant	175
– 3.11 class StackIntegerMBS	87
* 3.11.1 Bottom as integer	88
* 3.11.1 clear	88
* 3.11.1 close	89

* 3.11.1 Contains(o as integer) as boolean	89
* 3.11.1 Deep as integer	89
* 3.11.1 Pop as integer	90
* 3.11.1 PopBottom as integer	90
* 3.11.1 Push(o as integer) as boolean	90
* 3.11.1 Top as integer	91
* 3.11.2 IsEmpty as Boolean	91
– 3.19 class StringHashSetMBS	135
* 3.19.1 Clear	136
* 3.19.1 Constructor	136
* 3.19.1 CountKey(key as string) as integer	136
* 3.19.1 find(key as string) as StringHashSetIteratorMBS	136
* 3.19.1 first as StringHashSetIteratorMBS	136
* 3.19.1 insert(key as string)	137
* 3.19.1 Key(index as integer) as string	137
* 3.19.1 Keys as string()	137
* 3.19.1 last as StringHashSetIteratorMBS	138
* 3.19.1 lookup(key as string) as boolean	138
* 3.19.1 Remove(first as StringHashSetIteratorMBS, last as StringHashSetIteratorMBS)	139
* 3.19.1 Remove(key as string) as integer	139
* 3.19.1 Remove(pos as StringHashSetIteratorMBS)	139
* 3.19.2 BinCount as Integer	140
* 3.19.2 Count as Integer	140
* 3.19.2 Empty as Boolean	141
* 3.19.2 MaxSize as Integer	141
– 3.14 class ComplexDoubleMBS	101
* 3.14.1 abs as double	101
* 3.14.1 Add(c as ComplexDoubleMBS)	102
* 3.14.1 Add(x as double)	102
* 3.14.1 arg as double	102
* 3.14.1 conj as ComplexDoubleMBS	102
* 3.14.1 Constructor(other as ComplexDoubleMBS)	103
* 3.14.1 Constructor(x as double = 0.0, y as double = 0.0)	103
* 3.14.1 cos as ComplexDoubleMBS	103
* 3.14.1 cosh as ComplexDoubleMBS	104
* 3.14.1 Divide(c as ComplexDoubleMBS)	104
* 3.14.1 Divide(x as double)	104
* 3.14.1 exp as ComplexDoubleMBS	104
* 3.14.1 log as ComplexDoubleMBS	105
* 3.14.1 log10 as ComplexDoubleMBS	105
* 3.14.1 Multiply(c as ComplexDoubleMBS)	105

* 3.14.1 Multiply(x as double)	105
* 3.14.1 norm as double	106
* 3.14.1 Operator_ Add(c as ComplexDoubleMBS) as ComplexDoubleMBS	106
* 3.14.1 Operator_ Add(x as double) as ComplexDoubleMBS	106
* 3.14.1 Operator_ Compare(c as ComplexDoubleMBS) as integer	107
* 3.14.1 Operator_ Divide(c as ComplexDoubleMBS) as ComplexDoubleMBS	107
* 3.14.1 Operator_ Divide(x as double) as ComplexDoubleMBS	108
* 3.14.1 Operator_ Multiply(c as ComplexDoubleMBS) as ComplexDoubleMBS	108
* 3.14.1 Operator_ Multiply(x as double) as ComplexDoubleMBS	108
* 3.14.1 Operator_ Power(x as ComplexDoubleMBS) as ComplexDoubleMBS	109
* 3.14.1 Operator_ Subtract(c as ComplexDoubleMBS) as ComplexDoubleMBS	109
* 3.14.1 Operator_ Subtract(x as double) as ComplexDoubleMBS	110
* 3.14.1 PI as double	110
* 3.14.1 polar(rho as double, theta as double) as ComplexDoubleMBS	110
* 3.14.1 pow(x as ComplexDoubleMBS) as ComplexDoubleMBS	110
* 3.14.1 pow(x as double) as ComplexDoubleMBS	111
* 3.14.1 pow(x as double, y as ComplexDoubleMBS) as ComplexDoubleMBS	111
* 3.14.1 sin as ComplexDoubleMBS	112
* 3.14.1 sinh as ComplexDoubleMBS	112
* 3.14.1 sqrt as ComplexDoubleMBS	112
* 3.14.1 str as string	113
* 3.14.1 Subtract(c as ComplexDoubleMBS)	113
* 3.14.1 Subtract(x as double)	113
* 3.14.1 tan as ComplexDoubleMBS	114
* 3.14.1 tanh as ComplexDoubleMBS	114
* 3.14.2 Imag as Double	114
* 3.14.2 Real as Double	115
– 3.3 class IntegerToStringHashMapIteratorMBS	53
* 3.3.1 isEqual(other as IntegerToStringHashMapIteratorMBS) as boolean	54
* 3.3.1 isNotEqual(other as IntegerToStringHashMapIteratorMBS) as boolean	54
* 3.3.1 Key as integer	55
* 3.3.1 MoveNext	55
* 3.3.1 Value as string	56
– 3.21 class IntegerToIntegerOrderedMapMBS	144
* 3.21.1 Clear	144
* 3.21.1 Constructor	144
* 3.21.1 CountKey(key as integer) as integer	145
* 3.21.1 find(key as integer) as IntegerToIntegerOrderedMapIteratorMBS	145
* 3.21.1 first as IntegerToIntegerOrderedMapIteratorMBS	145
* 3.21.1 hasKey(key as integer) as boolean	146
* 3.21.1 Key(index as integer) as integer	146

* 3.21.1 Keys as integer()	146
* 3.21.1 last as IntegerToIntegerOrderedMapIteratorMBS	146
* 3.21.1 lookup(key as integer, defaultvalue as integer) as integer	147
* 3.21.1 LowerBound(key as integer) as IntegerToIntegerOrderedMapIteratorMBS	148
* 3.21.1 Remove(first as IntegerToIntegerOrderedMapIteratorMBS, last as IntegerToIntegerOrderedMapIteratorMBS)	148
* 3.21.1 Remove(key as integer) as integer	148
* 3.21.1 Remove(pos as IntegerToIntegerOrderedMapIteratorMBS)	148
* 3.21.1 UpperBound(key as integer) as IntegerToIntegerOrderedMapIteratorMBS	149
* 3.21.1 value(key as integer) as integer	149
* 3.21.1 ValueAtIndex(index as integer) as integer	149
* 3.21.1 Values as integer()	149
* 3.21.2 Count as Integer	150
* 3.21.2 Empty as Boolean	150
* 3.21.2 MaxSize as Integer	150
– 3.28 class IntegerToIntegerOrderedMapIteratorMBS	182
* 3.28.1 isEqual(other as IntegerToIntegerOrderedMapIteratorMBS) as boolean	183
* 3.28.1 isNotEqual(other as IntegerToIntegerOrderedMapIteratorMBS) as boolean	184
* 3.28.1 Key as integer	184
* 3.28.1 MoveNext	184
* 3.28.1 MovePrev	185
* 3.28.1 Value as integer	185
– 3.4 class IntegerToStringOrderedMapIteratorMBS	56
* 3.4.1 isEqual(other as IntegerToStringOrderedMapIteratorMBS) as boolean	56
* 3.4.1 isNotEqual(other as IntegerToStringOrderedMapIteratorMBS) as boolean	57
* 3.4.1 Key as integer	58
* 3.4.1 MoveNext	58
* 3.4.1 MovePrev	58
* 3.4.1 Value as string	59
– 3.29 class StringToStringOrderedMapMBS	185
* 3.29.1 Clear	186
* 3.29.1 Constructor	186
* 3.29.1 CountKey(key as string) as integer	187
* 3.29.1 find(key as string) as StringToStringOrderedMapIteratorMBS	187
* 3.29.1 first as StringToStringOrderedMapIteratorMBS	187
* 3.29.1 hasKey(key as string) as boolean	188
* 3.29.1 Key(index as integer) as string	188
* 3.29.1 Keys as string()	188
* 3.29.1 last as StringToStringOrderedMapIteratorMBS	188
* 3.29.1 lookup(key as string, defaultvalue as string) as string	189
* 3.29.1 LowerBound(key as string) as StringToStringOrderedMapIteratorMBS	190

* 3.29.1 Remove(first as StringToStringOrderedMapIteratorMBS, last as StringToStringOrderedMapIteratorMBS)	190
* 3.29.1 Remove(key as string) as integer	190
* 3.29.1 Remove(pos as StringToStringOrderedMapIteratorMBS)	190
* 3.29.1 UpperBound(key as string) as StringToStringOrderedMapIteratorMBS	191
* 3.29.1 value(key as string) as string	191
* 3.29.1 ValueAtIndex(index as integer) as string	191
* 3.29.1 Values as string()	191
* 3.29.2 Count as Integer	192
* 3.29.2 Empty as Boolean	192
* 3.29.2 MaxSize as Integer	192
– 3.15 class IntegerToStringHashMapMBS	115
* 3.15.1 Clear	116
* 3.15.1 Constructor	116
* 3.15.1 CountKey(key as integer) as integer	116
* 3.15.1 find(key as integer) as IntegerToStringHashMapIteratorMBS	116
* 3.15.1 first as IntegerToStringHashMapIteratorMBS	116
* 3.15.1 hasKey(key as integer) as boolean	117
* 3.15.1 Key(index as integer) as integer	117
* 3.15.1 Keys as integer()	117
* 3.15.1 last as IntegerToStringHashMapIteratorMBS	118
* 3.15.1 lookup(key as integer, defaultvalue as string) as string	118
* 3.15.1 Remove(first as IntegerToStringHashMapIteratorMBS, last as IntegerToStringHashMapIteratorMBS)	119
* 3.15.1 Remove(key as integer) as integer	119
* 3.15.1 Remove(pos as IntegerToStringHashMapIteratorMBS)	119
* 3.15.1 value(key as integer) as string	120
* 3.15.1 ValueAtIndex(index as integer) as string	120
* 3.15.1 Values as string()	120
* 3.15.2 BinCount as Integer	121
* 3.15.2 Count as Integer	121
* 3.15.2 Empty as Boolean	122
* 3.15.2 MaxSize as Integer	122
– 3.45 class StringOrderedSetMBS	263
* 3.45.1 Clear	264
* 3.45.1 Constructor	264
* 3.45.1 CountKey(key as string) as integer	264
* 3.45.1 find(key as string) as StringOrderedSetIteratorMBS	264
* 3.45.1 first as StringOrderedSetIteratorMBS	265
* 3.45.1 insert(key as string)	265
* 3.45.1 Key(index as integer) as string	265

* 3.45.1 Keys as string()	266
* 3.45.1 last as StringOrderedSetIteratorMBS	266
* 3.45.1 lookup(key as string) as boolean	267
* 3.45.1 LowerBound(key as string) as StringOrderedSetIteratorMBS	267
* 3.45.1 Remove(first as StringOrderedSetIteratorMBS, last as StringOrderedSetIteratorMBS)	267
* 3.45.1 Remove(key as string) as integer	268
* 3.45.1 Remove(pos as StringOrderedSetIteratorMBS)	268
* 3.45.1 UpperBound(key as string) as StringOrderedSetIteratorMBS	268
* 3.45.2 Count as Integer	268
* 3.45.2 Empty as Boolean	269
* 3.45.2 MaxSize as Integer	269
– 3.12 class IntegerToVariantHashMapIteratorMBS	92
* 3.12.1 isEqual(other as IntegerToVariantHashMapIteratorMBS) as boolean	93
* 3.12.1 isNotEqual(other as IntegerToVariantHashMapIteratorMBS) as boolean	93
* 3.12.1 Key as integer	94
* 3.12.1 MoveNext	94
* 3.12.1 Value as variant	95
– 3.6 class IntegerToIntegerHashMapIteratorMBS	63
* 3.6.1 isEqual(other as IntegerToIntegerHashMapIteratorMBS) as boolean	64
* 3.6.1 isNotEqual(other as IntegerToIntegerHashMapIteratorMBS) as boolean	65
* 3.6.1 Key as integer	65
* 3.6.1 MoveNext	65
* 3.6.1 Value as integer	66
– 3.20 class StringHashSetIteratorMBS	141
* 3.20.1 isEqual(other as StringHashSetIteratorMBS) as boolean	142
* 3.20.1 isNotEqual(other as StringHashSetIteratorMBS) as boolean	142
* 3.20.1 Key as string	143
* 3.20.1 MoveNext	143
– 3.22 class StackObjectMBS	151
* 3.22.1 Bottom as object	151
* 3.22.1 clear	151
* 3.22.1 close	152
* 3.22.1 Contains(o as object) as boolean	152
* 3.22.1 Deep as integer	152
* 3.22.1 Pop as object	153
* 3.22.1 PopBottom as object	153
* 3.22.1 Push(o as object) as boolean	153
* 3.22.1 Top as object	154
* 3.22.2 IsEmpty as Boolean	154
– 3.27 class StackVariantMBS	178

* 3.27.1 Bottom as variant	178
* 3.27.1 clear	179
* 3.27.1 close	179
* 3.27.1 Contains(o as variant) as boolean	179
* 3.27.1 Deep as integer	180
* 3.27.1 Pop as variant	180
* 3.27.1 PopBottom as variant	181
* 3.27.1 Push(o as variant) as boolean	181
* 3.27.1 Top as variant	181
* 3.27.2 IsEmpty as Boolean	182
– 3.7 Globals	66
* 3.7 FFTDoubleAbsMBS(x() as ComplexDoubleMBS, N as integer = -1) as double()	66
* 3.7 FFTDoubleAbsMBS(x() as double, N as integer = -1) as double()	66
* 3.7 FFTDoubleMBS(x() as ComplexDoubleMBS, N as integer = -1) as ComplexDoubleMBS()	67
* 3.7 FFTDoubleMBS(x() as double, N as integer = -1) as ComplexDoubleMBS()	67
* 3.7 FFTSingleAbsMBS(x() as ComplexSingleMBS, N as integer = -1) as single()	67
* 3.7 FFTSingleAbsMBS(x() as single, N as integer = -1) as single()	67
* 3.7 FFTSingleMBS(x() as ComplexSingleMBS, N as integer = -1) as ComplexSingleMBS()	68
* 3.7 FFTSingleMBS(x() as single, N as integer = -1) as ComplexSingleMBS()	68
– 3.16 class IntegerOrderedSetIteratorMBS	122
* 3.16.1 isEqual(other as IntegerOrderedSetIteratorMBS) as boolean	123
* 3.16.1 isNotEqual(other as IntegerOrderedSetIteratorMBS) as boolean	123
* 3.16.1 Key as integer	124
* 3.16.1 MoveNext	124
* 3.16.1 MovePrev	125
– 3.32 class VariantHashSetMBS	202
* 3.32.1 Clear	203
* 3.32.1 Constructor	203
* 3.32.1 CountKey(key as variant) as integer	203
* 3.32.1 find(key as variant) as VariantHashSetIteratorMBS	204
* 3.32.1 first as VariantHashSetIteratorMBS	204
* 3.32.1 insert(key as variant)	204
* 3.32.1 Key(index as integer) as variant	205
* 3.32.1 Keys as variant()	205
* 3.32.1 last as VariantHashSetIteratorMBS	205
* 3.32.1 lookup(key as variant) as boolean	206
* 3.32.1 Remove(first as VariantHashSetIteratorMBS, last as VariantHashSetIteratorMBS)	206
* 3.32.1 Remove(key as variant) as integer	207
* 3.32.1 Remove(pos as VariantHashSetIteratorMBS)	207

* 3.32.2 BinCount as Integer	207
* 3.32.2 Count as Integer	208
* 3.32.2 Empty as Boolean	208
* 3.32.2 MaxSize as Integer	208
– 3.34 class StringToVariantOrderedMapIteratorMBS	216
* 3.34.1 isEqual(other as StringToVariantOrderedMapIteratorMBS) as boolean	216
* 3.34.1 isNotEqual(other as StringToVariantOrderedMapIteratorMBS) as boolean	217
* 3.34.1 Key as string	217
* 3.34.1 MoveNext	218
* 3.34.1 MovePrev	218
* 3.34.1 Value as variant	218
– 3.31 class IntegerHashSetIteratorMBS	200
* 3.31.1 isEqual(other as IntegerHashSetIteratorMBS) as boolean	200
* 3.31.1 isNotEqual(other as IntegerHashSetIteratorMBS) as boolean	201
* 3.31.1 Key as integer	201
* 3.31.1 MoveNext	202
– 3.17 class VariantHashSetIteratorMBS	125
* 3.17.1 isEqual(other as VariantHashSetIteratorMBS) as boolean	126
* 3.17.1 isNotEqual(other as VariantHashSetIteratorMBS) as boolean	126
* 3.17.1 Key as variant	127
* 3.17.1 MoveNext	127
– 3.42 class VariantToVariantMapIteratorMBS	255
* 3.42.1 isEqual(other as VariantToVariantMapIteratorMBS) as boolean	255
* 3.42.1 isNotEqual(other as VariantToVariantMapIteratorMBS) as boolean	256
* 3.42.1 Key as variant	256
* 3.42.1 MoveNext	257
* 3.42.1 MovePrev	257
* 3.42.1 Value as variant	257
– 3.44 class StringToStringHashMapIteratorMBS	260
* 3.44.1 isEqual(other as StringToStringHashMapIteratorMBS) as boolean	261
* 3.44.1 isNotEqual(other as StringToStringHashMapIteratorMBS) as boolean	262
* 3.44.1 Key as string	262
* 3.44.1 MoveNext	262
* 3.44.1 Value as string	263
– 3.26 class VariantOrderedSetIteratorMBS	175
* 3.26.1 isEqual(other as VariantOrderedSetIteratorMBS) as boolean	176
* 3.26.1 isNotEqual(other as VariantOrderedSetIteratorMBS) as boolean	176
* 3.26.1 Key as variant	177
* 3.26.1 MoveNext	177
* 3.26.1 MovePrev	178

– 3.39 class <code>StringOrderedSetIteratorMBS</code>	242
* 3.39.1 <code>isEqual(other as StringOrderedSetIteratorMBS)</code> as boolean	242
* 3.39.1 <code>isNotEqual(other as StringOrderedSetIteratorMBS)</code> as boolean	243
* 3.39.1 <code>Key</code> as string	244
* 3.39.1 <code>MoveNext</code>	244
* 3.39.1 <code>MovePrev</code>	244
– 3.24 class <code>StringToStringOrderedMapIteratorMBS</code>	169
* 3.24.1 <code>isEqual(other as StringToStringOrderedMapIteratorMBS)</code> as boolean	170
* 3.24.1 <code>isNotEqual(other as StringToStringOrderedMapIteratorMBS)</code> as boolean	170
* 3.24.1 <code>Key</code> as string	171
* 3.24.1 <code>MoveNext</code>	171
* 3.24.1 <code>MovePrev</code>	172
* 3.24.1 <code>Value</code> as string	172
• 4 <b>Math</b>	271
– 4.1 class <code>UnsignedInteger64MBS</code>	271
* 4.1.1 <code>Constructor</code>	271
* 4.1.1 <code>Constructor(value as double)</code>	272
* 4.1.1 <code>Constructor(value as string)</code>	272
* 4.1.1 <code>DoubleValue</code> as double	273
* 4.1.1 <code>IntegerValue</code> as integer	273
* 4.1.1 <code>Is32bit</code> as boolean	273
* 4.1.1 <code>Maximum</code> as <code>UnsignedInteger64MBS</code>	274
* 4.1.1 <code>MemoryblockValue</code> as <code>Memoryblock</code>	274
* 4.1.1 <code>Operator_ Add(value as UnsignedInteger64MBS)</code> as <code>UnsignedInteger64MBS</code>	274
* 4.1.1 <code>Operator_ AddRight(value as double)</code> as <code>UnsignedInteger64MBS</code>	275
* 4.1.1 <code>Operator_ Compare(value as UnsignedInteger64MBS)</code> as integer	276
* 4.1.1 <code>Operator_ Convert</code> as double	276
* 4.1.1 <code>Operator_ Convert</code> as string	277
* 4.1.1 <code>Operator_ Convert(value as double)</code>	277
* 4.1.1 <code>Operator_ Convert(value as string)</code>	277
* 4.1.1 <code>Operator_ Divide(value as UnsignedInteger64MBS)</code> as <code>UnsignedInteger64MBS</code>	278
* 4.1.1 <code>Operator_ DivideRight(value as double)</code> as <code>UnsignedInteger64MBS</code>	278
* 4.1.1 <code>Operator_ IntegerDivide(value as UnsignedInteger64MBS)</code> as <code>UnsignedInteger64MBS</code>	279
* 4.1.1 <code>Operator_ IntegerDivideRight(value as integer)</code> as <code>UnsignedInteger64MBS</code>	279
* 4.1.1 <code>Operator_ Modulo(value as UnsignedInteger64MBS)</code> as <code>UnsignedInteger64MBS</code>	279
* 4.1.1 <code>Operator_ ModuloRight(value as double)</code> as <code>UnsignedInteger64MBS</code>	280
* 4.1.1 <code>Operator_ Multiply(value as UnsignedInteger64MBS)</code> as <code>UnsignedInteger64MBS</code>	280
* 4.1.1 <code>Operator_ MultiplyRight(value as double)</code> as <code>UnsignedInteger64MBS</code>	280
* 4.1.1 <code>Operator_ Negate(value as UnsignedInteger64MBS)</code> as <code>UnsignedInteger64MBS</code>	281

* 4.1.1 Operator_ Sub(value as UnsignedInteger64MBS) as UnsignedInteger64MBS	281
* 4.1.1 Operator_ SubRight(value as double) as UnsignedInteger64MBS	281
* 4.1.1 SignedInteger64 as SignedInteger64MBS	282
* 4.1.1 StringValue as string	282
* 4.1.2 HigherInteger as Integer	282
* 4.1.2 LowerInteger as Integer	283
– 4.2 class SignedInteger64MBS	283
* 4.2.1 Abs as SignedInteger64MBS	284
* 4.2.1 Constructor	284
* 4.2.1 Constructor(value as double)	284
* 4.2.1 Constructor(value as string)	285
* 4.2.1 DoubleValue as double	285
* 4.2.1 IntegerValue as integer	286
* 4.2.1 Is32bit as boolean	286
* 4.2.1 Maximum as SignedInteger64MBS	286
* 4.2.1 MemoryblockValue as Memoryblock	287
* 4.2.1 Minimum as SignedInteger64MBS	287
* 4.2.1 Operator_ Add(value as SignedInteger64MBS) as SignedInteger64MBS	287
* 4.2.1 Operator_ AddRight(doublevalue as double) as SignedInteger64MBS	288
* 4.2.1 Operator_ Compare(value as SignedInteger64MBS) as integer	289
* 4.2.1 Operator_ Convert as double	289
* 4.2.1 Operator_ Convert as string	290
* 4.2.1 Operator_ Convert(value as double)	290
* 4.2.1 Operator_ Convert(value as string)	290
* 4.2.1 Operator_ Divide(value as SignedInteger64MBS) as SignedInteger64MBS	291
* 4.2.1 Operator_ DivideRight(doublevalue as double) as SignedInteger64MBS	291
* 4.2.1 Operator_ IntegerDivide(value as SignedInteger64MBS) as SignedInteger64MBS	292
* 4.2.1 Operator_ IntegerDivideRight(integervalue as integer) as SignedInteger64MBS	292
* 4.2.1 Operator_ Modulo(value as SignedInteger64MBS) as SignedInteger64MBS	292
* 4.2.1 Operator_ ModuloRight(doublevalue as double) as SignedInteger64MBS	293
* 4.2.1 Operator_ Multiply(value as SignedInteger64MBS) as SignedInteger64MBS	293
* 4.2.1 Operator_ MultiplyRight(doublevalue as double) as SignedInteger64MBS	293
* 4.2.1 Operator_ Negate(value as SignedInteger64MBS) as SignedInteger64MBS	294
* 4.2.1 Operator_ Sub(value as SignedInteger64MBS) as SignedInteger64MBS	294
* 4.2.1 Operator_ SubRight(doublevalue as double) as SignedInteger64MBS	294
* 4.2.1 StringValue as string	295
* 4.2.1 UnsignedInteger64 as UnsignedInteger64MBS	295
* 4.2.2 HigherInteger as Integer	295
* 4.2.2 LowerInteger as Integer	296
* 4.2.2 Sign as Integer	296
• 5 Notifications	297

– 5.2 class NotificationMBS	298
* 5.2.1 Constructor(name as string = "", ref as variant = nil, tag as variant = nil)	299
* 5.2.1 RegisterReceiver(target as NotificationReceiverMBS, name as string = "", ref as variant = nil)	299
* 5.2.1 Send(name as string, ref as object = nil, tag as variant = nil)	300
* 5.2.1 Send(notification as NotificationMBS)	300
* 5.2.1 SendDelayed(name as string, ref as object = nil, tag as variant = nil)	301
* 5.2.1 SendDelayed(notification as NotificationMBS)	301
* 5.2.1 SendNotification	302
* 5.2.1 SendNotificationDelayed	302
* 5.2.1 UnregisterReceiver(target as NotificationReceiverMBS)	302
* 5.2.2 Name as String	303
* 5.2.2 Ref as Variant	303
* 5.2.2 Tag as Variant	303
– 5.1 class NotificationObserverMBS	297
* 5.1.1 Constructor(name as string = "", ref as object = nil, tag as variant = nil)	297
* 5.1.2 ReceivedNotification(name as string, ref as variant, tag as variant, notification as NotificationMBS)	298

# Chapter 2

## Basic

### 2.1 class StringHandleMBS

`class StringHandleMBS`

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for attaching strings together very fast.

**Example:**

```
dim s as StringHandleMBS
```

```
s=new StringHandleMBS
```

```
// Add some text
```

```
s.Add "Hello"
```

```
s.Add " "
```

```
s.Add "World"
```

```
// Insert a string
```

```
s.Insert " great",6
```

```
// check it
```

```
MsgBox s.Copy
```

```
// Delete the great from above
```

```
s.Delete 6,7
```

```
// check
```

```
MsgBox s.Copy
```

```
// Insert again  
s.Insert " great ",6
```

```
// check  
MsgBox s.Copy
```

```
// Now we extract the middle, so it's deleted  
MsgBox s.Extract(6,7)
```

```
// check again  
MsgBox s.Copy
```

**Notes:** The class initialized itself on the first use.

### 2.1.1 Methods

#### Add(data as string)

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds the string at the end of the current data.

**Notes:** Note that all strings added must have the same encoding.

#### clone as StringHandleMBS

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new StringHandleMBS object with the same content.

#### Constructor

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The constructor of this class.

See also:

- 2.1.1 Constructor(initvalue as string)

### Constructor(initvalue as string)

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The second constructor of this class which sets the value directly to the given REALbasic string.

#### Example:

*// An utility function you can define in a module:*

```
Function BinaryReplaceAll(s as string, a as string,b as string) As string
dim h as StringHandleMBS
```

```
h=new StringHandleMBS(s)
h.ReplaceAll(a,b)
```

```
Return h.Copy
End Function
```

See also:

- 2.1.1 Constructor

### Copy as string

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the content as a REALbasic string.

**Notes:** This string will have the encoding set in the encoding property.

### Delete(start as integer, length as integer)

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Deletes the bytes within the range.

#### Notes:

One based like RB's string functions.

The start and length parameters use bytes not charcters as unit. You can use the Copy method to get a RB

string for characterwise editing.

### **Extract(start as integer, length as integer) as string**

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a part of the string.

#### **Notes:**

One based and the returned part is removed form the string data.

The start and length parameters use bytes not charcters as unit. You can use the Copy method to get a RB string for characterwise editing.

### **Insert(data as string, position as integer)**

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Inserts the string data at the given byte position.

#### **Notes:**

One based.

Note that on Unicode the character position and the byte position are not equal!  
(On 16bit Unicode charpos=2\*bytepos and on US ASCII charpos=bytepos)

The position parameter uses bytes not characters as unit. You can use the Copy method to get a RB string for characterwise editing.

### **InStr(srcOfs as Integer, target as String) as Integer**

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Searches for a binary string inside the StringHandle.

#### **Notes:**

The same as InStr but with a second parameter to specify the start of the search inside the string handle.

The srcOfs parameter uses bytes not characters as unit. You can use the Copy method to get a RB string for characterwise editing.

See also:

- 2.1.1 InStr(target as String) as Integer

### InStr(target as String) as Integer

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Searches for a binary string inside the StringHandle.

**Example:**

```
// We test if RB returns the same values as the plugin

dim s as String
dim h as StringHandleMBS

s="Christian"
h=new StringHandleMBS

h.Add s

MsgBox "MBS: "+str(h.InStr("is"))+", RB: "+str(InStr(s,"is"))
MsgBox "MBS: "+str(h.InStr(5,"ia"))+", RB: "+str(InStr(5,s,"ia"))
MsgBox "MBS: "+str(h.InStr("xy"))+", RB: "+str(InStr(s,"xy"))
```

**Notes:**

Works like InstrB in RB which searches in binary mode.

Note that the string you use for searching must have the same encoding as the strings inside the stringhandle or you won't find stuff like "" (umlauts).

The search is case sensitive.

See also:

- 2.1.1 InStr(srcOfs as Integer, target as String) as Integer

### Left(length as integer) as string

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a copy of the first left bytes of the string.

**Notes:**

May return less strings if the stored string is not long enough.

The length parameter uses bytes not characters as unit. You can use the Copy method to get a RB string for characterwise editing.

### Mid(start as integer, length as integer) as string

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a part of the string.

#### Notes:

One based.

The length parameter uses bytes not characters as unit. You can use the Copy method to get a RB string for characterwise editing.

### Replace(a as String, b as string)

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Replaces the first string found with content of a with the content of b.

#### Example:

```
dim s as StringHandleMBS
s=new StringHandleMBS
s.Add "Hallo Leutle, Hellau"
s.Replace("H","h")
s.Replace("l","i")
MsgBox s.Copy+" " +str(s.Len)
```

#### Notes:

Note that all strings are compared binary and must have the same encoding.

Basicly this is just a call to instr, one to delete and one to insert.

See also:

- 2.1.1 Replace(startpos as Integer, a as String, b as string)

**Replace(startpos as Integer, a as String, b as string)**

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Replaces the first string found with content of a with the content of b.

**Notes:**

If you don't give a startpos parameter the call uses one and is equal to Replace(a,b).

Startpos is one based like all indexes in this class.

The startpos parameter uses bytes not characters as unit. You can use the Copy method to get a RB string for characterwise editing.

See also:

- 2.1.1 Replace(a as String, b as string)

28

**ReplaceAll(a as String, b as string)**

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Replaces all strings with content of a with the content of b.

**Example:**

```
dim s as StringHandleMBS
```

```
s=new StringHandleMBS
```

```
s.Add "Hallo Leutle, Hellau"
```

```
s.Replaceall("H","h")
```

```
s.Replaceall("l","i")
```

```
MsgBox s.Copy+" "+str(s.Len)
```

**Notes:**

Note that all strings are compared binary and must have the same encoding.

Basicly this is just a loop with calls to instr, to delete and to insert.

See also:

- 2.1.1 ReplaceAll(startpos as Integer, a as String, b as string)

30

**ReplaceAll(startpos as Integer, a as String, b as string)**

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Replaces all strings with content of a with the content of b.

**Notes:**

If you don't give a startpos parameter the call uses one and is equal to ReplaceAll(a,b). Startpos is one based like all indexes in this class.

The startpos parameter uses bytes not characters as unit. You can use the Copy method to get a RB string for characterwise editing.

See also:

- 2.1.1 ReplaceAll(a as String, b as string)

29

**Right(length as integer) as string**

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a copy of the first right bytes of the string.

**Example:**

```
// There was a small bug in the Right and the Mid function for the version 3.1 of this plugin.
// This test failed in 3.1, but works in 3.2:
```

```
dim Text as StringhandleMBS
dim Part as String
```

```
Text = New StringHandleMBS
Text.Add "." + chr(13) + chr(10)
```

```
Part = Text.Copy
' Now Part is ".<CR><LF>" which is correct
```

```
if lenb(Part)<>3 then
MsgBox "Failed on Copy "+str(lenb(Part))
end if
```

```
Part = Text.Right(1)
' Now Part is "<LF>" which is correct
```

```
if lenb(Part)<>1 then
MsgBox "Failed on Right(1) "+str(lenb(Part))
end if
```

```

Part = Text.Right(2)
' Now Part is "<CR><LF>" which is correct

if lenb(Part)<>2 then
MsgBox "Failed on Right(2)" +str(lenb(Part))
end if

Part = Text.Right(3)
' Now Part is "" which is wrong!

if lenb(Part)<>3 then
MsgBox "Failed on Right(3)" +str(lenb(Part))
end if

```

**Notes:**

May return less strings if the stored string is not long enough.  
The length parameter uses bytes not characters as unit. You can use the Copy method to get a RB string for characterwise editing.

**Truncate(length as integer)**

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Truncates the string handle content.

**Notes:**

Sets the length of the string back to the given value if it's greater.

The length parameter uses bytes not characters as unit. You can use the Copy method to get a RB string for characterwise editing.

## 2.1.2 Properties

### BlockLen as Int64

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The size of the memory currently used for this class.

**Notes:**

This value increases by BlockSize if more memory is needed.  
(Read only property)

### BlockSize as Int64

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The size of the blocks to allocate for storing the data.

**Notes:** (Read and Write property)

### Encoding as Int64

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The encoding to use for returned strings.

**Notes:**

Only useful on Realbasic 4.5 and newer.

Some example values for encoding:

MacRoman	0	Also for ASCII or binary data used.
WindowsLatin1	& h0500	ANSI codepage 1252
ISOLatin1	& h0201	ISO 8859-1
NextStepLatin	& h0B01	NextStep encoding
Unicode	& h0100	16 bit Unicode
UTF8	& h08000100	8 bit Unicode
Invalid	& hFFFFFFFF	(Binary)
Invalid	& hFFFF	(Binary)

(Read and Write property)

**Len as Int64**

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the len in bytes of the stored string.

**Notes:** (Read only property)

**ReplaceCount as Int64**

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** After a call to one of the Replace functions the number of items replaced.

**Notes:** (Read and Write property)

## 2.2 class WeakRefMBS

**class WeakRefMBS**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for a weak reference.

**Example:**

```
dim s as new StrongRefMBS
```

```
s.Value = window1
```

```
dim w as WeakRefMBS = s.NewWeakRef
```

```
MsgBox window(s.Value).title // shows title, so works
```

**Notes:**

This class was planned to be implemented as Theodore H. Smith wrote about such a class on the plugins list. I asked for such a class myself using Realbugs at the time of RB 3.5.

## 2.2.1 Methods

### close

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor: Releases the connection to the StrongRef.

#### Notes:

There is no need to call this method except you want to free all resources of this object now without waiting for Realbasic to do it for you.

(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

## 2.2.2 Properties

### Ref as StrongRefMBS

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The strong reference which this object points to.

#### Example:

```
dim s as new StrongRefMBS
```

```
s.Value = window1
```

```
dim w as WeakRefMBS = s.NewWeakRef
```

```
MsgBox window(w.Ref.Value).title
```

**Notes:** (Read only property)

### Value as object

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value of the strong reference.

#### Example:

```

dim s as new StrongRefMBS

s.Value = window1

dim w as WeakRefMBS = s.NewWeakRef

MsgBox window(s.Value).title // shows title, so works

```

**Notes:**

equal to this lines:

```

if me.ref=nil then
return nil
else
return me.ref.value
end if
(Read only property)

```

## 2.3 class StrongRefMBS

```
class StrongRefMBS
```

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for a strong Reference.

**Example:**

```

dim s as new StrongRefMBS

s.Value = window1

dim w as WeakRefMBS = s.NewWeakRef

MsgBox window(s.Value).title // shows title, so works

```

**Notes:**

You may want to have a reference to e.g. a Window in some other window. But if you use a property you keep a hard reference to this window and Realbasic will never be able to release the destination window. (Same for circular reference).

Now using a Strong Reference keep by the destination Window on itself you can weak reference the window and if the window is closed the reference's value is set to nil.

### 2.3.1 Methods

#### close

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor: Releases all connections to Weak References.

**Notes:**

There is no need to call this method except you want to free all resources of this object now without waiting for Realbasic to do it for you.

(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

#### NewWeakRef as WeakRefMBS

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new weak reference to the current strong reference.

**Example:**

```
dim s as new StrongRefMBS
```

```
s.Value = window1
```

```
dim w as WeakRefMBS = s.NewWeakRef
```

```
MsgBox window(s.Value).title // shows title, so works
```

**Notes:** Returns nil on any error.

## 2.3.2 Properties

### RefCount as integer

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts the number of weak references pointing to this object.

**Example:**

```
dim s as new StrongRefMBS
```

```
s.Value = window1
```

```
dim w as WeakRefMBS = s.NewWeakRef
```

```
MsgBox window(s.Value).title // shows title, so works
```

```
MsgBox str(s.RefCount)
```

**Notes:** (Read only property)

### Value as object

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value of the reference.

**Example:**

```
dim s as new StrongRefMBS
```

```
s.Value = window1
```

```
dim w as WeakRefMBS = s.NewWeakRef
```

```
MsgBox window(s.Value).title // shows title, so works
```

**Notes:**

The destination object.  
(Read and Write property)



## Chapter 3

# Data Types

### 3.1 class IntegerToVariantHashMapMBS

class IntegerToVariantHashMapMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An alternative dictionary class for a hash map with integers as keys and variants as values.

**Example:**

```
dim s as new IntegerToVariantHashMapMBS
```

```
s.Value(1)="Hello"
```

```
s.Value(2)="World"
```

```
MsgBox str(s.Count) // shows 2
```

```
MsgBox s.Value(1)+" "+s.Value(2) // shows "Hello World"
```

**Notes:** You can choose whether you want to keep the map ordered using the OrderedMap class or whether you prefer a higher speed with the HashMap class.

### 3.1.1 Methods

#### Clear

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all of the elements.

#### Constructor

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The default constructor.

#### CountKey(key as integer) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how often a key is used in this map.

#### find(key as integer) as IntegerToVariantHashMapIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds the key and returns an iterator.

**Notes:** Returns the same value as the last method if the item was not found.

#### first as IntegerToVariantHashMapIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the beginning of the map.

**Example:**

```
// Create a map
dim m as new IntegerToVariantHashMapMBS

m.value(1)=2
m.value(2)=4
```

```

m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToVariantHashMapIteratorMBS = m.first
dim e as IntegerToVariantHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend

```

### HasKey(key as integer) as boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns True if Key is in the map and False if it is not. Returns a Boolean.

### Key(index as integer) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

### Keys as integer()

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the keys as an array.

#### Example:

```

dim m as new IntegerToVariantHashMapMBS

m.Value(1)="Hello"
m.Value(2)="World"

for each v as integer in m.keys
MsgBox str(v)

```

`next`

**Notes:** The order is stable and matches the order returned by the Values method at least until the Dictionary is modified. Use this method with For Each to loop through all the keys.

### `last` as `IntegerToVariantHashMapIteratorMBS`

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the end of the map.

**Example:**

```
// Create a map
dim m as new IntegerToVariantHashMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToVariantHashMapIteratorMBS = m.first
dim e as IntegerToVariantHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

### `lookup(key as integer, defaultvalue as variant) as variant`

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Looks up the passed value of Key.

**Example:**

```
dim map as new IntegerToVariantHashMapMBS

map.value(10)=1
map.value(100)=2
```

```
map.value(1000)=3
```

```
MsgBox str(map.lookup(5,0)) // shows 0 as value is missing
```

```
MsgBox str(map.lookup(10,0)) // shows 1 as value is found
```

**Notes:** If Key is found, it returns the corresponding value. If Key is not found, it returns the passed defaultValue.

### Remove(first as IntegerToVariantHashMapIteratorMBS, last as IntegerToVariantHashMapIteratorMBS)

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all elements in a range.

See also:

- 3.1.1 Remove(key as integer) as integer 43
- 3.1.1 Remove(pos as IntegerToVariantHashMapIteratorMBS) 43

### Remove(key as integer) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element with the given key.

See also:

- 3.1.1 Remove(first as IntegerToVariantHashMapIteratorMBS, last as IntegerToVariantHashMapIteratorMBS) 43
- 3.1.1 Remove(pos as IntegerToVariantHashMapIteratorMBS) 43

### Remove(pos as IntegerToVariantHashMapIteratorMBS)

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element pointed to by the pos iterator.

See also:

- 3.1.1 Remove(first as IntegerToVariantHashMapIteratorMBS, last as IntegerToVariantHashMapIteratorMBS) 43

- 3.1.1 Remove(key as integer) as integer

43

### value(key as integer) as variant

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value associated with the given key.

#### Notes:

If you query for a key which does not exist, a KeyNotFoundException is raised.  
(Read and Write computed property)

### ValueAtIndex(index as integer) as variant

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value with the given index.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

### Values as variant()

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the values as an array

#### Example:

```
// Create a map
dim m as new IntegerToVariantHashMapMBS

m.Value(1)="Hello"
m.Value(2)="World"

for each v as Variant in m.Keys
  MsgBox v
next
```

**Notes:** The order is stable and matches the order returned by Keys at least until the Map is modified. Use this method with For Each to loop through all the values.

### 3.1.2 Properties

#### BinCount as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of bins the hash table uses.

**Example:**

```
dim v as new IntegerToVariantHashMapMBS
```

```
v.value(1)="Hello"  
v.value(2)="World"
```

```
MsgBox str(v.BinCount)
```

**Notes:**

This is a measure of the hash table size, independent of the number of items the Dictionary contains.  
(Read only property)

#### Count as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items in this map.

**Example:**

```
dim map as new IntegerToVariantHashMapMBS
```

```
map.Value(1)=3  
map.Value(2)=4
```

```
MsgBox str(map.Count)
```

**Notes:** (Read only property)

**Empty as Boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the size is zero.

**Notes:** (Read only property)

**MaxSize as Integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the largest possible size for this map.

**Notes:**

Value is -1 if no limit is defined.  
(Read only property)

**3.2 class IntegerToVariantOrderedMapMBS****class IntegerToVariantOrderedMapMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An alternative dictionary class for an ordered map with integers as keys and variants as values.

**Example:**

```
dim s as new IntegerToVariantOrderedMapMBS
```

```
s.Value(1)="Hello"
```

```
s.Value(2)="World"
```

```
MsgBox str(s.Count) // shows 2
```

```
MsgBox s.Value(1)+" "+s.Value(2) // shows "Hello World"
```

**Notes:** You can choose whether you want to keep the map ordered using the OrderedMap class or whether you prefer a higher speed with the HashMap class.

### 3.2.1 Methods

#### Clear

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all of the elements.

#### Constructor

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The default constructor.

#### CountKey(key as integer) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how often a key is used in this map.

#### find(key as integer) as IntegerToVariantOrderedMapIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds the key and returns an iterator.

**Notes:** Returns the same value as the last method if the item was not found.

#### first as IntegerToVariantOrderedMapIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the beginning of the map.

**Example:**

```
// Create a map
dim m as new IntegerToVariantOrderedMapMBS

m.value(1)=2
m.value(2)=4
```

```

m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToVariantOrderedMapIteratorMBS = m.first
dim e as IntegerToVariantOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend

```

### hasKey(key as integer) as boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns True if Key is in the map and False if it is not. Returns a Boolean.

### Key(index as integer) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

### Keys as integer()

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the keys as an array.

#### Example:

```

dim m as new IntegerToVariantOrderedMapMBS

m.Value(1)="Hello"
m.Value(2)="World"

for each v as integer in m.keys
MsgBox str(v)

```

next

**Notes:** The order is stable and matches the order returned by the Values method at least until the Dictionary is modified. Use this method with For Each to loop through all the keys.

### last as IntegerToVariantOrderedMapIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the end of the map.

**Example:**

```
// Create a map
dim m as new IntegerToVariantOrderedMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToVariantOrderedMapIteratorMBS = m.first
dim e as IntegerToVariantOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

### lookup(key as integer, defaultvalue as variant) as variant

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Looks up the passed value of Key.

**Example:**

```
dim map as new IntegerToVariantOrderedMapMBS

map.value(10)=1
map.value(100)=2
```

```
map.value(1000)=3
```

```
MsgBox str(map.lookup(5,0)) // shows 0 as value is missing
```

```
MsgBox str(map.lookup(10,0)) // shows 1 as value is found
```

**Notes:** If Key is found, it returns the corresponding value. If Key is not found, it returns the passed defaultValue.

### LowerBound(key as integer) as IntegerToVariantOrderedMapIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator for the first element whose key is not less than k.

### Remove(first as IntegerToVariantOrderedMapIteratorMBS, last as IntegerToVariantOrderedMapIteratorMBS)

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all elements in a range.

See also:

- 3.2.1 Remove(key as integer) as integer 50
- 3.2.1 Remove(pos as IntegerToVariantOrderedMapIteratorMBS) 51

### Remove(key as integer) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element with the given key.

See also:

- 3.2.1 Remove(first as IntegerToVariantOrderedMapIteratorMBS, last as IntegerToVariantOrderedMapIteratorMBS) 50
- 3.2.1 Remove(pos as IntegerToVariantOrderedMapIteratorMBS) 51

**Remove(pos as IntegerToVariantOrderedMapIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element pointed to by the pos iterator.

See also:

- 3.2.1 Remove(first as IntegerToVariantOrderedMapIteratorMBS, last as IntegerToVariantOrderedMapIteratorMBS) 50
- 3.2.1 Remove(key as integer) as integer 50

**UpperBound(key as integer) as IntegerToVariantOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator for the first element whose key is greater than k.

**value(key as integer) as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value associated with the given key.

**Notes:**

If you query for a key which does not exist, a KeyNotFoundException is raised.  
(Read and Write computed property)

**ValueAtIndex(index as integer) as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value with the given index.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

**Values as variant()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the values as an array

**Example:**

```
// Create a map
dim m as new IntegerToVariantOrderedMapMBS

m.Value(1)="Hello"
m.Value(2)="World"

for each v as Variant in m.Keys
MsgBox v
next
```

**Notes:** The order is stable and matches the order returned by Keys at least until the Map is modified. Use this method with For Each to loop through all the values.

### 3.2.2 Properties

#### Count as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items in this map.

**Example:**

```
dim map as new IntegerToVariantOrderedMapMBS

map.Value(1)=3
map.Value(2)=4

MsgBox str(map.Count)
```

**Notes:** (Read only property)

#### Empty as Boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the size is zero.

**Notes:** (Read only property)

### MaxSize as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the largest possible size for this map.

**Notes:**

Value is -1 if no limit is defined.  
(Read only property)

## 3.3 class IntegerToStringHashMapIteratorMBS

### class IntegerToStringHashMapIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The iterator for the IntegerToStringHashMap class.

**Example:**

```
// Create a map
dim m as new IntegerToStringHashMapMBS

m.value(1)="Hello"
m.value(2)="World"
m.value(3)="!"

// get iterators pointing to first and after last element
dim i as IntegerToStringHashMapIteratorMBS = m.first
dim e as IntegerToStringHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

### 3.3.1 Methods

**isEqual**(other as IntegerToStringHashMapIteratorMBS) as boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are equal.

**Example:**

```
// Create a map
dim m as new IntegerToStringHashMapMBS

m.value(1)="Hello"
m.value(2)="World"
m.value(3)="!"

// get iterators pointing to first and after last element
dim i as IntegerToStringHashMapIteratorMBS = m.first
dim e as IntegerToStringHashMapIteratorMBS = m.last

// Show all keys and values
while i.isEqual(e) = false
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

**isNotEqual**(other as IntegerToStringHashMapIteratorMBS) as boolean

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are not equal.

**Example:**

```
// Create a map
dim m as new IntegerToStringHashMapMBS

m.value(1)="Hello"
m.value(2)="World"
m.value(3)="!"

// get iterators pointing to first and after last element
dim i as IntegerToStringHashMapIteratorMBS = m.first
dim e as IntegerToStringHashMapIteratorMBS = m.last
```

```
// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

### Key as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

### MoveNext

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the next item.

#### Example:

```
// Create a map
dim m as new IntegerToStringHashMapMBS

m.value(1)="Hello"
m.value(2)="World"
m.value(3)="!"

// get iterators pointing to first and after last element
dim i as IntegerToStringHashMapIteratorMBS = m.first
dim e as IntegerToStringHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

**Value as string**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value of the current item in the iterator.

**Notes:** (Read and Write computed property)

**3.4 class IntegerToStringOrderedMapIteratorMBS****class IntegerToStringOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The iterator for the IntegerToStringOrderedMap class.

**Example:**

```
// Create a map
dim m as new IntegerToStringOrderedMapMBS

m.value(1)="Hello"
m.value(2)="World"
m.value(3)="!"

// get iterators pointing to first and after last element
dim i as IntegerToStringOrderedMapIteratorMBS = m.first
dim e as IntegerToStringOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +i.Value
  i.MoveNext
wend
```

**3.4.1 Methods****isEqual(other as IntegerToStringOrderedMapIteratorMBS) as boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are equal.

**Example:**

```
// Create a map
dim m as new IntegerToStringOrderedMapMBS

m.value(1)="Hello"
m.value(2)="World"
m.value(3)="!"

// get iterators pointing to first and after last element
dim i as IntegerToStringOrderedMapIteratorMBS = m.first
dim e as IntegerToStringOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isEqual(e) = false
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

**isNotEqual(other as IntegerToStringOrderedMapIteratorMBS) as boolean**

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are not equal.

**Example:**

```
// Create a map
dim m as new IntegerToStringOrderedMapMBS

m.value(1)="Hello"
m.value(2)="World"
m.value(3)="!"

// get iterators pointing to first and after last element
dim i as IntegerToStringOrderedMapIteratorMBS = m.first
dim e as IntegerToStringOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

**Key as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

**MoveNext**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the next item.

**Example:**

```
// Create a map
dim m as new IntegerToStringOrderedMapMBS

m.value(1)="Hello"
m.value(2)="World"
m.value(3)="!"

// get iterators pointing to first and after last element
dim i as IntegerToStringOrderedMapIteratorMBS = m.first
dim e as IntegerToStringOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

**MovePrev**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the previous item.

### Value as string

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value of the current item in the iterator.

**Notes:** (Read and Write computed property)

## 3.5 class StackStringMBS

### class StackStringMBS

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for a stack of strings.

**Example:**

```
dim s as new StackStringMBS
call s.Push "Hello"
MsgBox s.pop
```

### 3.5.1 Methods

#### Bottom as string

Plugin Version: 3.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the bottom item of the stack and returns the value.

**Example:**

```
dim s as new StackStringMBS
call s.Push "Hello"
MsgBox s.bottom
```

**clear**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Clears the stack.

**Example:**

```
dim s as new StackStringMBS
```

```
call s.Push "abc"
```

```
s.Clear
```

```
if s.IsEmpty then  
  MsgBox "OK"  
end if
```

**close**

Plugin Version: 3.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor.

**Notes:**

There is no need to call this method except you want to free all resources of this object now without waiting for Realbasic to do it for you.

(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

**Contains(o as string) as boolean**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if one of the string references on the stack is equal to the given double reference.

**Example:**

```
dim s as new StackStringMBS
```

```
call s.Push "Hello"
```

```
if s.Contains("Hello") then  
  MsgBox "found. OK"  
else  
  MsgBox "not found. Failed"  
end if
```

**Deep as integer**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how much items are on the stack.

**Example:**

```
dim s as new StackStringMBS
```

```
MsgBox str(s.Deep)
```

```
call s.Push "Hello"
```

```
MsgBox str(s.Deep)
```

**Pop as string**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes the top item of the stack and returns the value.

**Example:**

```
dim s as new StackStringMBS
```

```
call s.Push "Hello"
```

```
MsgBox s.pop
```

**Notes:** Returns "" on any error.

**PopBottom as string**

Plugin Version: 3.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes the bottom item of the stack and returns the value.

**Example:**

```
dim s as new StackStringMBS
call s.Push "Hello"
MsgBox s.PopBottom
```

### Push(o as string) as boolean

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Pushes a value on the stack.

**Example:**

```
dim s as new StackStringMBS
call s.Push "Hello"
```

**Notes:**

Returns true if successful.  
May fail on low memory.

### Top as string

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of the top item on the stack.

**Example:**

```
dim s as new StackStringMBS
call s.Push "Hello"
MsgBox s.Top
```

**Notes:** Returns "" on any error.

### 3.5.2 Properties

#### IsEmpty as Boolean

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the stack is empty.

**Example:**

```
dim s as new StackStringMBS
```

```
if s.IsEmpty then  
  MsgBox "IsEmpty ok"  
else  
  MsgBox "IsEmpty failed"  
end if
```

```
call s.Push "Hello"
```

```
if s.IsEmpty then  
  MsgBox "IsEmpty failed"  
else  
  MsgBox "IsEmpty ok"  
end if
```

**Notes:** (Read only property)

## 3.6 class IntegerToIntegerHashMapIteratorMBS

```
class IntegerToIntegerHashMapIteratorMBS
```

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The iterator for the IntegerToIntegerHashMap class.

**Example:**

```
// Create a map  
dim m as new IntegerToIntegerHashMapMBS  
  
m.value(1)=2  
m.value(2)=4
```

```

m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToIntegerHashMapIteratorMBS = m.first
dim e as IntegerToIntegerHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +str(i.Value)
i.MoveNext
wend

```

### 3.6.1 Methods

**isEqual**(other as IntegerToIntegerHashMapIteratorMBS) as boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are equal.

**Example:**

```

// Create a map
dim m as new IntegerToIntegerHashMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToIntegerHashMapIteratorMBS = m.first
dim e as IntegerToIntegerHashMapIteratorMBS = m.last

// Show all keys and values
while i.isEqual(e) = false
MsgBox str(i.Key)+" ->" +str(i.Value)
i.MoveNext
wend

```

**isNotEqual**(other as IntegerToIntegerHashMapIteratorMBS) as boolean

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are not equal.

**Example:**

```
// Create a map
dim m as new IntegerToIntegerHashMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToIntegerHashMapIteratorMBS = m.first
dim e as IntegerToIntegerHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +str(i.Value)
  i.MoveNext
wend
```

**Key as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

**MoveNext**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the next item.

**Example:**

```
// Create a map
dim m as new IntegerToIntegerHashMapMBS

m.value(1)=2
```

```

m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToIntegerHashMapIteratorMBS = m.first
dim e as IntegerToIntegerHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +str(i.Value)
i.MoveNext
wend

```

### Value as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value of the current item in the iterator.

**Notes:** (Read and Write computed property)

## 3.7 Globals

**FFTDoubleAbsMBS(x() as ComplexDoubleMBS, N as integer = -1) as double()**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Performs a Fast Fourier Transformation and applies abs operation on result.

**Notes:** If N is not provided, the plugin chooses a value.

See also:

- 3.7 FFTDoubleAbsMBS(x() as double, N as integer = -1) as double()

66

**FFTDoubleAbsMBS(x() as double, N as integer = -1) as double()**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Performs a Fast Fourier Transformation and applies abs operation on result.

**Notes:** If N is not provided, the plugin chooses a value.

See also:

- 3.7 FFTDoubleAbsMBS(x() as ComplexDoubleMBS, N as integer = -1) as double() 66

#### FFTDoubleMBS(x() as ComplexDoubleMBS, N as integer = -1) as ComplexDoubleMBS()

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Performs a Fast Fourier Transformation.

**Notes:** If N is not provided, the plugin chooses a value.

See also:

- 3.7 FFTDoubleMBS(x() as double, N as integer = -1) as ComplexDoubleMBS() 67

#### FFTDoubleMBS(x() as double, N as integer = -1) as ComplexDoubleMBS()

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Performs a Fast Fourier Transformation.

**Notes:** If N is not provided, the plugin chooses a value.

See also:

- 3.7 FFTDoubleMBS(x() as ComplexDoubleMBS, N as integer = -1) as ComplexDoubleMBS() 67

#### FFTSingleAbsMBS(x() as ComplexSingleMBS, N as integer = -1) as single()

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Performs a Fast Fourier Transformation and applies abs operation on result.

**Notes:** If N is not provided, the plugin chooses a value.

See also:

- 3.7 FFTSingleAbsMBS(x() as single, N as integer = -1) as single() 67

#### FFTSingleAbsMBS(x() as single, N as integer = -1) as single()

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Performs a Fast Fourier Transformation and applies abs operation on result.

**Notes:** If N is not provided, the plugin chooses a value.

See also:

- 3.7 FFTSingleAbsMBS(x() as ComplexSingleMBS, N as integer = -1) as single() 67

**FFTSingleMBS(x() as ComplexSingleMBS, N as integer = -1) as ComplexSingleMBS()**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Performs a Fast Fourier Transformation.

See also:

- 3.7 FFTSingleMBS(x() as single, N as integer = -1) as ComplexSingleMBS() 68

**FFTSingleMBS(x() as single, N as integer = -1) as ComplexSingleMBS()**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Performs a Fast Fourier Transformation.

**Notes:** If N is not provided, the plugin chooses a value.

See also:

- 3.7 FFTSingleMBS(x() as ComplexSingleMBS, N as integer = -1) as ComplexSingleMBS() 68

## 3.8 class StringToVariantHashMapMBS

**class StringToVariantHashMapMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An alternative dictionary class for a hash map with strings as keys and variants as values.

**Example:**

```
dim s as new StringToVariantHashMapMBS
```

```
s.Value("Test")="Hello"
```

```
s.Value("test")="World"
```

```
MsgBox str(s.Count) // shows 2
```

```
s.Value(ConvertEncoding("test",encodings.UTF16))="Just a"
```

```
s.Value(ConvertEncoding("Test",encodings.UTF16))="test"
```

```
MsgBox str(s.Count) // shows 4
```

**Notes:**

All string comparison is done binary. So "test" and "Test" will be different keys. Also the comparison does not check the string encoding, so "test" can refer to two items if one key has UTF8 and one key has UTF16 encoding.

You can choose whether you want to keep the map ordered using the `OrderedMap` class or whether you prefer a higher speed with the `HashMap` class.

### 3.8.1 Methods

#### Clear

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all of the elements.

#### Constructor

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The default constructor.

#### CountKey(key as string) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how often a key is used in this map.

#### find(key as string) as StringToVariantHashMapIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds the key and returns an iterator.

**Notes:** Returns the same value as the last method if the item was not found.

**first** as **StringToVariantHashMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the beginning of the map.

**Example:**

```
// Create a map
dim m as new StringToVariantHashMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToVariantHashMapIteratorMBS = m.first
dim e as StringToVariantHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +i.Value
  i.MoveNext
wend
```

**hasKey(key as string) as boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns True if Key is in the map and False if it is not. Returns a Boolean.

**Key(index as integer) as string**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an `OutOfBoundsException` error. The first item has the index zero.

**Keys as string()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the keys as an array.

**Example:**

```
dim m as new StringToVariantHashMapMBS
```

```
m.Value("1")="Hello"
m.Value("2")="World"
```

```
for each v as string in m.Keys
  MsgBox str(v)
next
```

**Notes:** The order is stable and matches the order returned by the Values method at least until the Dictionary is modified. Use this method with For Each to loop through all the keys.

**last as StringToVariantHashMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the end of the map.

**Example:**

```
// Create a map
dim m as new StringToVariantHashMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToVariantHashMapIteratorMBS = m.first
dim e as StringToVariantHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +i.Value
  i.MoveNext
wend
```

**lookup(key as string, defaultvalue as variant) as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Looks up the passed value of Key.

**Example:**

```
dim map as new StringToVariantHashMapMBS
```

```
map.value("a")="Hello"
map.value("b")="World"
map.value("c")="!"
```

```
MsgBox str(map.lookup("d","?")) // shows "?" as value is missing
MsgBox str(map.lookup("a","?")) // shows "Hello" as value is found
```

**Notes:** If Key is found, it returns the corresponding value. If Key is not found, it returns the passed defaultvalue.

**Remove(first as StringToVariantHashMapIteratorMBS, last as StringToVariantHashMapIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all elements in a range.

See also:

- 3.8.1 Remove(key as string) as integer 72
- 3.8.1 Remove(pos as StringToVariantHashMapIteratorMBS) 73

**Remove(key as string) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element with the given key.

See also:

- 3.8.1 Remove(first as StringToVariantHashMapIteratorMBS, last as StringToVariantHashMapIteratorMBS) 72
- 3.8.1 Remove(pos as StringToVariantHashMapIteratorMBS) 73

### Remove(pos as StringToVariantHashMapIteratorMBS)

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element pointed to by the pos iterator.

See also:

- 3.8.1 Remove(first as StringToVariantHashMapIteratorMBS, last as StringToVariantHashMapIteratorMBS) 72
- 3.8.1 Remove(key as string) as integer 72

### value(key as string) as variant

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value associated with the given key.

**Notes:**

If you query for a key which does not exist, a KeyNotFoundException is raised.  
(Read and Write computed property)

### ValueAtIndex(index as integer) as variant

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value with the given index.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

### Values as variant()

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the values as an array

**Example:**

```

dim m as new StringToVariantHashMapMBS

m.Value("1")="Hello"
m.Value("2")="World"

for each v as string in m.Values
  MsgBox str(v)
next

```

**Notes:** The order is stable and matches the order returned by Keys at least until the Map is modified. Use this method with For Each to loop through all the values.

### 3.8.2 Properties

#### BinCount as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of bins the hash table uses.

**Example:**

```

dim v as new StringToVariantHashMapMBS

v.value("1")="Hello"
v.value("2")="World"

MsgBox str(v.BinCount)

```

**Notes:**

This is a measure of the hash table size, independent of the number of items the Dictionary contains. (Read only property)

#### Count as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items in this map.

**Example:**

```

dim map as new StringToVariantHashMapMBS
map.Value("?")=?"
map.Value("Hello")="World"

MsgBox str(map.Count)

```

**Notes:** (Read only property)

**Empty as Boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the size is zero.

**Notes:** (Read only property)

**MaxSize as Integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the largest possible size for this map.

**Notes:**

Value is -1 if no limit is defined.  
(Read only property)

### 3.9 class IntegerOrderedSetMBS

**class IntegerOrderedSetMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An alternative dictionary class for an ordered set with integers.

**Example:**

```
dim s as new IntegerOrderedSetMBS
```

```
s.insert 1
```

```
s.insert 2
```

```
MsgBox str(s.Count) // shows 2
```

**Notes:** You can choose whether you want to keep the set ordered using the `OrderedSet` class or whether you prefer a higher speed with the `HashSet` class.

### 3.9.1 Methods

#### Clear

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all of the elements.

#### Constructor

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The default constructor.

#### CountKey(key as integer) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how often a key is used in this set.

#### find(key as integer) as IntegerOrderedSetIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds the key and returns an iterator.

**Notes:** Returns the same value as the last method if the item was not found.

**first as IntegerOrderedSetIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the beginning of the set.

**Example:**

```
// Create a map
dim m as new IntegerOrderedSetMBS

m.insert(1)
m.insert(2)
m.insert(3)

// get iterators pointing to first and after last element
dim i as IntegerOrderedSetIteratorMBS = m.first
dim e as IntegerOrderedSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)
  i.MoveNext
wend
```

**insert(key as integer)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds a value to the set.

**Key(index as integer) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an `OutOfBoundsException` error. The first item has the index zero.

**Keys as integer()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the keys as an array.

**Example:**

```
dim m as new IntegerOrderedSetMBS
```

```
m.insert(1)
```

```
m.insert(2)
```

```
for each v as integer in m.Keys
```

```
MsgBox str(v)
```

```
next
```

**Notes:** The order is stable and matches the order returned by the Values method at least until the Dictionary is modified. Use this method with For Each to loop through all the keys.

**last as IntegerOrderedSetIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the end of the set.

**Example:**

```
// Create a map
```

```
dim m as new IntegerOrderedSetMBS
```

```
m.insert(1)
```

```
m.insert(2)
```

```
m.insert(3)
```

```
// get iterators pointing to first and after last element
```

```
dim i as IntegerOrderedSetIteratorMBS = m.first
```

```
dim e as IntegerOrderedSetIteratorMBS = m.last
```

```
// Show all keys and values
```

```
while i.isNotEqual(e)
```

```
MsgBox str(i.Key)
```

```
i.MoveNext
```

```
wend
```

**lookup(key as integer) as boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Checks whether an element with the given key exists in this set.

**Example:**

```
dim set as new IntegerOrderedSetMBS

set.insert 1
set.insert 2

MsgBox str(set.lookup(3)) // shows false as value is missing
MsgBox str(set.lookup(1)) // shows true as value is found
```

**Notes:** Returns true if yes and false if no.

**LowerBound(key as integer) as IntegerOrderedSetIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator for the first element whose key is not less than k.

**Remove(first as IntegerOrderedSetIteratorMBS, last as IntegerOrderedSetIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all elements in a range.

See also:

- 3.9.1 Remove(key as integer) as integer 80
- 3.9.1 Remove(pos as IntegerOrderedSetIteratorMBS) 80

**Remove(key as integer) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element with the given key.

See also:

- 3.9.1 Remove(first as IntegerOrderedSetIteratorMBS, last as IntegerOrderedSetIteratorMBS) 79
- 3.9.1 Remove(pos as IntegerOrderedSetIteratorMBS) 80

**Remove(pos as IntegerOrderedSetIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element pointed to by the pos iterator.

See also:

- 3.9.1 Remove(first as IntegerOrderedSetIteratorMBS, last as IntegerOrderedSetIteratorMBS) 79
- 3.9.1 Remove(key as integer) as integer 80

**UpperBound(key as integer) as IntegerOrderedSetIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator for the first element whose key is greater than k.

**3.9.2 Properties****Count as Integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items in this set.

**Example:**

```
dim set as new IntegerOrderedSetMBS
```

```
set.insert 1
```

```
set.insert 2
```

MsgBox str(set.Count)

**Notes:** (Read only property)

### Empty as Boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the size is zero.

**Notes:** (Read only property)

### MaxSize as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the largest possible size for this set.

**Notes:**

Value is -1 if no limit is defined.  
(Read only property)

## 3.10 class IntegerHashSetMBS

**class IntegerHashSetMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An alternative dictionary class for a hash set with integers.

**Example:**

```
dim s as new IntegerHashSetMBS
```

```
s.insert 1
```

```
s.insert 2
```

```
MsgBox str(s.Count) // shows 2
```

**Notes:** You can choose whether you want to keep the set ordered using the `OrderedSet` class or whether you prefer a higher speed with the `HashSet` class.

### 3.10.1 Methods

#### Clear

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all of the elements.

#### Constructor

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The default constructor.

#### CountKey(key as integer) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how often a key is used in this set.

#### find(key as integer) as IntegerHashSetIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds the key and returns an iterator.

**Notes:** Returns the same value as the last method if the item was not found.

#### first as IntegerHashSetIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the beginning of the set.

**Example:**

```
// Create a map
dim m as new IntegerHashSetMBS

m.insert(1)
m.insert(2)
m.insert(3)

// get iterators pointing to first and after last element
dim i as IntegerHashSetIteratorMBS = m.first
dim e as IntegerHashSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)
  i.MoveNext
wend
```

**insert(key as integer)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds a value to the set.

**Key(index as integer) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

**Keys as integer()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the keys as an array.

**Example:**

```

dim m as new IntegerHashSetMBS

m.insert(1)
m.insert(2)

for each v as integer in m.Keys
  MsgBox str(v)
next

```

**Notes:** The order is stable and matches the order returned by the Values method at least until the Dictionary is modified. Use this method with For Each to loop through all the keys.

#### last as IntegerHashSetIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the end of the set.

#### Example:

```

// Create a map
dim m as new IntegerHashSetMBS

m.insert(1)
m.insert(2)
m.insert(3)

// get iterators pointing to first and after last element
dim i as IntegerHashSetIteratorMBS = m.first
dim e as IntegerHashSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)
  i.MoveNext
wend

```

**lookup(key as integer) as boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Checks whether an element with the given key exists in this set.

**Example:**

```
dim set as new IntegerHashSetMBS
```

```
set.insert 1
```

```
set.insert 2
```

```
MsgBox str(set.lookup(3)) // shows false as value is missing
```

```
MsgBox str(set.lookup(1)) // shows true as value is found
```

**Notes:** Returns true if yes and false if no.

**Remove(first as IntegerHashSetIteratorMBS, last as IntegerHashSetIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all elements in a range.

See also:

- 3.10.1 Remove(key as integer) as integer 85
- 3.10.1 Remove(pos as IntegerHashSetIteratorMBS) 86

**Remove(key as integer) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element with the given key.

See also:

- 3.10.1 Remove(first as IntegerHashSetIteratorMBS, last as IntegerHashSetIteratorMBS) 85
- 3.10.1 Remove(pos as IntegerHashSetIteratorMBS) 86

**Remove(pos as IntegerHashSetIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element pointed to by the pos iterator.

See also:

- 3.10.1 Remove(first as IntegerHashSetIteratorMBS, last as IntegerHashSetIteratorMBS) 85
- 3.10.1 Remove(key as integer) as integer 85

**3.10.2 Properties****BinCount as Integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of bins the hash table uses.

**Example:**

```
dim v as new IntegerHashSetMBS
```

```
v.insert 1
v.insert 5
```

```
MsgBox str(v.BinCount)
```

**Notes:**

This is a measure of the hash table size, independent of the number of items the Dictionary contains.  
(Read only property)

**Count as Integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items in this set.

**Example:**

```
dim set as new IntegerHashSetMBS
```

```
set.insert 1  
set.insert 2
```

```
MsgBox str(set.Count)
```

**Notes:** (Read only property)

### Empty as Boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the size is zero.

**Notes:** (Read only property)

### MaxSize as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the largest possible size for this set.

**Notes:**

Value is -1 if no limit is defined.  
(Read only property)

## 3.11 class StackIntegerMBS

### class StackIntegerMBS

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for a stack of integers.

**Example:**

```
dim s as new StackIntegerMBS
```

```
call s.Push 5
```

```
MsgBox str(S.Top)
```

### 3.11.1 Methods

#### Bottom as integer

Plugin Version: 3.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the bottom item of the stack and returns the value.

**Example:**

```
dim s as new StackIntegerMBS
```

```
call s.Push 5
```

```
MsgBox str(s.Bottom)
```

#### clear

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Clears the stack.

**Example:**

```
dim s as new StackIntegerMBS
```

```
call s.Push 5
```

```
s.Clear
```

```
if s.IsEmpty then
```

```
MsgBox "OK"
```

```
end if
```

**close**

Plugin Version: 3.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor.

**Notes:**

There is no need to call this method except you want to free all resources of this object now without waiting for Realbasic to do it for you.

(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

**Contains(o as integer) as boolean**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if one of the items on the stack is equal to the given integer value.

**Example:**

```
dim s as new StackIntegerMBS
```

```
call s.Push 5
```

```
if s.Contains(5) then
  MsgBox "found. OK"
```

```
else
```

```
  MsgBox "not found. Failed"
```

```
end if
```

**Deep as integer**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how much items are on the stack.

**Example:**

```
dim s as new StackIntegerMBS
```

```
MsgBox str(s.Deep)
```

```
call s.Push 5
```

```
MsgBox str(s.Deep)
```

### Pop as integer

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes the top item of the stack and returns the value.

**Example:**

```
dim s as new StackIntegerMBS
```

```
call s.Push 5
```

```
MsgBox str(s.pop)
```

**Notes:** Returns 0 on any error.

### PopBottom as integer

Plugin Version: 3.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes the bottom item of the stack and returns the value.

**Example:**

```
dim s as new StackIntegerMBS
```

```
call s.Push 5
```

```
MsgBox str(s.PopBottom)
```

### Push(o as integer) as boolean

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Pushes a value on the stack.

**Example:**

```
dim s as new StackIntegerMBS
```

```
call s.Push 5
```

**Notes:**

Returns true if successful.  
May fail on low memory.

**Top as integer**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of the top item on the stack.

**Example:**

```
dim s as new StackIntegerMBS
call s.Push 5
MsgBox str(S.Top)
```

**Notes:** Returns 0 on any error.

### 3.11.2 Properties

**IsEmpty as Boolean**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the stack is empty.

**Example:**

```
dim s as new StackIntegerMBS
if s.IsEmpty then
MsgBox "IsEmpty ok"
else
MsgBox "IsEmpty failed"
end if
```

```

call s.Push 5

if s.IsEmpty then
MsgBox "IsEmpty failed"
else
MsgBox "IsEmpty ok"
end if

```

**Notes:** (Read only property)

### 3.12 class IntegerToVariantHashMapIteratorMBS

class IntegerToVariantHashMapIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The iterator for the IntegerToVariantHashMap class.

**Example:**

```

// Create a map
dim m as new IntegerToVariantHashMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToVariantHashMapIteratorMBS = m.first
dim e as IntegerToVariantHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend

```

### 3.12.1 Methods

**isEqual**(other as IntegerToVariantHashMapIteratorMBS) as boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are equal.

**Example:**

```
// Create a map
dim m as new IntegerToVariantHashMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToVariantHashMapIteratorMBS = m.first
dim e as IntegerToVariantHashMapIteratorMBS = m.last

// Show all keys and values
while i.isEqual(e) = false
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

**isNotEqual**(other as IntegerToVariantHashMapIteratorMBS) as boolean

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are not equal.

**Example:**

```
// Create a map
dim m as new IntegerToVariantHashMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToVariantHashMapIteratorMBS = m.first
dim e as IntegerToVariantHashMapIteratorMBS = m.last
```

```
// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +i.Value
  i.MoveNext
wend
```

### Key as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

### MoveNext

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the next item.

#### Example:

```
// Create a map
dim m as new IntegerToVariantHashMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToVariantHashMapIteratorMBS = m.first
dim e as IntegerToVariantHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +i.Value
  i.MoveNext
wend
```

**Value as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value of the current item in the iterator.

**Notes:** (Read and Write computed property)

**3.13 class VariantOrderedSetMBS****class VariantOrderedSetMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An alternative dictionary class for an ordered set with variants.

**Example:**

```
dim s as new VariantOrderedSetMBS
```

```
s.insert "test"
```

```
s.insert "Test"
```

```
MsgBox str(s.Count) // shows 2
```

```
s.insert ConvertEncoding("test",encodings.UTF16)
```

```
s.insert ConvertEncoding("Test",encodings.UTF16)
```

```
MsgBox str(s.Count) // shows 4
```

**Notes:**

When using variants for keys, you may get better results if all keys used have the same data type to improve comparison.

All string comparison is done binary. So "test" and "Test" will be different keys. Also the comparison does not check the string encoding, so "test" can refer to two items if one key has UTF8 and one key has UTF16 encoding.

You can choose whether you want to keep the set ordered using the OrderedSet class or whether you prefer a higher speed with the HashSet class.

### 3.13.1 Methods

#### Clear

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all of the elements.

#### Constructor

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The default constructor.

#### CountKey(key as variant) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how often a key is used in this set.

#### find(key as variant) as VariantOrderedSetIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds the key and returns an iterator.

**Notes:** Returns the same value as the last method if the item was not found.

#### first as VariantOrderedSetIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the beginning of the set.

**Example:**

```
// Create a map
dim m as new VariantOrderedSetMBS

m.insert(1.0) // double
m.insert("Hello") // string
```

```

m.insert(3) // integer

// get iterators pointing to first and after last element
dim i as VariantOrderedSetIteratorMBS = m.first
dim e as VariantOrderedSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox i.Key
  i.MoveNext
wend

```

**insert(key as variant)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds a value to the set.

**Key(index as integer) as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

**Keys as variant()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the keys as an array.

**Example:**

```

dim m as new VariantOrderedSetMBS

m.insert("1")
m.insert("2")

for each v as variant in m.Keys
  MsgBox v

```

`next`

**Notes:** The order is stable and matches the order returned by the Values method at least until the Dictionary is modified. Use this method with For Each to loop through all the keys.

### `last` as `VariantOrderedSetIteratorMBS`

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the end of the set.

**Example:**

```
// Create a map
dim m as new VariantOrderedSetMBS

m.insert(1.0) // double
m.insert("Hello") // string
m.insert(3) // integer

// get iterators pointing to first and after last element
dim i as VariantOrderedSetIteratorMBS = m.first
dim e as VariantOrderedSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox i.Key
  i.MoveNext
wend
```

### `lookup(key as variant) as boolean`

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Checks whether an element with the given key exists in this set.

**Example:**

```
dim set as new VariantOrderedSetMBS

set.insert "Hello"
set.insert "World"
```

```
MsgBox str(set.lookup("missed")) // shows false as value is missing
MsgBox str(set.lookup("Hello")) // shows true as value is found
```

**Notes:** Returns true if yes and false if no.

### LowerBound(key as variant) as VariantOrderedSetIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator for the first element whose key is not less than k.

### Remove(first as VariantOrderedSetIteratorMBS, last as VariantOrderedSetIteratorMBS)

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all elements in a range.

See also:

- 3.13.1 Remove(key as variant) as integer 99
- 3.13.1 Remove(pos as VariantOrderedSetIteratorMBS) 99

### Remove(key as variant) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element with the given key.

See also:

- 3.13.1 Remove(first as VariantOrderedSetIteratorMBS, last as VariantOrderedSetIteratorMBS) 99
- 3.13.1 Remove(pos as VariantOrderedSetIteratorMBS) 99

### Remove(pos as VariantOrderedSetIteratorMBS)

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element pointed to by the pos iterator.

See also:

- 3.13.1 Remove(first as VariantOrderedSetIteratorMBS, last as VariantOrderedSetIteratorMBS) 99
- 3.13.1 Remove(key as variant) as integer 99

### UpperBound(key as variant) as VariantOrderedSetIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator for the first element whose key is greater than k.

## 3.13.2 Properties

### Count as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items in this set.

**Example:**

```
dim set as new VariantOrderedSetMBS
```

```
set.insert 1
```

```
set.insert 2
```

```
MsgBox str(set.Count)
```

**Notes:** (Read only property)

### Empty as Boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the size is zero.

**Notes:** (Read only property)

**MaxSize as Integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the largest possible size for this set.

**Notes:**

Value is -1 if no limit is defined.  
(Read only property)

## 3.14 class ComplexDoubleMBS

**class ComplexDoubleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for complex numbers.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
dim d as new ComplexDoubleMBS(4,7)
```

```
dim sum as ComplexDoubleMBS = c+d
```

```
MsgBox sum.str
```

### 3.14.1 Methods

**abs as double**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The absolute value of the complex number.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
MsgBox str(c.abs)
```

**Add(c as ComplexDoubleMBS)**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds the given value to this complex number.

See also:

- 3.14.1 Add(x as double) 102

**Add(x as double)**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds the given value to this complex number.

See also:

- 3.14.1 Add(c as ComplexDoubleMBS) 102

**arg as double**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return phase angle of complex.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
MsgBox str(c.arg)
```

**conj as ComplexDoubleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns complex conjugate.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
dim e as ComplexDoubleMBS = c.conj
MsgBox e.str
```

**Constructor(other as ComplexDoubleMBS)**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new complex number with the values from the given one.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
dim d as new ComplexDoubleMBS(c)
```

```
d.Add 1 // modify second object
MsgBox "d: "+d.str+", c: "+c.str
```

See also:

- 3.14.1 Constructor(x as double = 0.0, y as double = 0.0) 103

**Constructor(x as double = 0.0, y as double = 0.0)**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new complex number with the given values.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
MsgBox str(c.Real)+" "+str(c.Imag)
```

See also:

- 3.14.1 Constructor(other as ComplexDoubleMBS) 103

**cos as ComplexDoubleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return cosine of complex.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
MsgBox c.cos.str
```

**cosh as ComplexDoubleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return hyperbolic cosine of complex.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
MsgBox c.cosh.str
```

**Divide(c as ComplexDoubleMBS)**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Divides this complex number by the given complex number.

See also:

- 3.14.1 Divide(x as double) 104

**Divide(x as double)**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Divides this complex number by the given value.

See also:

- 3.14.1 Divide(c as ComplexDoubleMBS) 104

**exp as ComplexDoubleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return exponential of complex.

**Notes:**

```
dim c as new ComplexDoubleMBS(1,2)
MsgBox c.exp.str
```

**log as ComplexDoubleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return natural logarithm of complex.

**Example:**

```
dim c as new ComplexDoubleMBS(10,10)
MsgBox c.log.str
```

**log10 as ComplexDoubleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return common logarithm of complex.

**Example:**

```
dim c as new ComplexDoubleMBS(10,0)
MsgBox c.log10.str
```

**Multiply(c as ComplexDoubleMBS)**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Multiplies this complex number with the given value.

See also:

- 3.14.1 Multiply(x as double)

105

**Multiply(x as double)**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Multiplies this complex number with the given value.

See also:

- 3.14.1 Multiply(c as ComplexDoubleMBS)

105

### norm as double

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return norm of complex number.

#### Example:

```
dim c as new ComplexDoubleMBS(1,2)
MsgBox str(c.norm)
```

### Operator\_ Add(c as ComplexDoubleMBS) as ComplexDoubleMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This method is called by Real Studio in order to add two complex numbers.

#### Example:

```
dim c as new ComplexDoubleMBS(1,2)
dim d as new ComplexDoubleMBS(3,4)
dim e as ComplexDoubleMBS = c+d
MsgBox e.str
```

See also:

- 3.14.1 Operator\_ Add(x as double) as ComplexDoubleMBS

106

### Operator\_ Add(x as double) as ComplexDoubleMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This method is called by Real Studio in order to add a value to this complex number.

#### Example:

```
dim c as new ComplexDoubleMBS(3,4)
dim e as ComplexDoubleMBS = c+2
MsgBox e.str
```

See also:

- 3.14.1 Operator\_ Add(c as ComplexDoubleMBS) as ComplexDoubleMBS 106

### Operator\_ Compare(c as ComplexDoubleMBS) as integer

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Compares two complex numbers.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
dim d as new ComplexDoubleMBS(1,2)

if c = d then
  MsgBox "equal"
else
  MsgBox "not equal"
end if
```

### Operator\_ Divide(c as ComplexDoubleMBS) as ComplexDoubleMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This method is called by Real Studio in order to divide one complex number by another.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
dim d as new ComplexDoubleMBS(3,4)
dim e as ComplexDoubleMBS = c/d
MsgBox e.str
```

See also:

- 3.14.1 Operator\_ Divide(x as double) as ComplexDoubleMBS 108

**Operator\_ Divide(x as double) as ComplexDoubleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This method is called by Real Studio in order to divide a complex number by the given value.

**Example:**

```
dim c as new ComplexDoubleMBS(3,4)
dim e as ComplexDoubleMBS = c/2
MsgBox e.str
```

See also:

- 3.14.1 Operator\_ Divide(c as ComplexDoubleMBS) as ComplexDoubleMBS 107

**Operator\_ Multiply(c as ComplexDoubleMBS) as ComplexDoubleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This method is called by Real Studio in order to multiply two complex numbers.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
dim d as new ComplexDoubleMBS(3,4)
dim e as ComplexDoubleMBS = c*d
MsgBox e.str
```

See also:

- 3.14.1 Operator\_ Multiply(x as double) as ComplexDoubleMBS 108

**Operator\_ Multiply(x as double) as ComplexDoubleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This method is called by Real Studio in order to multiply a double to a complex number.

**Example:**

```
dim c as new ComplexDoubleMBS(3,4)
dim e as ComplexDoubleMBS = c*2
```

MsgBox e.str

See also:

- 3.14.1 Operator\_ Multiply(c as ComplexDoubleMBS) as ComplexDoubleMBS 108

### Operator\_ Power(x as ComplexDoubleMBS) as ComplexDoubleMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This method is called by Real Studio in order to calculate the power of two complex numbers.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
dim d as new ComplexDoubleMBS(3,4)
dim e as ComplexDoubleMBS = c^d
MsgBox e.str
```

### Operator\_ Subtract(c as ComplexDoubleMBS) as ComplexDoubleMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This method is called by Real Studio in order to subtract one complex number from another.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
dim d as new ComplexDoubleMBS(3,4)
dim e as ComplexDoubleMBS = c-d
MsgBox e.str
```

See also:

- 3.14.1 Operator\_ Subtract(x as double) as ComplexDoubleMBS 110

**Operator\_ Subtract(x as double) as ComplexDoubleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This method is called by Real Studio in order to subtract the given value from this complex number.

**Example:**

```
dim c as new ComplexDoubleMBS(3,4)
dim e as ComplexDoubleMBS = c-2
MsgBox e.str
```

See also:

- 3.14.1 Operator\_ Subtract(c as ComplexDoubleMBS) as ComplexDoubleMBS 109

**PI as double**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The PI constant.

**Example:**

```
MsgBox str(ComplexDoubleMBS.PI)
```

**polar(rho as double, theta as double) as ComplexDoubleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new complex number with the given polar coordinate.

**Example:**

```
MsgBox ComplexDoubleMBS.polar(10, 0.5).str
```

**pow(x as ComplexDoubleMBS) as ComplexDoubleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return complex power.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
dim d as new ComplexDoubleMBS(2,0)
dim m as ComplexDoubleMBS = c.pow(d)
```

```
MsgBox "c: "+c.str+EndOfLine+"d: "+d.str+EndOfLine+"c^d: "+m.str
```

See also:

- 3.14.1 pow(x as double) as ComplexDoubleMBS 111
- 3.14.1 pow(x as double, y as ComplexDoubleMBS) as ComplexDoubleMBS 111

### pow(x as double) as ComplexDoubleMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return complex power.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
dim m as ComplexDoubleMBS = c.pow(2)
```

```
MsgBox "c: "+c.str+EndOfLine+"c^2: "+m.str
```

See also:

- 3.14.1 pow(x as ComplexDoubleMBS) as ComplexDoubleMBS 110
- 3.14.1 pow(x as double, y as ComplexDoubleMBS) as ComplexDoubleMBS 111

### pow(x as double, y as ComplexDoubleMBS) as ComplexDoubleMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Calculates the power of the given values.

**Example:**

```
dim x as new ComplexDoubleMBS(2,3)
dim c as ComplexDoubleMBS = ComplexDoubleMBS.pow(2, x)
```

MsgBox c.str

See also:

- 3.14.1 pow(x as ComplexDoubleMBS) as ComplexDoubleMBS 110
- 3.14.1 pow(x as double) as ComplexDoubleMBS 111

### sin as ComplexDoubleMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return sine of complex.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
MsgBox c.sin.str
```

### sinh as ComplexDoubleMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return hyperbolic sine of complex.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
MsgBox c.sinh.str
```

### sqrt as ComplexDoubleMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return square root of complex.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
dim r as ComplexDoubleMBS = c.sqrt
```

```
dim m as ComplexDoubleMBS = r*r
```

```
MsgBox "number: "+c.str+EndOfLine+"root: "+r.str+EndOfLine+"back: "+m.str
```

### str as string

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Shows the number in an human readable format.

#### Example:

```
dim c as new ComplexDoubleMBS(1,2)
MsgBox c.str
```

**Notes:** The actual format can change.

### Subtract(c as ComplexDoubleMBS)

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Subtracts the given complex number from this complex number.

See also:

- 3.14.1 Subtract(x as double)

113

### Subtract(x as double)

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Subtracts the given value from this complex number.

See also:

- 3.14.1 Subtract(c as ComplexDoubleMBS)

113

### tan as ComplexDoubleMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return tangent of complex.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
MsgBox c.tan.str
```

### tanh as ComplexDoubleMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return hyperbolic tangent of complex.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
MsgBox c.tanh.str
```

## 3.14.2 Properties

### Imag as Double

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Get or set the imaginary part of the complex number.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
MsgBox str(c.Imag)
```

**Notes:** (Read and Write property)

### Real as Double

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Get or set the real part of the complex number.

**Example:**

```
dim c as new ComplexDoubleMBS(1,2)
MsgBox str(c.Real)
```

**Notes:** (Read and Write property)

## 3.15 class IntegerToStringHashMapMBS

### class IntegerToStringHashMapMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An alternative dictionary class for a hash map with integers as keys and strings as values.

**Example:**

```
dim s as new IntegerToStringHashMapMBS

s.Value(1)="Hello"
s.Value(2)="World"

MsgBox str(s.Count) // shows 2

MsgBox s.Value(1)+" "+s.Value(2) // shows "Hello World"
```

**Notes:** You can choose whether you want to keep the map ordered using the OrderedMap class or whether you prefer a higher speed with the HashMap class.

### 3.15.1 Methods

#### Clear

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all of the elements.

#### Constructor

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The default constructor.

#### CountKey(key as integer) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how often a key is used in this map.

#### find(key as integer) as IntegerToStringHashMapIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds the key and returns an iterator.

**Notes:** Returns the same value as the last method if the item was not found.

#### first as IntegerToStringHashMapIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the beginning of the map.

**Example:**

```
// Create a map
dim m as new IntegerToStringHashMapMBS

m.value(1)="Hello"
m.value(2)="World"
```

```

m.value(3)="!"

// get iterators pointing to first and after last element
dim i as IntegerToStringHashMapIteratorMBS = m.first
dim e as IntegerToStringHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +i.Value
  i.MoveNext
wend

```

### HasKey(key as integer) as boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns True if Key is in the map and False if it is not. Returns a Boolean.

### Key(index as integer) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

### Keys as integer()

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the keys as an array.

#### Example:

```

dim m as new IntegerToStringHashMapMBS

m.Value(1)="Hello"
m.Value(2)="World"

for each v as integer in m.keys
  MsgBox str(v)

```

`next`

**Notes:** The order is stable and matches the order returned by the Values method at least until the Dictionary is modified. Use this method with For Each to loop through all the keys.

### `last` as `IntegerToStringHashMapIteratorMBS`

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the end of the map.

**Example:**

```
// Create a map
dim m as new IntegerToStringHashMapMBS

m.value(1)="Hello"
m.value(2)="World"
m.value(3)="!"

// get iterators pointing to first and after last element
dim i as IntegerToStringHashMapIteratorMBS = m.first
dim e as IntegerToStringHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

### `lookup(key as integer, defaultvalue as string) as string`

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Looks up the passed value of Key.

**Example:**

```
dim map as new IntegerToStringOrderedMapMBS

map.value(10)="Hello"
map.value(100)="World"
```

```
map.value(1000)="!"
```

```
MsgBox str(map.lookup(5,"?")) // shows "?" as value is missing
MsgBox str(map.lookup(10,"?")) // shows "Hello" as value is found
```

**Notes:** If Key is found, it returns the corresponding value. If Key is not found, it returns the passed defaultValue.

### Remove(first as IntegerToStringHashMapIteratorMBS, last as IntegerToStringHashMapIteratorMBS)

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all elements in a range.

See also:

- 3.15.1 Remove(key as integer) as integer 119
- 3.15.1 Remove(pos as IntegerToStringHashMapIteratorMBS) 119

### Remove(key as integer) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element with the given key.

See also:

- 3.15.1 Remove(first as IntegerToStringHashMapIteratorMBS, last as IntegerToStringHashMapIteratorMBS) 119
- 3.15.1 Remove(pos as IntegerToStringHashMapIteratorMBS) 119

### Remove(pos as IntegerToStringHashMapIteratorMBS)

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element pointed to by the pos iterator.

See also:

- 3.15.1 Remove(first as IntegerToStringHashMapIteratorMBS, last as IntegerToStringHashMapIteratorMBS) 119

- 3.15.1 Remove(key as integer) as integer

119

### value(key as integer) as string

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value associated with the given key.

#### Notes:

If you query for a key which does not exist, a `KeyNotFoundException` is raised.  
(Read and Write computed property)

### ValueAtIndex(index as integer) as string

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value with the given index.

**Notes:** If there is no `Indexth` item in the map, a call generates an `OutOfBoundsException` error. The first item has the index zero.

### Values as string()

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the values as an array

#### Example:

```
dim m as new IntegerToStringHashMapMBS
```

```
m.Value(1)="Hello"
```

```
m.Value(2)="World"
```

```
for each v as string in m.Values
```

```
MsgBox v
```

```
next
```

**Notes:** The order is stable and matches the order returned by `Keys` at least until the `Map` is modified. Use this method with `For Each` to loop through all the values.

### 3.15.2 Properties

#### BinCount as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of bins the hash table uses.

**Example:**

```
dim v as new IntegerToStringHashMapMBS
```

```
v.value(1)="Hello"  
v.value(2)="World"
```

```
MsgBox str(v.BinCount)
```

**Notes:**

This is a measure of the hash table size, independent of the number of items the Dictionary contains. (Read only property)

#### Count as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items in this map.

**Example:**

```
dim map as new IntegerToStringHashMapMBS
```

```
map.Value(1)="Hello"  
map.Value(2)="World"
```

```
MsgBox str(map.Count)
```

**Notes:** (Read only property)

**Empty as Boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the size is zero.

**Notes:** (Read only property)

**MaxSize as Integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the largest possible size for this map.

**Notes:**

Value is -1 if no limit is defined.

(Read only property)

**3.16 class IntegerOrderedSetIteratorMBS****class IntegerOrderedSetIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The iterator for the IntegerOrderedSet class.

**Example:**

```
// Create a map
dim m as new IntegerOrderedSetMBS

m.insert(1)
m.insert(2)
m.insert(3)

// get iterators pointing to first and after last element
dim i as IntegerOrderedSetIteratorMBS = m.first
dim e as IntegerOrderedSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)
  i.MoveNext
wend
```

### 3.16.1 Methods

**isEqual**(other as IntegerOrderedSetIteratorMBS) as boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are equal.

**Example:**

```
// Create a map
dim m as new IntegerOrderedSetMBS

m.insert(1)
m.insert(2)
m.insert(3)

// get iterators pointing to first and after last element
dim i as IntegerOrderedSetIteratorMBS = m.first
dim e as IntegerOrderedSetIteratorMBS = m.last

// Show all keys and values
while i.isEqual(e) = false
MsgBox str(i.Key)
i.MoveNext
wend
```

**isNotEqual**(other as IntegerOrderedSetIteratorMBS) as boolean

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are not equal.

**Example:**

```
// Create a map
dim m as new IntegerOrderedSetMBS

m.insert(1)
m.insert(2)
m.insert(3)
```

```
// get iterators pointing to first and after last element
dim i as IntegerOrderedSetIteratorMBS = m.first
dim e as IntegerOrderedSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)
  i.MoveNext
wend
```

### Key as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

### MoveNext

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the next item.

#### Example:

```
// Create a map
dim m as new IntegerOrderedSetMBS

m.insert(1)
m.insert(2)
m.insert(3)

// get iterators pointing to first and after last element
dim i as IntegerOrderedSetIteratorMBS = m.first
dim e as IntegerOrderedSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)
  i.MoveNext
wend
```

### MovePrev

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the previous item.

## 3.17 class VariantHashSetIteratorMBS

### class VariantHashSetIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The iterator for the VariantHashSet class.

#### Example:

```
// Create a map
dim m as new VariantHashSetMBS

m.insert(1.0) // double
m.insert("Hello") // string
m.insert(3) // integer

// get iterators pointing to first and after last element
dim i as VariantHashSetIteratorMBS = m.first
dim e as VariantHashSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox i.Key
  i.MoveNext
wend
```

### 3.17.1 Methods

**isEqual**(other as VariantHashSetIteratorMBS) as boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are equal.

**Example:**

```
// Create a map
dim m as new VariantHashSetMBS

m.insert(1.0) // double
m.insert("Hello") // string
m.insert(3) // integer

// get iterators pointing to first and after last element
dim i as VariantHashSetIteratorMBS = m.first
dim e as VariantHashSetIteratorMBS = m.last

// Show all keys and values
while i.isEqual(e) = false
MsgBox i.Key
i.MoveNext
wend
```

**isNotEqual**(other as VariantHashSetIteratorMBS) as boolean

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are not equal.

**Example:**

```
// Create a map
dim m as new VariantHashSetMBS

m.insert(1.0) // double
m.insert("Hello") // string
m.insert(3) // integer

// get iterators pointing to first and after last element
dim i as VariantHashSetIteratorMBS = m.first
dim e as VariantHashSetIteratorMBS = m.last
```

```
// Show all keys and values
while i.isNotEqual(e)
  MsgBox i.Key
  i.MoveNext
wend
```

### Key as variant

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

### MoveNext

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the next item.

#### Example:

```
// Create a map
dim m as new VariantHashSetMBS

m.insert(1.0) // double
m.insert("Hello") // string
m.insert(3) // integer

// get iterators pointing to first and after last element
dim i as VariantHashSetIteratorMBS = m.first
dim e as VariantHashSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox i.Key
  i.MoveNext
wend
```

## 3.18 class VariantToVariantHashMapMBS

### class VariantToVariantHashMapMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An alternative dictionary class for an hash map with variants and keys and values.

#### Example:

```
dim s as new VariantToVariantHashMapMBS

s.Value("Test")="Hello"
s.Value("test")="World"

MsgBox str(s.Count) // shows 2

s.Value(ConvertEncoding("test",encodings.UTF16))="Just a"
s.Value(ConvertEncoding("Test",encodings.UTF16))="test"

MsgBox str(s.Count) // shows 4
```

#### Notes:

When using variants for keys, you may get better results if all keys used have the same data type to improve comparison.

All string comparison is done binary. So "test" and "Test" will be different keys. Also the comparison does not check the string encoding, so "test" can refer to two items if one key has UTF8 and one key has UTF16 encoding.

You can choose whether you want to keep the map ordered using the OrderedMap class or whether you prefer a higher speed with the HashMap class.

### 3.18.1 Methods

#### Clear

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all of the elements.

**Constructor**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The default constructor.

**CountKey(key as variant) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how often a key is used in this map.

**find(key as variant) as VariantToVariantHashMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds the key and returns an iterator.

**Notes:** Returns the same value as the last method if the item was not found.

**first as VariantToVariantHashMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the beginning of the map.

**Example:**

```
// Create a map
dim m as new VariantToVariantHashMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as VariantToVariantHashMapIteratorMBS = m.first
dim e as VariantToVariantHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +str(i.Value)
i.MoveNext
wend
```

**hasKey(key as variant) as boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns True if Key is in the map and False if it is not. Returns a Boolean.

**Key(index as integer) as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

**Keys as variant()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the keys as an array.

**Example:**

```
// Create a map
dim m as new VariantToVariantHashMapMBS

m.Value(1)="Hello"
m.Value(2)="World"

for each v as Variant in m.Keys
MsgBox v
next
```

**Notes:** The order is stable and matches the order returned by the Values method at least until the Dictionary is modified. Use this method with For Each to loop through all the keys.

**last as VariantToVariantHashMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the end of the map.

**Example:**

```
// Create a map
dim m as new VariantToVariantHashMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as VariantToVariantHashMapIteratorMBS = m.first
dim e as VariantToVariantHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +str(i.Value)
i.MoveNext
wend
```

**lookup(key as variant, defaultvalue as variant) as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Looks up the passed value of Key.

**Example:**

```
dim map as new VariantToVariantHashMapMBS

map.value("a")="Hello"
map.value("b")="World"
map.value("c")="!"

MsgBox str(map.lookup("d","?")) // shows "?" as value is missing
MsgBox str(map.lookup("a","?")) // shows "Hello" as value is found
```

**Notes:** If Key is found, it returns the corresponding value. If Key is not found, it returns the passed defaultValue.

**Remove(first as VariantToVariantHashMapIteratorMBS, last as VariantToVariantHashMapIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all elements in a range.

See also:

- 3.18.1 Remove(key as variant) as integer 132
- 3.18.1 Remove(pos as VariantToVariantHashMapIteratorMBS) 132

**Remove(key as variant) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element with the given key.

See also:

- 3.18.1 Remove(first as VariantToVariantHashMapIteratorMBS, last as VariantToVariantHashMapIteratorMBS) 132
- 3.18.1 Remove(pos as VariantToVariantHashMapIteratorMBS) 132

**Remove(pos as VariantToVariantHashMapIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element pointed to by the pos iterator.

See also:

- 3.18.1 Remove(first as VariantToVariantHashMapIteratorMBS, last as VariantToVariantHashMapIteratorMBS) 132
- 3.18.1 Remove(key as variant) as integer 132

**value(key as variant) as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value associated with the given key.

**Notes:**

If you query for a key which does not exist, a `KeyNotFoundException` is raised.  
(Read and Write computed property)

### **ValueAtIndex(index as integer) as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value with the given index.

**Notes:** If there is no Indexth item in the map, a call generates an `OutOfBoundsException` error. The first item has the index zero.

### **Values as variant()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the values as an array

#### **Example:**

```
// Create a map
dim m as new VariantToVariantHashMapMBS

m.Value(1)="Hello"
m.Value(2)="World"

for each v as Variant in m.Values
  MsgBox v
next
```

**Notes:** The order is stable and matches the order returned by Keys at least until the Map is modified. Use this method with For Each to loop through all the values.

## **3.18.2 Properties**

### **BinCount as Integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of bins the hash table uses.

**Example:**

```
dim v as new VariantToVariantHashMapMBS
```

```
v.value(1)="Hello"  
v.value(2)="World"
```

```
MsgBox str(v.BinCount)
```

**Notes:**

This is a measure of the hash table size, independent of the number of items the Dictionary contains.  
(Read only property)

**Count as Integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items in this map.

**Example:**

```
dim map as new VariantToVariantHashMapMBS
```

```
map.Value(1)=true  
map.Value("Hello")="World"
```

```
MsgBox str(map.Count)
```

**Notes:** (Read only property)

**Empty as Boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the size is zero.

**Notes:** (Read only property)

**MaxSize as Integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the largest possible size for this map.

**Notes:**

Value is -1 if no limit is defined.  
(Read only property)

## 3.19 class StringHashSetMBS

**class StringHashSetMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An alternative dictionary class for a hash set with strings.

**Example:**

```
dim s as new StringHashSetMBS
```

```
s.insert "test"  
s.insert "Test"
```

```
MsgBox str(s.Count) // shows 2
```

```
s.insert ConvertEncoding("test",encodings.UTF16)  
s.insert ConvertEncoding("Test",encodings.UTF16)
```

```
MsgBox str(s.Count) // shows 4
```

**Notes:**

All string comparison is done binary. So "test" and "Test" will be different keys. Also the comparison does not check the string encoding, so "test" can refer to two items if one key has UTF8 and one key has UTF16 encoding.

You can choose whether you want to keep the set ordered using the OrderedSet class or whether you prefer a higher speed with the HashSet class.

### 3.19.1 Methods

#### Clear

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all of the elements.

#### Constructor

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The default constructor.

#### CountKey(key as string) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how often a key is used in this set.

#### find(key as string) as StringHashSetIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds the key and returns an iterator.

**Notes:** Returns the same value as the last method if the item was not found.

#### first as StringHashSetIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the beginning of the set.

**Example:**

```
// Create a map
dim m as new StringHashSetMBS

m.insert("1")
m.insert("2")
```

```

m.insert("3")

// get iterators pointing to first and after last element
dim i as StringHashSetIteratorMBS = m.first
dim e as StringHashSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox i.Key
  i.MoveNext
wend

```

**insert(key as string)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds a value to the set.

**Key(index as integer) as string**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

**Keys as string()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the keys as an array.

**Example:**

```

dim m as new StringHashSetMBS

m.insert("1")
m.insert("2")

for each v as string in m.Keys
  MsgBox v

```

next

**Notes:** The order is stable and matches the order returned by the Values method at least until the Dictionary is modified. Use this method with For Each to loop through all the keys.

### last as StringHashSetIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the end of the set.

**Example:**

```
// Create a map
dim m as new StringHashSetMBS

m.insert("1")
m.insert("2")
m.insert("3")

// get iterators pointing to first and after last element
dim i as StringHashSetIteratorMBS = m.first
dim e as StringHashSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox i.Key
  i.MoveNext
wend
```

### lookup(key as string) as boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Checks whether an element with the given key exists in this set.

**Example:**

```
dim set as new StringHashSetMBS

set.insert "Hello"
set.insert "World"
```

```
MsgBox str(set.lookup("missed")) // shows false as value is missing
MsgBox str(set.lookup("Hello")) // shows true as value is found
```

**Notes:** Returns true if yes and false if no.

### Remove(first as StringHashSetIteratorMBS, last as StringHashSetIteratorMBS)

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all elements in a range.

See also:

- 3.19.1 Remove(key as string) as integer 139
- 3.19.1 Remove(pos as StringHashSetIteratorMBS) 139

### Remove(key as string) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element with the given key.

See also:

- 3.19.1 Remove(first as StringHashSetIteratorMBS, last as StringHashSetIteratorMBS) 139
- 3.19.1 Remove(pos as StringHashSetIteratorMBS) 139

### Remove(pos as StringHashSetIteratorMBS)

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element pointed to by the pos iterator.

See also:

- 3.19.1 Remove(first as StringHashSetIteratorMBS, last as StringHashSetIteratorMBS) 139
- 3.19.1 Remove(key as string) as integer 139

### 3.19.2 Properties

#### BinCount as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of bins the hash table uses.

**Example:**

```
dim v as new StringHashSetMBS
```

```
v.insert "1"  
v.insert "Hello"
```

```
MsgBox str(v.BinCount)
```

**Notes:**

This is a measure of the hash table size, independent of the number of items the Dictionary contains. (Read only property)

#### Count as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items in this set.

**Example:**

```
dim set as new StringHashSetMBS
```

```
set.insert "a"  
set.insert "b"
```

```
MsgBox str(set.Count)
```

**Notes:** (Read only property)

**Empty as Boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the size is zero.

**Notes:** (Read only property)

**MaxSize as Integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the largest possible size for this set.

**Notes:**

Value is -1 if no limit is defined.

(Read only property)

**3.20 class StringHashSetIteratorMBS****class StringHashSetIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The iterator for the StringHashSet class.

**Example:**

```
// Create a map
dim m as new StringHashSetMBS

m.insert("1")
m.insert("2")
m.insert("3")

// get iterators pointing to first and after last element
dim i as StringHashSetIteratorMBS = m.first
dim e as StringHashSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox i.Key
i.MoveNext
wend
```

### 3.20.1 Methods

**isEqual**(other as `StringHashSetIteratorMBS`) as boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are equal.

**Example:**

```
// Create a map
dim m as new StringHashSetMBS

m.insert("1")
m.insert("2")
m.insert("3")

// get iterators pointing to first and after last element
dim i as StringHashSetIteratorMBS = m.first
dim e as StringHashSetIteratorMBS = m.last

// Show all keys and values
while i.isEqual(e) = false
MsgBox i.Key
i.MoveNext
wend
```

**isNotEqual**(other as `StringHashSetIteratorMBS`) as boolean

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are not equal.

**Example:**

```
// Create a map
dim m as new StringHashSetMBS

m.insert("1")
m.insert("2")
m.insert("3")
```

```
// get iterators pointing to first and after last element
dim i as StringHashSetIteratorMBS = m.first
dim e as StringHashSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox i.Key
  i.MoveNext
wend
```

### Key as string

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

### MoveNext

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the next item.

#### Example:

```
// Create a map
dim m as new StringHashSetMBS

m.insert("1")
m.insert("2")
m.insert("3")

// get iterators pointing to first and after last element
dim i as StringHashSetIteratorMBS = m.first
dim e as StringHashSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox i.Key
  i.MoveNext
wend
```

## 3.21 class IntegerToIntegerOrderedMapMBS

### class IntegerToIntegerOrderedMapMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An alternative dictionary class for an ordered map for integers for keys and values.

#### Example:

```
dim s as new IntegerToIntegerHashMapMBS
```

```
s.Value(1)=3
```

```
s.Value(2)=4
```

```
MsgBox str(s.Count) // shows 2
```

```
MsgBox str(s.Value(1)+s.Value(2)) // shows 7
```

**Notes:** You can choose whether you want to keep the map ordered using the OrderedMap class or whether you prefer a higher speed with the HashMap class.

### 3.21.1 Methods

#### Clear

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all of the elements.

#### Constructor

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The default constructor.

**CountKey(key as integer) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how often a key is used in this map.

**find(key as integer) as IntegerToIntegerOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds the key and returns an iterator.

**Notes:** Returns the same value as the last method if the item was not found.

**first as IntegerToIntegerOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the beginning of the map.

**Example:**

```
// Create a map
dim m as new IntegerToIntegerOrderedMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToIntegerOrderedMapIteratorMBS = m.first
dim e as IntegerToIntegerOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +str(i.Value)
i.MoveNext
wend
```

**hasKey(key as integer) as boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns True if Key is in the map and False if it is not. Returns a Boolean.

**Key(index as integer) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an `OutOfBoundsException` error. The first item has the index zero.

**Keys as integer()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the keys as an array.

**Example:**

```
dim m as new IntegerToIntegerOrderedMapMBS
```

```
m.Value(1)=5
```

```
m.Value(2)=7
```

```
for each v as integer in m.keys
```

```
MsgBox str(v)
```

```
next
```

**Notes:** The order is stable and matches the order returned by the `Values` method at least until the Dictionary is modified. Use this method with `For Each` to loop through all the keys.

**last as IntegerToIntegerOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the end of the map.

**Example:**

```

// Create a map
dim m as new IntegerToIntegerOrderedMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToIntegerOrderedMapIteratorMBS = m.first
dim e as IntegerToIntegerOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +str(i.Value)
i.MoveNext
wend

```

**lookup(key as integer, defaultvalue as integer) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Looks up the passed value of Key.

**Example:**

```

dim map as new IntegerToIntegerOrderedMapMBS

map.value(10)=1
map.value(100)=2
map.value(1000)=3

MsgBox str(map.lookup(5,0)) // shows 0 as value is missing
MsgBox str(map.lookup(10,0)) // shows 1 as value is found

```

**Notes:** If Key is found, it returns the corresponding value. If Key is not found, it returns the passed defaultValue.

**LowerBound(key as integer) as IntegerToIntegerOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator for the first element whose key is not less than k.

**Remove(first as IntegerToIntegerOrderedMapIteratorMBS, last as IntegerToIntegerOrderedMapIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all elements in a range.

See also:

- 3.21.1 Remove(key as integer) as integer 148
- 3.21.1 Remove(pos as IntegerToIntegerOrderedMapIteratorMBS) 148

**Remove(key as integer) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element with the given key.

See also:

- 3.21.1 Remove(first as IntegerToIntegerOrderedMapIteratorMBS, last as IntegerToIntegerOrderedMapIteratorMBS) 148
- 3.21.1 Remove(pos as IntegerToIntegerOrderedMapIteratorMBS) 148

**Remove(pos as IntegerToIntegerOrderedMapIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element pointed to by the pos iterator.

See also:

- 3.21.1 Remove(first as IntegerToIntegerOrderedMapIteratorMBS, last as IntegerToIntegerOrderedMapIteratorMBS) 148
- 3.21.1 Remove(key as integer) as integer 148

**UpperBound(key as integer) as IntegerToIntegerOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator for the first element whose key is greater than k.

**value(key as integer) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value associated with the given key.

**Notes:**

If you query for a key which does not exist, a `KeyNotFoundException` is raised.  
(Read and Write computed property)

**ValueAtIndex(index as integer) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value with the given index.

**Notes:** If there is no `Index`th item in the map, a call generates an `OutOfBoundsException` error. The first item has the index zero.

**Values as integer()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the values as an array

**Example:**

```
dim m as new IntegerToIntegerOrderedMapMBS
```

```
m.Value(1)=5
```

```
m.Value(2)=7
```

```
for each v as integer in m.Values
```

```
MsgBox str(v)
```

```
next
```

**Notes:** The order is stable and matches the order returned by Keys at least until the Map is modified. Use this method with For Each to loop through all the values.

### 3.21.2 Properties

#### Count as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items in this map.

**Example:**

```
dim map as new IntegerToIntegerOrderedMapMBS
```

```
map.Value(1)=3
```

```
map.Value(2)=4
```

```
MsgBox str(map.Count)
```

**Notes:** (Read only property)

#### Empty as Boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the size is zero.

**Notes:** (Read only property)

#### MaxSize as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the largest possible size for this map.

**Notes:**

Value is -1 if no limit is defined.

(Read only property)

## 3.22 class StackObjectMBS

### class StackObjectMBS

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for a stack of objects.

**Example:**

```
dim s as new StackObjectMBS
call s.Push window1
MsgBox window(s.top).title
```

### 3.22.1 Methods

#### Bottom as object

Plugin Version: 3.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the bottom item of the stack and returns the value.

**Example:**

```
dim s as new StackObjectMBS
call s.Push window1
MsgBox window(s.Bottom).title
```

#### clear

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Clears the stack.

**Example:**

```
dim s as new StackObjectMBS

call s.Push window1

s.Clear

if s.IsEmpty then
```

```
MsgBox "OK"  
end if
```

### close

Plugin Version: 3.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor.  
**Notes:**

There is no need to call this method except you want to free all resources of this object now without waiting for Realbasic to do it for you.  
(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

### Contains(o as object) as boolean

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if one of the object references on the stack is equal to the given object reference.

**Example:**

```
dim s as new StackObjectMBS  
  
call s.Push window1  
  
if s.Contains(window1) then  
MsgBox "found. OK"  
else  
MsgBox "not found. Failed"  
end if
```

### Deep as integer

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how much items are on the stack.

**Example:**

```
dim s as new StackObjectMBS
```

```
MsgBox str(s.Deep)
```

```
call s.Push window1
```

```
MsgBox str(s.Deep)
```

### Pop as object

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes the top item of the stack and returns the value.

#### Example:

```
dim s as new StackObjectMBS
call s.Push window1
MsgBox window(s.pop).title
```

**Notes:** Returns nil on any error.

### PopBottom as object

Plugin Version: 3.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes the bottom item of the stack and returns the value.

#### Example:

```
dim s as new StackObjectMBS
call s.Push window1
MsgBox window(s.PopBottom).title
```

### Push(o as object) as boolean

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Pushes a value on the stack.

#### Example:

```
dim s as new StackObjectMBS
call s.Push window1
```

**Notes:**

Returns true if successful.  
May fail on low memory.  
Does not push nil.

**Top as object**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of the top item on the stack.

**Example:**

```
dim s as new StackObjectMBS
call s.Push window1
MsgBox window(s.top).title
```

**Notes:** Returns nil on any error.

### 3.22.2 Properties

**IsEmpty as Boolean**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the stack is empty.

**Example:**

```
dim s as new StackObjectMBS

if s.IsEmpty then
MsgBox "IsEmpty ok"
else
MsgBox "IsEmpty failed"
end if
```

```

call s.Push window1

if s.IsEmpty then
MsgBox "IsEmpty failed"
else
MsgBox "IsEmpty ok"
end if

```

**Notes:** (Read only property)

## 3.23 class ComplexSingleMBS

**class** ComplexSingleMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for complex numbers.

**Example:**

```

dim c as new ComplexSingleMBS(1,2)
dim d as new ComplexSingleMBS(4,7)

dim sum as ComplexSingleMBS = c+d

MsgBox sum.str

```

### 3.23.1 Methods

**abs** as single

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The absolute value of the complex number.

**Example:**

```

dim c as new ComplexSingleMBS(1,2)
MsgBox str(c.abs)

```

**Add(c as ComplexSingleMBS)**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds the given value to this complex number.

See also:

- 3.23.1 Add(x as single) 156

**Add(x as single)**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds the given value to this complex number.

See also:

- 3.23.1 Add(c as ComplexSingleMBS) 156

**arg as single**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return phase angle of complex.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
MsgBox str(c.arg)
```

**conj as ComplexSingleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns complex conjugate.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
dim e as ComplexSingleMBS = c.conj
```

MsgBox e.str

### Constructor(other as ComplexSingleMBS)

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new complex number with the values from the given one.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
dim d as new ComplexSingleMBS(c)
```

```
d.Add 1 // modify second object
MsgBox "d: "+d.str+", c: "+c.str
```

See also:

- 3.23.1 Constructor(x as single = 0.0, y as single = 0.0)

157

### Constructor(x as single = 0.0, y as single = 0.0)

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new complex number with the given values.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
MsgBox str(c.Real)+" "+str(c.Imag)
```

See also:

- 3.23.1 Constructor(other as ComplexSingleMBS)

157

**cos as ComplexSingleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return cosine of complex.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
MsgBox c.cos.str
```

**cosh as ComplexSingleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return hyperbolic cosine of complex.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
MsgBox c.cosh.str
```

**Divide(c as ComplexSingleMBS)**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Divides this complex number by the given complex number.

See also:

- 3.23.1 Divide(x as single) 158

**Divide(x as single)**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Divides this complex number by the given value.

See also:

- 3.23.1 Divide(c as ComplexSingleMBS) 158

**exp as ComplexSingleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return exponential of complex.

**Notes:**

```
dim c as new ComplexSingleMBS(1,2)
MsgBox c.exp.str
```

**log as ComplexSingleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return natural logarithm of complex.

**Example:**

```
dim c as new ComplexSingleMBS(10,10)
MsgBox c.log.str
```

**log10 as ComplexSingleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return common logarithm of complex.

**Example:**

```
dim c as new ComplexSingleMBS(10,0)
MsgBox c.log10.str
```

**Multiply(c as ComplexSingleMBS)**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Multitplies this complex number with the given value.

See also:

- 3.23.1 Multiply(x as single)

**Multiply(x as single)**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Multiplies this complex number with the given value.

See also:

- 3.23.1 Multiply(c as ComplexSingleMBS) 159

**norm as single**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return norm of complex number.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
MsgBox str(c.norm)
```

**Operator\_ Add(c as ComplexSingleMBS) as ComplexSingleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This method is called by Real Studio in order to add two complex numbers.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
dim d as new ComplexSingleMBS(3,4)
dim e as ComplexSingleMBS = c+d
MsgBox e.str
```

See also:

- 3.23.1 Operator\_ Add(x as single) as ComplexSingleMBS 161

**Operator\_ Add(x as single) as ComplexSingleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This method is called by Real Studio in order to add a value to this complex number.

**Example:**

```
dim c as new ComplexSingleMBS(3,4)
dim e as ComplexSingleMBS = c+2
MsgBox e.str
```

See also:

- 3.23.1 Operator\_ Add(c as ComplexSingleMBS) as ComplexSingleMBS

160

**Operator\_ Compare(c as ComplexSingleMBS) as integer**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Compares two complex numbers.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
dim d as new ComplexSingleMBS(1,2)

if c = d then
  MsgBox "equal"
else
  MsgBox "not equal"
end if
```

**Operator\_ Divide(c as ComplexSingleMBS) as ComplexSingleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This method is called by Real Studio in order to divide one complex number by another.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
dim d as new ComplexSingleMBS(3,4)
```

```
dim e as ComplexSingleMBS = c/d
MsgBox e.str
```

See also:

- 3.23.1 Operator\_ Divide(x as single) as ComplexSingleMBS 162

### Operator\_ Divide(x as single) as ComplexSingleMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This method is called by Real Studio in order to divide a complex number by the given value.

#### Example:

```
dim c as new ComplexSingleMBS(3,4)
dim e as ComplexSingleMBS = c/2
MsgBox e.str
```

See also:

- 3.23.1 Operator\_ Divide(c as ComplexSingleMBS) as ComplexSingleMBS 161

### Operator\_ Multiply(c as ComplexSingleMBS) as ComplexSingleMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This method is called by Real Studio in order to multiply two complex numbers.

#### Example:

```
dim c as new ComplexSingleMBS(1,2)
dim d as new ComplexSingleMBS(3,4)
dim e as ComplexSingleMBS = c*d
MsgBox e.str
```

See also:

- 3.23.1 Operator\_ Multiply(x as single) as ComplexSingleMBS 163

**Operator\_ Multiply(x as single) as ComplexSingleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This method is called by Real Studio in order to multiply a single to a complex number.

**Example:**

```
dim c as new ComplexSingleMBS(3,4)
dim e as ComplexSingleMBS = c*2
MsgBox e.str
```

See also:

- 3.23.1 Operator\_ Multiply(c as ComplexSingleMBS) as ComplexSingleMBS

162

**Operator\_ Power(x as ComplexSingleMBS) as ComplexSingleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This method is called by Real Studio in order to calculate the power of two complex numbers.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
dim d as new ComplexSingleMBS(3,4)
dim e as ComplexSingleMBS = c^d
MsgBox e.str
```

**Operator\_ Subtract(c as ComplexSingleMBS) as ComplexSingleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This method is called by Real Studio in order to subtract one complex number from another.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
dim d as new ComplexSingleMBS(3,4)
dim e as ComplexSingleMBS = c-d
MsgBox e.str
```

See also:

- 3.23.1 Operator\_ Subtract(x as single) as ComplexSingleMBS 164

### Operator\_ Subtract(x as single) as ComplexSingleMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This method is called by Real Studio in order to subtract the given value from this complex number.

**Example:**

```
dim c as new ComplexSingleMBS(3,4)
dim e as ComplexSingleMBS = c-2
MsgBox e.str
```

See also:

- 3.23.1 Operator\_ Subtract(c as ComplexSingleMBS) as ComplexSingleMBS 163

### PI as double

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The PI constant.

**Example:**

```
MsgBox str(ComplexSingleMBS.PI)
```

### polar(rho as single, theta as single) as ComplexSingleMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new complex number with the given polar coordinate.

**Example:**

```
MsgBox ComplexSingleMBS.polar(10, 0.5).str
```

**pow(x as ComplexSingleMBS) as ComplexSingleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return complex power.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
dim d as new ComplexSingleMBS(2,0)
dim m as ComplexSingleMBS = c.pow(d)
```

```
MsgBox "c: "+c.str+EndOfLine+"d: "+d.str+EndOfLine+"c^d: "+m.str
```

See also:

- 3.23.1 pow(x as single) as ComplexSingleMBS 165
- 3.23.1 pow(x as single, y as ComplexSingleMBS) as ComplexSingleMBS 166

**pow(x as single) as ComplexSingleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return complex power.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
dim m as ComplexSingleMBS = c.pow(2)
```

```
MsgBox "c: "+c.str+EndOfLine+"c^2: "+m.str
```

See also:

- 3.23.1 pow(x as ComplexSingleMBS) as ComplexSingleMBS 165
- 3.23.1 pow(x as single, y as ComplexSingleMBS) as ComplexSingleMBS 166

**pow(x as single, y as ComplexSingleMBS) as ComplexSingleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Calculates the power of the given values.

**Example:**

```
dim x as new ComplexSingleMBS(2,3)
dim c as ComplexSingleMBS = ComplexSingleMBS.pow(2, x)
```

```
MsgBox c.str
```

See also:

- 3.23.1 pow(x as ComplexSingleMBS) as ComplexSingleMBS 165
- 3.23.1 pow(x as single) as ComplexSingleMBS 165

**sin as ComplexSingleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return sine of complex.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
MsgBox c.sin.str
```

**sinh as ComplexSingleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return hyperbolic sine of complex.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
MsgBox c.sinh.str
```

**sqrt as ComplexSingleMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return square root of complex.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
dim r as ComplexSingleMBS = c.sqrt
dim m as ComplexSingleMBS = r*r
```

```
MsgBox "number: "+c.str+EndOfLine+"root: "+r.str+EndOfLine+"back: "+m.str
```

**str as string**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Shows the number in an human readable format.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
MsgBox c.str
```

**Notes:** The actual format can change.

**Subtract(c as ComplexSingleMBS)**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Subtracts the given complex number from this complex number.

See also:

- 3.23.1 Subtract(x as single)

**Subtract(x as single)**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Subtracts the given value from this complex number.

See also:

- 3.23.1 Subtract(c as ComplexSingleMBS)

167

### tan as ComplexSingleMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return tangent of complex.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
MsgBox c.tan.str
```

### tanh as ComplexSingleMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Return hyperbolic tangent of complex.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
MsgBox c.tanh.str
```

## 3.23.2 Properties

### Imag as single

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Get or set the imaginary part of the complex number.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
MsgBox str(c.Imag)
```

**Notes:** (Read and Write property)

### Real as single

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Get or set the real part of the complex number.

**Example:**

```
dim c as new ComplexSingleMBS(1,2)
MsgBox str(c.Real)
```

**Notes:** (Read and Write property)

## 3.24 class StringToStringOrderedMapIteratorMBS

### class StringToStringOrderedMapIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The iterator for the StringToStringOrderedMap class.

**Example:**

```
// Create a map
dim m as new StringToStringOrderedMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToStringOrderedMapIteratorMBS = m.first
dim e as StringToStringOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

### 3.24.1 Methods

**isEqual**(other as `StringToStringOrderedMapIteratorMBS`) as boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are equal.

**Example:**

```
// Create a map
dim m as new StringToStringOrderedMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToStringOrderedMapIteratorMBS = m.first
dim e as StringToStringOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isEqual(e) = false
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

**isNotEqual**(other as `StringToStringOrderedMapIteratorMBS`) as boolean

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are not equal.

**Example:**

```
// Create a map
dim m as new StringToStringOrderedMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"
```

```
// get iterators pointing to first and after last element
dim i as StringToStringOrderedMapIteratorMBS = m.first
dim e as StringToStringOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e) = false
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

### Key as string

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

### MoveNext

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the next item.

#### Example:

```
// Create a map
dim m as new StringToStringOrderedMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToStringOrderedMapIteratorMBS = m.first
dim e as StringToStringOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

**MovePrev**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the previous item.

**Value as string**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value of the current item in the iterator.

**Notes:** (Read and Write computed property)

**3.25 class IntegerToVariantOrderedMapIteratorMBS****class IntegerToVariantOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The iterator for the IntegerToVariantOrderedMap class.

**Example:**

```
// Create a map
dim m as new IntegerToVariantOrderedMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToVariantOrderedMapIteratorMBS = m.first
dim e as IntegerToVariantOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

### 3.25.1 Methods

**isEqual**(other as IntegerToVariantOrderedMapIteratorMBS) as boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are equal.

**Example:**

```
// Create a map
dim m as new IntegerToVariantOrderedMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToVariantOrderedMapIteratorMBS = m.first
dim e as IntegerToVariantOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isEqual(e) = false
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

**isNotEqual**(other as IntegerToVariantOrderedMapIteratorMBS) as boolean

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are not equal.

**Example:**

```
// Create a map
dim m as new IntegerToVariantOrderedMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8
```

```
// get iterators pointing to first and after last element
dim i as IntegerToVariantOrderedMapIteratorMBS = m.first
dim e as IntegerToVariantOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

### Key as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

### MoveNext

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the next item.

#### Example:

```
// Create a map
dim m as new IntegerToVariantOrderedMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToVariantOrderedMapIteratorMBS = m.first
dim e as IntegerToVariantOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

**MovePrev**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the previous item.

**Value as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value of the current item in the iterator.

**Notes:** (Read and Write computed property)

**3.26 class VariantOrderedSetIteratorMBS****class VariantOrderedSetIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The iterator for the VariantOrderedSet class.

**Example:**

```
// Create a map
dim m as new VariantOrderedSetMBS

m.insert(1.0) // double
m.insert("Hello") // string
m.insert(3) // integer

// get iterators pointing to first and after last element
dim i as VariantOrderedSetIteratorMBS = m.first
dim e as VariantOrderedSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox i.Key
  i.MoveNext
wend
```

### 3.26.1 Methods

**isEqual**(other as VariantOrderedSetIteratorMBS) as boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are equal.

**Example:**

```
// Create a map
dim m as new VariantOrderedSetMBS

m.insert(1.0) // double
m.insert("Hello") // string
m.insert(3) // integer

// get iterators pointing to first and after last element
dim i as VariantOrderedSetIteratorMBS = m.first
dim e as VariantOrderedSetIteratorMBS = m.last

// Show all keys and values
while i.isEqual(e) = false
  MsgBox i.Key
  i.MoveNext
wend
```

**isNotEqual**(other as VariantOrderedSetIteratorMBS) as boolean

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are not equal.

**Example:**

```
// Create a map
dim m as new VariantOrderedSetMBS

m.insert(1.0) // double
m.insert("Hello") // string
m.insert(3) // integer
```

```

// get iterators pointing to first and after last element
dim i as VariantOrderedSetIteratorMBS = m.first
dim e as VariantOrderedSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox i.Key
  i.MoveNext
wend

```

### Key as variant

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

### MoveNext

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the next item.

#### Example:

```

// Create a map
dim m as new VariantOrderedSetMBS

m.insert(1.0) // double
m.insert("Hello") // string
m.insert(3) // integer

// get iterators pointing to first and after last element
dim i as VariantOrderedSetIteratorMBS = m.first
dim e as VariantOrderedSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox i.Key
  i.MoveNext
wend

```

## MovePrev

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the previous item.

## 3.27 class StackVariantMBS

### class StackVariantMBS

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for a stack of variants.

#### Example:

```
dim s as new StackVariantMBS
```

```
call s.Push 5
```

```
MsgBox s.PopBottom
```

### 3.27.1 Methods

#### Bottom as variant

Plugin Version: 3.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the bottom item of the stack and returns the value.

#### Example:

```
dim s as new StackVariantMBS
```

```
call s.Push 5
```

```
MsgBox s.Bottom
```

**clear**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Clears the stack.

**Example:**

```
dim s as new StackVariantMBS
call s.Push 5
s.Clear
if s.IsEmpty then
  MsgBox "OK"
end if
```

**close**

Plugin Version: 3.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor.

**Notes:**

There is no need to call this method except you want to free all resources of this object now without waiting for Realbasic to do it for you.

(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

**Contains(o as variant) as boolean**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if one of the variant references on the stack is equal to the given variant reference.

**Example:**

```
dim s as new StackObjectMBS
call s.Push window1
if s.Contains(window1) then
```

```
MsgBox "found. OK"  
else  
MsgBox "not found. Failed"  
end if
```

### Deep as integer

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how much items are on the stack.

**Example:**

```
dim s as new StackVariantMBS  
  
MsgBox str(s.Deep)  
  
call s.Push window1  
  
MsgBox str(s.Deep)
```

### Pop as variant

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes the top item of the stack and returns the value.

**Example:**

```
dim s as new StackVariantMBS  
  
call s.Push 5  
  
MsgBox s.pop
```

**Notes:** Returns nil on any error.

**PopBottom as variant**

Plugin Version: 3.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes the bottom item of the stack and returns the value.

**Example:**

```
dim s as new StackVariantMBS
```

```
call s.Push 5
```

```
MsgBox s.PopBottom
```

**Push(o as variant) as boolean**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Pushes a value on the stack.

**Example:**

```
dim s as new StackVariantMBS
```

```
call s.Push 5
```

**Notes:**

Returns true if successful.  
May fail on low memory.

**Top as variant**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of the top item on the stack.

**Example:**

```
dim s as new StackVariantMBS
```

```
call s.Push 5
```

```
MsgBox s.top
```

**Notes:** Returns nil on any error.

### 3.27.2 Properties

#### IsEmpty as Boolean

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the stack is empty.

**Example:**

```
dim s as new StackVariantMBS
```

```
if s.IsEmpty then  
  MsgBox "IsEmpty ok"  
else  
  MsgBox "IsEmpty failed"  
end if
```

```
call s.Push 5
```

```
if s.IsEmpty then  
  MsgBox "IsEmpty failed"  
else  
  MsgBox "IsEmpty ok"  
end if
```

**Notes:** (Read only property)

## 3.28 class IntegerToIntegerOrderedMapIteratorMBS

```
class IntegerToIntegerOrderedMapIteratorMBS
```

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The iterator for the IntegerToIntegerOrderedMap class.

**Example:**

```
// Create a map
dim m as new IntegerToIntegerOrderedMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToIntegerOrderedMapIteratorMBS = m.first
dim e as IntegerToIntegerOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +str(i.Value)
i.MoveNext
wend
```

**3.28.1 Methods**

**isEqual(other as IntegerToIntegerOrderedMapIteratorMBS) as boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are equal.

**Example:**

```
// Create a map
dim m as new IntegerToIntegerOrderedMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToIntegerOrderedMapIteratorMBS = m.first
dim e as IntegerToIntegerOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isEqual(e) = false
MsgBox str(i.Key)+" ->" +str(i.Value)
i.MoveNext
```

wend

**isNotEqual**(other as IntegerToIntegerOrderedMapIteratorMBS) as boolean

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are not equal.

**Example:**

```
// Create a map
dim m as new IntegerToIntegerOrderedMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToIntegerOrderedMapIteratorMBS = m.first
dim e as IntegerToIntegerOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +str(i.Value)
  i.MoveNext
wend
```

**Key as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

**MoveNext**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the next item.

**Example:**

```

// Create a map
dim m as new IntegerToIntegerOrderedMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToIntegerOrderedMapIteratorMBS = m.first
dim e as IntegerToIntegerOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +str(i.Value)
  i.MoveNext
wend

```

### MovePrev

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the previous item.

### Value as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value of the current item in the iterator.

**Notes:** (Read and Write computed property)

## 3.29 class StringToStringOrderedMapMBS

```
class StringToStringOrderedMapMBS
```

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An alternative dictionary class for an ordered map with strings for keys and values.

**Example:**

```
dim s as new StringToStringOrderedMapMBS

s.Value("Test")="Hello"
s.Value("test")="World"

MsgBox str(s.Count) // shows 2

s.Value(ConvertEncoding("test",encodings.UTF16))="Just a"
s.Value(ConvertEncoding("Test",encodings.UTF16))="test"

MsgBox str(s.Count) // shows 4
```

### Notes:

All string comparison is done binary. So "test" and "Test" will be different keys. Also the comparison does not check the string encoding, so "test" can refer to two items if one key has UTF8 and one key has UTF16 encoding.

You can choose whether you want to keep the map ordered using the OrderedMap class or whether you prefer a higher speed with the HashMap class.

## 3.29.1 Methods

### Clear

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all of the elements.

### Constructor

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The default constructor.

**CountKey(key as string) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how often a key is used in this map.

**find(key as string) as StringToStringOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds the key and returns an iterator.

**Notes:** Returns the same value as the last method if the item was not found.

**first as StringToStringOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the beginning of the map.

**Example:**

```
// Create a map
dim m as new StringToStringOrderedMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToStringOrderedMapIteratorMBS = m.first
dim e as StringToStringOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +i.Value
  i.MoveNext
wend
```

**hasKey(key as string) as boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns True if Key is in the map and False if it is not. Returns a Boolean.

**Key(index as integer) as string**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an `OutOfBoundsException` error. The first item has the index zero.

**Keys as string()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the keys as an array.

**Example:**

```
dim m as new StringToStringOrderedMapMBS
```

```
m.Value("1")="Hello"
m.Value("2")="World"
```

```
for each v as string in m.Keys
  MsgBox str(v)
next
```

**Notes:** The order is stable and matches the order returned by the `Values` method at least until the Dictionary is modified. Use this method with `For Each` to loop through all the keys.

**last as StringToStringOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the end of the map.

**Example:**

```

// Create a map
dim m as new StringToStringOrderedMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToStringOrderedMapIteratorMBS = m.first
dim e as StringToStringOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend

```

**lookup(key as string, defaultvalue as string) as string**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Looks up the passed value of Key.

**Example:**

```

dim map as new StringToStringOrderedMapMBS

map.value("a")="Hello"
map.value("b")="World"
map.value("c")="!"

MsgBox str(map.lookup("d","?")) // shows "?" as value is missing
MsgBox str(map.lookup("a","?")) // shows "Hello" as value is found

```

**Notes:** If Key is found, it returns the corresponding value. If Key is not found, it returns the passed defaultValue.

**LowerBound(key as string) as StringToStringOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator for the first element whose key is not less than k.

**Remove(first as StringToStringOrderedMapIteratorMBS, last as StringToStringOrderedMapIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all elements in a range.

See also:

- 3.29.1 Remove(key as string) as integer 190
- 3.29.1 Remove(pos as StringToStringOrderedMapIteratorMBS) 190

**Remove(key as string) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element with the given key.

See also:

- 3.29.1 Remove(first as StringToStringOrderedMapIteratorMBS, last as StringToStringOrderedMapIteratorMBS) 190
- 3.29.1 Remove(pos as StringToStringOrderedMapIteratorMBS) 190

**Remove(pos as StringToStringOrderedMapIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element pointed to by the pos iterator.

See also:

- 3.29.1 Remove(first as StringToStringOrderedMapIteratorMBS, last as StringToStringOrderedMapIteratorMBS) 190
- 3.29.1 Remove(key as string) as integer 190

**UpperBound(key as string) as StringToStringOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator for the first element whose key is greater than k.

**value(key as string) as string**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value associated with the given key.

**Notes:**

If you query for a key which does not exist, a `KeyNotFoundException` is raised.  
(Read and Write computed property)

**ValueAtIndex(index as integer) as string**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value with the given index.

**Notes:** If there is no `Index`th item in the map, a call generates an `OutOfBoundsException` error. The first item has the index zero.

**Values as string()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the values as an array

**Example:**

```
dim m as new StringToStringOrderedMapMBS
```

```
m.Value("1")="Hello"  
m.Value("2")="World"
```

```
for each v as string in m.Values  
MsgBox str(v)  
next
```

**Notes:** The order is stable and matches the order returned by Keys at least until the Map is modified. Use this method with For Each to loop through all the values.

### 3.29.2 Properties

#### Count as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items in this map.

**Example:**

```
dim map as new StringToStringOrderedMapMBS
```

```
map.Value("?")="?"
```

```
map.Value("Hello")="World"
```

```
MsgBox str(map.Count)
```

**Notes:** (Read only property)

#### Empty as Boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the size is zero.

**Notes:** (Read only property)

#### MaxSize as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the largest possible size for this map.

**Notes:**

Value is -1 if no limit is defined.

(Read only property)

## 3.30 class VariantToVariantOrderedMapMBS

class **VariantToVariantOrderedMapMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An alternative dictionary class for an ordered map with variants as keys and values.

**Example:**

```
dim s as new VariantToVariantOrderedMapMBS
```

```
s.Value("Test")="Hello"
s.Value("test")="World"
```

```
MsgBox str(s.Count) // shows 2
```

```
s.Value(ConvertEncoding("test",encodings.UTF16))="Just a"
s.Value(ConvertEncoding("Test",encodings.UTF16))="test"
```

```
MsgBox str(s.Count) // shows 4
```

**Notes:**

When using variants for keys, you may get better results if all keys used have the same data type to improve comparison.

All string comparison is done binary. So "test" and "Test" will be different keys. Also the comparison does not check the string encoding, so "test" can refer to two items if one key has UTF8 and one key has UTF16 encoding.

You can choose whether you want to keep the map ordered using the OrderedMap class or whether you prefer a higher speed with the HashMap class.

### 3.30.1 Methods

**Clear**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all of the elements.

**Constructor**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The default constructor.

**CountKey(key as variant) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how often a key is used in this map.

**find(key as variant) as VariantToVariantMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds the key and returns an iterator.

**Notes:** Returns the same value as the last method if the item was not found.

**first as VariantToVariantMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the beginning of the map.

**Example:**

```
// Create a map
dim m as new VariantToVariantOrderedMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as VariantToVariantMapIteratorMBS = m.first
dim e as VariantToVariantMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +str(i.Value)
i.MoveNext
wend
```

**hasKey(key as variant) as boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns True if Key is in the map and False if it is not. Returns a Boolean.

**Key(index as integer) as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

**Keys as variant()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the keys as an array.

**Example:**

```
// Create a map
dim m as new VariantToVariantOrderedMapMBS

m.Value(1)="Hello"
m.Value(2)="World"

for each v as Variant in m.Keys
  MsgBox v
next
```

**Notes:** The order is stable and matches the order returned by the Values method at least until the Dictionary is modified. Use this method with For Each to loop through all the keys.

**last as VariantToVariantMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the end of the map.

**Example:**

```
// Create a map
dim m as new VariantToVariantOrderedMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as VariantToVariantMapIteratorMBS = m.first
dim e as VariantToVariantMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +str(i.Value)
i.MoveNext
wend
```

**lookup(key as variant, defaultvalue as variant) as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Looks up the passed value of Key.

**Example:**

```
dim map as new VariantToVariantOrderedMapMBS

map.value("a")="Hello"
map.value("b")="World"
map.value("c")="!"

MsgBox str(map.lookup("d","?")) // shows "?" as value is missing
MsgBox str(map.lookup("a","?")) // shows "Hello" as value is found
```

**Notes:** If Key is found, it returns the corresponding value. If Key is not found, it returns the passed defaultValue.

**LowerBound(key as variant) as VariantToVariantMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator for the first element whose key is not less than k.

**Remove(first as VariantToVariantMapIteratorMBS, last as VariantToVariantMapIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all elements in a range.

See also:

- 3.30.1 Remove(key as variant) as integer 197
- 3.30.1 Remove(pos as VariantToVariantMapIteratorMBS) 197

**Remove(key as variant) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element with the given key.

See also:

- 3.30.1 Remove(first as VariantToVariantMapIteratorMBS, last as VariantToVariantMapIteratorMBS) 197
- 3.30.1 Remove(pos as VariantToVariantMapIteratorMBS) 197

**Remove(pos as VariantToVariantMapIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element pointed to by the pos iterator.

See also:

- 3.30.1 Remove(first as VariantToVariantMapIteratorMBS, last as VariantToVariantMapIteratorMBS) 197
- 3.30.1 Remove(key as variant) as integer 197

**UpperBound(key as variant) as VariantToVariantMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator for the first element whose key is greater than k.

**value(key as variant) as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value associated with the given key.

**Notes:**

If you query for a key which does not exist, a `KeyNotFoundException` is raised.  
(Read and Write computed property)

**ValueAtIndex(index as integer) as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value with the given index.

**Notes:** If there is no `Index`th item in the map, a call generates an `OutOfBoundsException` error. The first item has the index zero.

**Values as variant()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the values as an array

**Example:**

```
// Create a map
dim m as new VariantToVariantOrderedMapMBS

m.Value(1)="Hello"
m.Value(2)="World"

for each v as Variant in m.Values
MsgBox v
next
```

**Notes:** The order is stable and matches the order returned by Keys at least until the Map is modified. Use this method with For Each to loop through all the values.

### 3.30.2 Properties

#### Count as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items in this map.

**Example:**

```
dim map as new VariantToVariantOrderedMapMBS
```

```
map.Value(1)=true  
map.Value("Hello")="World"
```

```
MsgBox str(map.Count)
```

**Notes:** (Read only property)

#### Empty as Boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the size is zero.

**Notes:** (Read only property)

#### MaxSize as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the largest possible size for this map.

**Notes:**

Value is -1 if no limit is defined.  
(Read only property)

### 3.31 class IntegerHashSetIteratorMBS

class IntegerHashSetIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The iterator for the IntegerHashSet class.

**Example:**

```
// Create a map
dim m as new IntegerHashSetMBS

m.insert(1)
m.insert(2)
m.insert(3)

// get iterators pointing to first and after last element
dim i as IntegerHashSetIteratorMBS = m.first
dim e as IntegerHashSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)
  i.MoveNext
wend
```

#### 3.31.1 Methods

isEqual(other as IntegerHashSetIteratorMBS) as boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are equal.

**Example:**

```
// Create a map
dim m as new IntegerHashSetMBS

m.insert(1)
m.insert(2)
m.insert(3)

// get iterators pointing to first and after last element
```

```

dim i as IntegerHashSetIteratorMBS = m.first
dim e as IntegerHashSetIteratorMBS = m.last

// Show all keys and values
while i.isEqual(e) = false
MsgBox str(i.Key)
i.MoveNext
wend

```

**isNotEqual(other as IntegerHashSetIteratorMBS) as boolean**

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are not equal.

**Example:**

```

// Create a map
dim m as new IntegerHashSetMBS

m.insert(1)
m.insert(2)
m.insert(3)

// get iterators pointing to first and after last element
dim i as IntegerHashSetIteratorMBS = m.first
dim e as IntegerHashSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)
i.MoveNext
wend

```

**Key as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an `OutOfBoundsException` error. The first item has the index zero.

**MoveNext**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the next item.

**Example:**

```
// Create a map
dim m as new IntegerHashSetMBS

m.insert(1)
m.insert(2)
m.insert(3)

// get iterators pointing to first and after last element
dim i as IntegerHashSetIteratorMBS = m.first
dim e as IntegerHashSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)
  i.MoveNext
wend
```

**3.32 class VariantHashSetMBS****class VariantHashSetMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An alternative dictionary class for a hash set with variants.

**Example:**

```
dim s as new VariantHashSetMBS

s.insert "test"
s.insert "Test"

MsgBox str(s.Count) // shows 2

s.insert ConvertEncoding("test",encodings.UTF16)
s.insert ConvertEncoding("Test",encodings.UTF16)
```

```
MsgBox str(s.Count) // shows 4
```

**Notes:**

When using variants for keys, you may get better results if all keys used have the same data type to improve comparison.

All string comparison is done binary. So "test" and "Test" will be different keys. Also the comparison does not check the string encoding, so "test" can refer to two items if one key has UTF8 and one key has UTF16 encoding.

You can choose whether you want to keep the set ordered using the OrderedSet class or whether you prefer a higher speed with the HashSet class.

### 3.32.1 Methods

**Clear**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all of the elements.

**Constructor**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The default constructor.

**CountKey(key as variant) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how often a key is used in this set.

**find(key as variant) as VariantHashSetIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds the key and returns an iterator.

**Notes:** Returns the same value as the last method if the item was not found.

**first as VariantHashSetIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the beginning of the set.

**Example:**

```
// Create a map
dim m as new VariantHashSetMBS

m.insert(1.0) // double
m.insert("Hello") // string
m.insert(3) // integer

// get iterators pointing to first and after last element
dim i as VariantHashSetIteratorMBS = m.first
dim e as VariantHashSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox i.Key
  i.MoveNext
wend
```

**insert(key as variant)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds a value to the set.

**Key(index as integer) as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

**Keys as variant()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the keys as an array.

**Example:**

```
dim m as new VariantHashSetMBS
```

```
m.insert("1")
```

```
m.insert("2")
```

```
for each v as variant in m.Keys
```

```
MsgBox v
```

```
next
```

**Notes:** The order is stable and matches the order returned by the Values method at least until the Dictionary is modified. Use this method with For Each to loop through all the keys.

**last as VariantHashSetIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the end of the set.

**Example:**

```
// Create a map
```

```
dim m as new VariantHashSetMBS
```

```
m.insert(1.0) // double
```

```
m.insert("Hello") // string
```

```
m.insert(3) // integer
```

```
// get iterators pointing to first and after last element
dim i as VariantHashSetIteratorMBS = m.first
dim e as VariantHashSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox i.Key
i.MoveNext
wend
```

### lookup(key as variant) as boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Checks whether an element with the given key exists in this set.

#### Example:

```
dim set as new VariantHashSetMBS

set.insert "Hello"
set.insert "World"

MsgBox str(set.lookup("missed")) // shows false as value is missing
MsgBox str(set.lookup("Hello")) // shows true as value is found
```

**Notes:** Returns true if yes and false if no.

### Remove(first as VariantHashSetIteratorMBS, last as VariantHashSetIteratorMBS)

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all elements in a range.

See also:

- 3.32.1 Remove(key as variant) as integer 207
- 3.32.1 Remove(pos as VariantHashSetIteratorMBS) 207

**Remove(key as variant) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element with the given key.

See also:

- 3.32.1 Remove(first as VariantHashSetIteratorMBS, last as VariantHashSetIteratorMBS) 206
- 3.32.1 Remove(pos as VariantHashSetIteratorMBS) 207

**Remove(pos as VariantHashSetIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element pointed to by the pos iterator.

See also:

- 3.32.1 Remove(first as VariantHashSetIteratorMBS, last as VariantHashSetIteratorMBS) 206
- 3.32.1 Remove(key as variant) as integer 207

**3.32.2 Properties****BinCount as Integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of bins the hash table uses.

**Example:**

```
dim v as new VariantHashSetMBS
```

```
v.insert "1"  
v.insert "Hello"
```

```
MsgBox str(v.BinCount)
```

**Notes:**

This is a measure of the hash table size, independent of the number of items the Dictionary contains.  
(Read only property)

### Count as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items in this set.

**Example:**

```
dim set as new VariantHashSetMBS
```

```
set.insert 1
```

```
set.insert 2
```

```
MsgBox str(set.Count)
```

**Notes:** (Read only property)

### Empty as Boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the size is zero.

**Notes:** (Read only property)

### MaxSize as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the largest possible size for this set.

**Notes:**

Value is -1 if no limit is defined.

(Read only property)

## 3.33 class IntegerToStringOrderedMapMBS

class IntegerToStringOrderedMapMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An alternative dictionary class for an ordered map with integer keys and string values.

**Example:**

```
dim s as new IntegerToStringOrderedMapMBS

s.Value(1)="Hello"
s.Value(2)="World"

MsgBox str(s.Count) // shows 2

MsgBox s.Value(1)+" "+s.Value(2) // shows "Hello World"
```

**Notes:** You can choose whether you want to keep the map ordered using the OrderedMap class or whether you prefer a higher speed with the HashMap class.

### 3.33.1 Methods

**Clear**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all of the elements.

**Constructor**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The default constructor.

**CountKey(key as integer) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how often a key is used in this map.

**find(key as integer) as IntegerToStringOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds the key and returns an iterator.

**Notes:** Returns the same value as the last method if the item was not found.

**first as IntegerToStringOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the beginning of the map.

**Example:**

```
// Create a map
dim m as new IntegerToStringOrderedMapMBS

m.value(1)="Hello"
m.value(2)="World"
m.value(3)="!"

// get iterators pointing to first and after last element
dim i as IntegerToStringOrderedMapIteratorMBS = m.first
dim e as IntegerToStringOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +i.Value
  i.MoveNext
wend
```

**hasKey(key as integer) as boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns True if Key is in the map and False if it is not. Returns a Boolean.

**Key(index as integer) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

**Keys as integer()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the keys as an array.

**Example:**

```
dim m as new IntegerToStringOrderedMapMBS
```

```
m.Value(1)="Hello"
m.Value(2)="World"
```

```
for each v as integer in m.keys
  MsgBox str(v)
next
```

**Notes:** The order is stable and matches the order returned by the Values method at least until the Dictionary is modified. Use this method with For Each to loop through all the keys.

**last as IntegerToStringOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the end of the map.

**Example:**

```

// Create a map
dim m as new IntegerToStringOrderedMapMBS

m.value(1)="Hello"
m.value(2)="World"
m.value(3)="!"

// get iterators pointing to first and after last element
dim i as IntegerToStringOrderedMapIteratorMBS = m.first
dim e as IntegerToStringOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend

```

**lookup(key as integer, defaultvalue as string) as string**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Looks up the passed value of Key.

**Example:**

```

dim map as new IntegerToStringOrderedMapMBS

map.value(10)="Hello"
map.value(100)="World"
map.value(1000)="!"

MsgBox str(map.lookup(5,"?")) // shows "?" as value is missing
MsgBox str(map.lookup(10,"?")) // shows "Hello" as value is found

```

**Notes:** If Key is found, it returns the corresponding value. If Key is not found, it returns the passed defaultValue.

**LowerBound(key as integer) as IntegerToStringOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator for the first element whose key is not less than k.

**Remove(first as IntegerToStringOrderedMapIteratorMBS, last as IntegerToStringOrderedMapIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all elements in a range.

See also:

- 3.33.1 Remove(key as integer) as integer 213
- 3.33.1 Remove(pos as IntegerToStringOrderedMapIteratorMBS) 213

**Remove(key as integer) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element with the given key.

See also:

- 3.33.1 Remove(first as IntegerToStringOrderedMapIteratorMBS, last as IntegerToStringOrderedMapIteratorMBS) 213
- 3.33.1 Remove(pos as IntegerToStringOrderedMapIteratorMBS) 213

**Remove(pos as IntegerToStringOrderedMapIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element pointed to by the pos iterator.

See also:

- 3.33.1 Remove(first as IntegerToStringOrderedMapIteratorMBS, last as IntegerToStringOrderedMapIteratorMBS) 213
- 3.33.1 Remove(key as integer) as integer 213

**UpperBound(key as integer) as IntegerToStringOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator for the first element whose key is greater than k.

**value(key as integer) as string**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value associated with the given key.

**Notes:**

If you query for a key which does not exist, a `KeyNotFoundException` is raised.  
(Read and Write computed property)

**ValueAtIndex(index as integer) as string**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value with the given index.

**Notes:** If there is no `Index`th item in the map, a call generates an `OutOfBoundsException` error. The first item has the index zero.

**Values as string()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the values as an array

**Example:**

```
dim m as new IntegerToStringOrderedMapMBS
```

```
m.Value(1)="Hello"  
m.Value(2)="World"
```

```
for each v as string in m.Values  
MsgBox v  
next
```

**Notes:** The order is stable and matches the order returned by Keys at least until the Map is modified. Use this method with For Each to loop through all the values.

### 3.33.2 Properties

#### Count as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items in this map.

**Example:**

```
dim map as new IntegerToStringOrderedMapMBS
```

```
map.Value(1)="Hello"
```

```
map.Value(2)="World"
```

```
MsgBox str(map.Count)
```

**Notes:** (Read only property)

#### Empty as Boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the size is zero.

**Notes:** (Read only property)

#### MaxSize as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the largest possible size for this map.

**Notes:**

Value is -1 if no limit is defined.

(Read only property)

### 3.34 class StringToVariantOrderedMapIteratorMBS

class StringToVariantOrderedMapIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The iterator for the StringToVariantOrderedMap class.

**Example:**

```
// Create a map
dim m as new StringToVariantOrderedMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToVariantOrderedMapIteratorMBS = m.first
dim e as StringToVariantOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +i.Value
  i.MoveNext
wend
```

#### 3.34.1 Methods

isEqual(other as StringToVariantOrderedMapIteratorMBS) as boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are equal.

**Example:**

```
// Create a map
dim m as new StringToVariantOrderedMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
```

```

dim i as StringToVariantOrderedMapIteratorMBS = m.first
dim e as StringToVariantOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isEqual(e) = false
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend

```

**isNotEqual(other as StringToVariantOrderedMapIteratorMBS) as boolean**

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are not equal.

**Example:**

```

// Create a map
dim m as new StringToVariantOrderedMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToVariantOrderedMapIteratorMBS = m.first
dim e as StringToVariantOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend

```

**Key as string**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an `OutOfBoundsException` error. The first item has the index zero.

### MoveNext

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the next item.

**Example:**

```
// Create a map
dim m as new StringToVariantOrderedMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToVariantOrderedMapIteratorMBS = m.first
dim e as StringToVariantOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +i.Value
  i.MoveNext
wend
```

### MovePrev

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the previous item.

### Value as variant

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value of the current item in the iterator.

**Notes:** (Read and Write computed property)

## 3.35 class StackSingleMBS

class StackSingleMBS

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for a stack of singles.

**Example:**

```
dim s as new StackSingleMBS
```

```
call s.Push 5
```

```
MsgBox str(S.Top)
```

### 3.35.1 Methods

**Bottom as single**

Plugin Version: 3.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the bottom item of the stack and returns the value.

**Example:**

```
dim s as new StackSingleMBS
```

```
call s.Push 5
```

```
MsgBox str(s.Bottom)
```

**clear**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Clears the stack.

**Example:**

```
dim s as new StackSingleMBS
```

```
call s.Push 5
```

```
s.Clear
```

```
if s.IsEmpty then  
  MsgBox "OK"  
end if
```

**close**

Plugin Version: 3.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor.  
**Notes:**

There is no need to call this method except you want to free all resources of this object now without waiting for Realbasic to do it for you.  
(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

**Contains(o as single) as boolean**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if one of the items on the stack is equal to the given single value.

**Example:**

```
dim s as new StackSingleMBS  
  
call s.Push 5  
  
if s.Contains(5) then  
  MsgBox "found. OK"  
else  
  MsgBox "not found. Failed"  
end if
```

**Deep as integer**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how much items are on the stack.

**Example:**

```
dim s as new StackSingleMBS
MsgBox str(s.Deep)
call s.Push 5
MsgBox str(s.Deep)
```

**Pop as single**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes the top item of the stack and returns the value.

**Example:**

```
dim s as new StackSingleMBS
call s.Push 5
MsgBox str(s.Pop)
```

**Notes:** Returns 0 on any error.

**PopBottom as single**

Plugin Version: 3.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes the bottom item of the stack and returns the value.

**Example:**

```
dim s as new StackSingleMBS
call s.Push 5
MsgBox str(S.PopBottom)
```

**Push(o as single) as boolean**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Pushes a value on the stack.

**Example:**

```
dim s as new StackSingleMBS
call s.Push 5
```

**Notes:**

Returns true if successful.  
May fail on low memory.

**Top as single**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of the top item on the stack.

**Example:**

```
dim s as new StackSingleMBS
```

```
call s.Push 5
```

```
MsgBox str(S.Top)
```

**Notes:** Returns 0 on any error.

### 3.35.2 Properties

**IsEmpty as Boolean**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the stack is empty.

**Example:**

```

dim s as new StackSingleMBS

if s.IsEmpty then
  MsgBox "IsEmpty ok"
else
  MsgBox "IsEmpty failed"
end if

call s.Push 5

if s.IsEmpty then
  MsgBox "IsEmpty failed"
else
  MsgBox "IsEmpty ok"
end if

```

**Notes:** (Read only property)

### 3.36 class StringToStringHashMapMBS

```
class StringToStringHashMapMBS
```

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An alternative dictionary class for a hash map with strings as keys and values.

**Example:**

```

dim s as new StringToStringHashMapMBS

s.Value("Test")="Hello"
s.Value("test")="World"

MsgBox str(s.Count) // shows 2

s.Value(ConvertEncoding("test",encodings.UTF16))="Just a"
s.Value(ConvertEncoding("Test",encodings.UTF16))="test"

MsgBox str(s.Count) // shows 4

```

**Notes:**

All string comparison is done binary. So "test" and "Test" will be different keys. Also the comparison does not check the string encoding, so "test" can refer to two items if one key has UTF8 and one key has UTF16 encoding.

You can choose whether you want to keep the map ordered using the `OrderedMap` class or whether you prefer a higher speed with the `HashMap` class.

### 3.36.1 Methods

#### Clear

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all of the elements.

#### Constructor

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The default constructor.

#### CountKey(key as string) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how often a key is used in this map.

#### find(key as string) as StringToStringHashMapIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds the key and returns an iterator.

**Notes:** Returns the same value as the last method if the item was not found.

**first** as **StringToStringHashMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the beginning of the map.

**Example:**

```
// Create a map
dim m as new StringToStringHashMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToStringHashMapIteratorMBS = m.first
dim e as StringToStringHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +i.Value
  i.MoveNext
wend
```

**hasKey(key as string) as boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns True if Key is in the map and False if it is not. Returns a Boolean.

**Key(index as integer) as string**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an `OutOfBoundsException` error. The first item has the index zero.

**Keys as string()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the keys as an array.

**Example:**

```
dim m as new StringToStringHashMapMBS
```

```
m.Value("1")="Hello"
m.Value("2")="World"
```

```
for each v as string in m.Keys
  MsgBox str(v)
next
```

**Notes:** The order is stable and matches the order returned by the Values method at least until the Dictionary is modified. Use this method with For Each to loop through all the keys.

**last as StringToStringHashMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the end of the map.

**Example:**

```
// Create a map
dim m as new StringToStringHashMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToStringHashMapIteratorMBS = m.first
dim e as StringToStringHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +i.Value
  i.MoveNext
wend
```

**lookup(key as string, defaultvalue as string) as string**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Looks up the passed value of Key.

**Example:**

```
dim map as new StringToStringHashMapMBS
```

```
map.value("a")="Hello"
map.value("b")="World"
map.value("c")="!"
```

```
MsgBox str(map.lookup("d","?")) // shows "?" as value is missing
MsgBox str(map.lookup("a","?")) // shows "Hello" as value is found
```

**Notes:** If Key is found, it returns the corresponding value. If Key is not found, it returns the passed defaultValue.

**Remove(first as StringToStringHashMapIteratorMBS, last as StringToStringHashMapIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all elements in a range.

See also:

- 3.36.1 Remove(key as string) as integer 227
- 3.36.1 Remove(pos as StringToStringHashMapIteratorMBS) 228

**Remove(key as string) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element with the given key.

See also:

- 3.36.1 Remove(first as StringToStringHashMapIteratorMBS, last as StringToStringHashMapIteratorMBS) 227
- 3.36.1 Remove(pos as StringToStringHashMapIteratorMBS) 228

### Remove(pos as StringToStringHashMapIteratorMBS)

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element pointed to by the pos iterator.

See also:

- 3.36.1 Remove(first as StringToStringHashMapIteratorMBS, last as StringToStringHashMapIteratorMBS) 227
- 3.36.1 Remove(key as string) as integer 227

### value(key as string) as string

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value associated with the given key.

**Notes:**

If you query for a key which does not exist, a KeyNotFoundException is raised.  
(Read and Write computed property)

### ValueAtIndex(index as integer) as string

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value with the given index.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

### Values as string()

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the values as an array

**Example:**

```

dim m as new StringToStringHashMapMBS

m.Value("1")="Hello"
m.Value("2")="World"

for each v as string in m.Values
  MsgBox str(v)
next

```

**Notes:** The order is stable and matches the order returned by Keys at least until the Map is modified. Use this method with For Each to loop through all the values.

### 3.36.2 Properties

#### BinCount as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of bins the hash table uses.

**Example:**

```

dim v as new StringToStringHashMapMBS

v.value("1")="Hello"
v.value("2")="World"

MsgBox str(v.BinCount)

```

**Notes:**

This is a measure of the hash table size, independent of the number of items the Dictionary contains. (Read only property)

#### Count as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items in this map.

**Example:**

```
dim map as new StringToStringHashMapMBS
map.Value("?")=??"
map.Value("Hello")=?World"
MsgBox str(map.Count)
```

**Notes:** (Read only property)

**Empty as Boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the size is zero.

**Notes:** (Read only property)

**MaxSize as Integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the largest possible size for this map.

**Notes:**

Value is -1 if no limit is defined.  
(Read only property)

### 3.37 class IntegerToIntegerHashMapMBS

**class IntegerToIntegerHashMapMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An alternative dictionary class for a hash map with integers as keys and values.

**Example:**

```
dim s as new IntegerToIntegerHashMapMBS  
  
s.Value(1)=3  
s.Value(2)=4  
  
MsgBox str(s.Count) // shows 2  
  
MsgBox str(s.Value(1)+s.Value(2)) // shows 7
```

**Notes:** You can choose whether you want to keep the map ordered using the `OrderedMap` class or whether you prefer a higher speed with the `HashMap` class.

### 3.37.1 Methods

#### Clear

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all of the elements.

#### Constructor

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The default constructor.

#### CountKey(key as integer) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how often a key is used in this map.

#### find(key as integer) as IntegerToIntegerHashMapIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds the key and returns an iterator.

**Notes:** Returns the same value as the last method if the item was not found.

### first as IntegerToIntegerHashMapIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the beginning of the map.

**Example:**

```
// Create a map
dim m as new IntegerToIntegerHashMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as IntegerToIntegerHashMapIteratorMBS = m.first
dim e as IntegerToIntegerHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +str(i.Value)
  i.MoveNext
wend
```

### hasKey(key as integer) as boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns True if Key is in the map and False if it is not. Returns a Boolean.

### Key(index as integer) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an `OutOfBoundsException` error. The first item has the index zero.

**Keys as integer()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the keys as an array.

**Example:**

```
dim m as new IntegerToIntegerHashMapMBS
```

```
m.Value(1)=5
```

```
m.Value(2)=7
```

```
for each v as integer in m.keys
```

```
MsgBox str(v)
```

```
next
```

**Notes:** The order is stable and matches the order returned by the Values method at least until the Dictionary is modified. Use this method with For Each to loop through all the keys.

**last as IntegerToIntegerHashMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the end of the map.

**Example:**

```
// Create a map
```

```
dim m as new IntegerToIntegerHashMapMBS
```

```
m.value(1)=2
```

```
m.value(2)=4
```

```
m.value(3)=8
```

```
// get iterators pointing to first and after last element
```

```
dim i as IntegerToIntegerHashMapIteratorMBS = m.first
```

```
dim e as IntegerToIntegerHashMapIteratorMBS = m.last
```

```
// Show all keys and values
```

```
while i.isNotEqual(e)
```

```
MsgBox str(i.Key)+" ->" +str(i.Value)
```

```
i.MoveNext
```

```
wend
```

**lookup(key as integer, defaultvalue as integer) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Looks up the passed value of Key.

**Example:**

```
dim map as new IntegerToIntegerHashMapMBS
```

```
map.value(10)=1
map.value(100)=2
map.value(1000)=3
```

```
MsgBox str(map.lookup(5,0)) // shows 0 as value is missing
MsgBox str(map.lookup(10,0)) // shows 1 as value is found
```

**Notes:** If Key is found, it returns the corresponding value. If Key is not found, it returns the passed defaultValue.

**Remove(first as IntegerToIntegerHashMapIteratorMBS, last as IntegerToIntegerHashMapIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all elements in a range.

See also:

- 3.37.1 Remove(key as integer) as integer 234
- 3.37.1 Remove(pos as IntegerToIntegerHashMapIteratorMBS) 235

**Remove(key as integer) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element with the given key.

See also:

- 3.37.1 Remove(first as IntegerToIntegerHashMapIteratorMBS, last as IntegerToIntegerHashMapIteratorMBS) 234
- 3.37.1 Remove(pos as IntegerToIntegerHashMapIteratorMBS) 235

### Remove(pos as IntegerToIntegerHashMapIteratorMBS)

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element pointed to by the pos iterator.

See also:

- 3.37.1 Remove(first as IntegerToIntegerHashMapIteratorMBS, last as IntegerToIntegerHashMapIteratorMBS) 234
- 3.37.1 Remove(key as integer) as integer 234

### value(key as integer) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value associated with the given key.

**Notes:**

If you query for a key which does not exist, a KeyNotFoundException is raised.  
(Read and Write computed property)

### ValueAtIndex(index as integer) as integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value with the given index.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

### Values as integer()

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the values as an array

**Example:**

```
dim m as new IntegerToIntegerHashMapMBS

m.Value(1)=5
m.Value(2)=7

for each v as integer in m.Values
  MsgBox str(v)
next
```

**Notes:** The order is stable and matches the order returned by Keys at least until the Map is modified. Use this method with For Each to loop through all the values.

### 3.37.2 Properties

#### BinCount as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of bins the hash table uses.

**Example:**

```
dim v as new IntegerToIntegerHashMapMBS

v.value(1)=10
v.value(2)=20

MsgBox str(v.BinCount)
```

**Notes:**

This is a measure of the hash table size, independent of the number of items the Dictionary contains. (Read only property)

#### Count as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items in this map.

**Example:**

```

dim map as new IntegerToIntegerHashMapMBS
map.Value(1)=3
map.Value(2)=4

MsgBox str(map.Count)

```

**Notes:** (Read only property)

**Empty as Boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the size is zero.

**Notes:** (Read only property)

**MaxSize as Integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the largest possible size for this map.

**Notes:**

Value is -1 if no limit is defined.  
(Read only property)

### 3.38 class StackDoubleMBS

**class StackDoubleMBS**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for a stack of doubles.

**Example:**

```
dim s as new StackDoubleMBS
call s.Push 5
MsgBox str(s.Top)
```

### 3.38.1 Methods

#### Bottom as double

Plugin Version: 3.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the bottom item of the stack and returns the value.

**Example:**

```
dim s as new StackDoubleMBS
call s.Push 5
MsgBox str(s.Bottom)
```

#### clear

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Clears the stack.

**Example:**

```
dim s as new StackDoubleMBS

call s.Push 5

s.Clear

if s.IsEmpty then
MsgBox "OK"
end if
```

**close**

Plugin Version: 3.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor.

**Notes:**

There is no need to call this method except you want to free all resources of this object now without waiting for Realbasic to do it for you.

(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

**Contains(o as double) as boolean**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if one of the items on the stack is equal to the given double value.

**Example:**

```
dim s as new StackDoubleMBS
```

```
call s.Push 5
```

```
if s.Contains(5) then
  MsgBox "found. OK"
```

```
else
```

```
  MsgBox "not found. Failed"
```

```
end if
```

**Deep as integer**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how much items are on the stack.

**Example:**

```
dim s as new StackDoubleMBS
```

```
MsgBox str(s.Deep)
```

```
call s.Push 5
```

```
MsgBox str(s.Deep)
```

### Pop as double

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes the top item of the stack and returns the value.

**Example:**

```
dim s as new StackDoubleMBS
call s.Push 5
MsgBox str(s.pop)
```

**Notes:** Returns 0 on any error.

### PopBottom as double

Plugin Version: 3.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes the bottom item of the stack and returns the value.

**Example:**

```
dim s as new StackDoubleMBS
call s.Push 5
MsgBox str(s.PopBottom)
```

### Push(o as double) as boolean

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Pushes a value on the stack.

**Example:**

```
dim s as new StackDoubleMBS
call s.Push 5
```

**Notes:**

Returns true if successful.  
May fail on low memory.

**Top as double**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of the top item on the stack.

**Example:**

```
dim s as new StackDoubleMBS
call s.Push 5
MsgBox str(s.Top)
```

**Notes:** Returns 0 on any error.

### 3.38.2 Properties

**IsEmpty as Boolean**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the stack is empty.

**Example:**

```
dim s as new StackDoubleMBS

if s.IsEmpty then
MsgBox "IsEmpty ok"
else
MsgBox "IsEmpty failed"
end if

call s.Push 5

if s.IsEmpty then
MsgBox "IsEmpty failed"
else
```

```
MsgBox "IsEmpty ok"
end if
```

**Notes:** (Read only property)

### 3.39 class StringOrderedSetIteratorMBS

```
class StringOrderedSetIteratorMBS
```

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The iterator for the StringOrderedSet class.

**Example:**

```
// Create a map
dim m as new StringOrderedSetMBS

m.insert("1")
m.insert("2")
m.insert("3")

// get iterators pointing to first and after last element
dim i as StringOrderedSetIteratorMBS = m.first
dim e as StringOrderedSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox i.Key
i.MoveNext
wend
```

#### 3.39.1 Methods

```
isEqual(other as StringOrderedSetIteratorMBS) as boolean
```

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are equal.

**Example:**

```
// Create a map
dim m as new StringOrderedSetMBS

m.insert("1")
m.insert("2")
m.insert("3")

// get iterators pointing to first and after last element
dim i as StringOrderedSetIteratorMBS = m.first
dim e as StringOrderedSetIteratorMBS = m.last

// Show all keys and values
while i.isEqual(e) = false
MsgBox i.Key
i.MoveNext
wend
```

**isNotEqual(other as StringOrderedSetIteratorMBS) as boolean**

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are not equal.

**Example:**

```
// Create a map
dim m as new StringOrderedSetMBS

m.insert("1")
m.insert("2")
m.insert("3")

// get iterators pointing to first and after last element
dim i as StringOrderedSetIteratorMBS = m.first
dim e as StringOrderedSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox i.Key
i.MoveNext
wend
```

### Key as string

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

### MoveNext

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the next item.

#### Example:

```
// Create a map
dim m as new StringOrderedSetMBS

m.insert("1")
m.insert("2")
m.insert("3")

// get iterators pointing to first and after last element
dim i as StringOrderedSetIteratorMBS = m.first
dim e as StringOrderedSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox i.Key
i.MoveNext
wend
```

### MovePrev

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the previous item.

## 3.40 class VariantToVariantHashMapIteratorMBS

**class VariantToVariantHashMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The iterator for the VariantToVariantHashMap class.

**Example:**

```
// Create a map
dim m as new VariantToVariantHashMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as VariantToVariantHashMapIteratorMBS = m.first
dim e as VariantToVariantHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +str(i.Value)
  i.MoveNext
wend
```

### 3.40.1 Methods

**isEqual(other as VariantToVariantHashMapIteratorMBS) as boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are equal.

**Example:**

```
// Create a map
dim m as new VariantToVariantHashMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
```

```

dim i as VariantToVariantHashMapIteratorMBS = m.first
dim e as VariantToVariantHashMapIteratorMBS = m.last

// Show all keys and values
while i.isEqual(e) = false
MsgBox str(i.Key)+" ->" +str(i.Value)
i.MoveNext
wend

```

**isNotEqual**(other as VariantToVariantHashMapIteratorMBS) as boolean

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are not equal.

**Example:**

```

// Create a map
dim m as new VariantToVariantHashMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as VariantToVariantHashMapIteratorMBS = m.first
dim e as VariantToVariantHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +str(i.Value)
i.MoveNext
wend

```

**Key as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an `OutOfBoundsException` error. The first item has the index zero.

**MoveNext**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the next item.

**Example:**

```
// Create a map
dim m as new VariantToVariantHashMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as VariantToVariantHashMapIteratorMBS = m.first
dim e as VariantToVariantHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +str(i.Value)
i.MoveNext
wend
```

**Value as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value of the current item in the iterator.

**Notes:** (Read and Write computed property)

**3.41 class StringToVariantOrderedMapMBS**

**class StringToVariantOrderedMapMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An alternative dictionary class for an ordered map with strings as keys and variants as values.

**Example:**

```
dim s as new StringToVariantOrderedMapMBS

s.Value("Test")="Hello"
s.Value("test")="World"

MsgBox str(s.Count) // shows 2

s.Value(ConvertEncoding("test",encodings.UTF16))="Just a"
s.Value(ConvertEncoding("Test",encodings.UTF16))="test"

MsgBox str(s.Count) // shows 4
```

### Notes:

All string comparison is done binary. So "test" and "Test" will be different keys. Also the comparison does not check the string encoding, so "test" can refer to two items if one key has UTF8 and one key has UTF16 encoding.

You can choose whether you want to keep the map ordered using the OrderedMap class or whether you prefer a higher speed with the HashMap class.

## 3.41.1 Methods

### Clear

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all of the elements.

### Constructor

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The default constructor.

**CountKey(key as string) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how often a key is used in this map.

**find(key as string) as StringToVariantOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds the key and returns an iterator.

**Notes:** Returns the same value as the last method if the item was not found.

**first as StringToVariantOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the beginning of the map.

**Example:**

```
// Create a map
dim m as new StringToVariantOrderedMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToVariantOrderedMapIteratorMBS = m.first
dim e as StringToVariantOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend
```

**HasKey(key as string) as boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns True if Key is in the map and False if it is not. Returns a Boolean.

**Key(index as integer) as string**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an `OutOfBoundsException` error. The first item has the index zero.

**Keys as string()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the keys as an array.

**Example:**

```
dim m as new StringToVariantOrderedMapMBS
```

```
m.Value("1")="Hello"
m.Value("2")="World"
```

```
for each v as string in m.Keys
  MsgBox v
next
```

**Notes:** The order is stable and matches the order returned by the `Values` method at least until the Dictionary is modified. Use this method with `For Each` to loop through all the keys.

**last as StringToVariantOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the end of the map.

**Example:**

```

// Create a map
dim m as new StringToVariantOrderedMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToVariantOrderedMapIteratorMBS = m.first
dim e as StringToVariantOrderedMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend

```

**lookup(key as string, defaultvalue as variant) as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Looks up the passed value of Key.

**Example:**

```

dim map as new StringToVariantOrderedMapMBS

map.value("a")="Hello"
map.value("b")="World"
map.value("c")="!"

MsgBox str(map.lookup("d","?")) // shows "?" as value is missing
MsgBox str(map.lookup("a","?")) // shows "Hello" as value is found

```

**Notes:** If Key is found, it returns the corresponding value. If Key is not found, it returns the passed defaultValue.

**LowerBound(key as string) as StringToVariantOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator for the first element whose key is not less than k.

**Remove(first as StringToVariantOrderedMapIteratorMBS, last as StringToVariantOrderedMapIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all elements in a range.

See also:

- 3.41.1 Remove(key as string) as integer 252
- 3.41.1 Remove(pos as StringToVariantOrderedMapIteratorMBS) 252

**Remove(key as string) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element with the given key.

See also:

- 3.41.1 Remove(first as StringToVariantOrderedMapIteratorMBS, last as StringToVariantOrderedMapIteratorMBS) 252
- 3.41.1 Remove(pos as StringToVariantOrderedMapIteratorMBS) 252

**Remove(pos as StringToVariantOrderedMapIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element pointed to by the pos iterator.

See also:

- 3.41.1 Remove(first as StringToVariantOrderedMapIteratorMBS, last as StringToVariantOrderedMapIteratorMBS) 252
- 3.41.1 Remove(key as string) as integer 252

**UpperBound(key as string) as StringToVariantOrderedMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator for the first element whose key is greater than k.

**value(key as string) as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value associated with the given key.

**Notes:**

If you query for a key which does not exist, a KeyNotFoundException is raised.  
(Read and Write computed property)

**ValueAtIndex(index as integer) as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value with the given index.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

**Values as variant()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the values as an array

**Example:**

```
dim m as new StringToVariantOrderedMapMBS
```

```
m.Value("1")="Hello"  
m.Value("2")="World"
```

```
for each v as string in m.Values  
MsgBox v  
next
```

**Notes:** The order is stable and matches the order returned by Keys at least until the Map is modified. Use this method with For Each to loop through all the values.

### 3.41.2 Properties

#### Count as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items in this map.

**Example:**

```
dim map as new StringToVariantOrderedMapMBS
```

```
map.Value("?")="?"
map.Value("Hello")="World"
```

```
MsgBox str(map.Count)
```

**Notes:** (Read only property)

#### Empty as Boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the size is zero.

**Notes:** (Read only property)

#### MaxSize as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the largest possible size for this map.

**Notes:**

Value is -1 if no limit is defined.  
(Read only property)

## 3.42 class VariantToVariantMapIteratorMBS

class VariantToVariantMapIteratorMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The iterator for the VariantToVariantMap class.

**Example:**

```
// Create a map
dim m as new VariantToVariantOrderedMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as VariantToVariantMapIteratorMBS = m.first
dim e as VariantToVariantMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +str(i.Value)
i.MoveNext
wend
```

### 3.42.1 Methods

isEqual(other as VariantToVariantMapIteratorMBS) as boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are equal.

**Example:**

```
// Create a map
dim m as new VariantToVariantOrderedMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
```

```

dim i as VariantToVariantMapIteratorMBS = m.first
dim e as VariantToVariantMapIteratorMBS = m.last

// Show all keys and values
while i.isEqual(e) = false
MsgBox str(i.Key)+" ->" +str(i.Value)
i.MoveNext
wend

```

**isNotEqual(other as VariantToVariantMapIteratorMBS) as boolean**

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are not equal.

**Example:**

```

// Create a map
dim m as new VariantToVariantOrderedMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as VariantToVariantMapIteratorMBS = m.first
dim e as VariantToVariantMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +str(i.Value)
i.MoveNext
wend

```

**Key as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an `OutOfBoundsException` error. The first item has the index zero.

**MoveNext**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the next item.

**Example:**

```
// Create a map
dim m as new VariantToVariantOrderedMapMBS

m.value(1)=2
m.value(2)=4
m.value(3)=8

// get iterators pointing to first and after last element
dim i as VariantToVariantMapIteratorMBS = m.first
dim e as VariantToVariantMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +str(i.Value)
  i.MoveNext
wend
```

**MovePrev**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the previous item.

**Value as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value of the current item in the iterator.

**Notes:** (Read and Write computed property)

### 3.43 class StringToVariantHashMapIteratorMBS

**class StringToVariantHashMapIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The iterator for the StringToVariantHashMap class.

**Example:**

```
// Create a map
dim m as new StringToVariantHashMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToVariantHashMapIteratorMBS = m.first
dim e as StringToVariantHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +i.Value
  i.MoveNext
wend
```

#### 3.43.1 Methods

**isEqual(other as StringToVariantHashMapIteratorMBS) as boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are equal.

**Example:**

```
// Create a map
dim m as new StringToVariantHashMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
```

```

dim i as StringToVariantHashMapIteratorMBS = m.first
dim e as StringToVariantHashMapIteratorMBS = m.last

// Show all keys and values
while i.isEqual(e) = false
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend

```

**isNotEqual(other as StringToVariantHashMapIteratorMBS) as boolean**

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are not equal.

**Example:**

```

// Create a map
dim m as new StringToVariantHashMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToVariantHashMapIteratorMBS = m.first
dim e as StringToVariantHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend

```

**Key as string**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an `OutOfBoundsException` error. The first item has the index zero.

**MoveNext**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the next item.

**Example:**

```
// Create a map
dim m as new StringToVariantHashMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToVariantHashMapIteratorMBS = m.first
dim e as StringToVariantHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +i.Value
  i.MoveNext
wend
```

**Value as variant**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value of the current item in the iterator.

**Notes:** (Read and Write computed property)

**3.44 class StringToStringHashMapIteratorMBS**

```
class StringToStringHashMapIteratorMBS
```

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The iterator for the StringToStringHashMap class.

**Example:**

```

// Create a map
dim m as new StringToStringHashMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToStringHashMapIteratorMBS = m.first
dim e as StringToStringHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend

```

### 3.44.1 Methods

**isEqual(other as StringToStringHashMapIteratorMBS) as boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are equal.

**Example:**

```

// Create a map
dim m as new StringToStringHashMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToStringHashMapIteratorMBS = m.first
dim e as StringToStringHashMapIteratorMBS = m.last

// Show all keys and values
while i.isEqual(e) = false
MsgBox str(i.Key)+" ->" +i.Value
i.MoveNext
wend

```

**isNotEqual(other as StringToStringHashMapIteratorMBS) as boolean**

Plugin Version: 10.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if both iterators are not equal.

**Example:**

```
// Create a map
dim m as new StringToStringHashMapMBS

m.value("1")="Hello"
m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToStringHashMapIteratorMBS = m.first
dim e as StringToStringHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +i.Value
  i.MoveNext
wend
```

**Key as string**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an OutOfBoundsException error. The first item has the index zero.

**MoveNext**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the iterator to the next item.

**Example:**

```
// Create a map
dim m as new StringToStringHashMapMBS

m.value("1")="Hello"
```

```

m.value("2")="World"
m.value("3")="!"

// get iterators pointing to first and after last element
dim i as StringToStringHashMapIteratorMBS = m.first
dim e as StringToStringHashMapIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox str(i.Key)+" ->" +i.Value
  i.MoveNext
wend

```

### Value as string

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value of the current item in the iterator.

**Notes:** (Read and Write computed property)

## 3.45 class StringOrderedSetMBS

class StringOrderedSetMBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An alternative dictionary class for an ordered set of strings.

**Example:**

```

dim s as new StringOrderedSetMBS

s.insert "test"
s.insert "Test"

MsgBox str(s.Count) // shows 2

s.insert ConvertEncoding("test",encodings.UTF16)
s.insert ConvertEncoding("Test",encodings.UTF16)

MsgBox str(s.Count) // shows 4

```

**Notes:**

All string comparison is done binary. So "test" and "Test" will be different keys. Also the comparison does not check the string encoding, so "test" can refer to two items if one key has UTF8 and one key has UTF16 encoding.

You can choose whether you want to keep the set ordered using the `OrderedSet` class or whether you prefer a higher speed with the `HashSet` class.

### 3.45.1 Methods

**Clear**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all of the elements.

**Constructor**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The default constructor.

**CountKey(key as string) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Counts how often a key is used in this set.

**find(key as string) as StringOrderedSetIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds the key and returns an iterator.

**Notes:** Returns the same value as the last method if the item was not found.

**first** as **StringOrderedSetIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the beginning of the set.

**Example:**

```
// Create a map
dim m as new StringOrderedSetMBS

m.insert("1")
m.insert("2")
m.insert("3")

// get iterators pointing to first and after last element
dim i as StringOrderedSetIteratorMBS = m.first
dim e as StringOrderedSetIteratorMBS = m.last

// Show all keys and values
while i.isNotEqual(e)
  MsgBox i.Key
  i.MoveNext
wend
```

**insert(key as string)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds a value to the set.

**Key(index as integer) as string**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the value of key for the Indexth sequential item.

**Notes:** If there is no Indexth item in the map, a call generates an `OutOfBoundsException` error. The first item has the index zero.

**Keys as string()**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns all the keys as an array.

**Example:**

```
dim m as new StringOrderedSetMBS
```

```
m.insert("1")
```

```
m.insert("2")
```

```
for each v as string in m.Keys
```

```
MsgBox v
```

```
next
```

**Notes:** The order is stable and matches the order returned by the Values method at least until the Dictionary is modified. Use this method with For Each to loop through all the keys.

**last as StringOrderedSetIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator pointing to the end of the set.

**Example:**

```
// Create a map
```

```
dim m as new StringOrderedSetMBS
```

```
m.insert("1")
```

```
m.insert("2")
```

```
m.insert("3")
```

```
// get iterators pointing to first and after last element
```

```
dim i as StringOrderedSetIteratorMBS = m.first
```

```
dim e as StringOrderedSetIteratorMBS = m.last
```

```
// Show all keys and values
```

```
while i.isNotEqual(e)
```

```
MsgBox i.Key
```

```
i.MoveNext
```

```
wend
```

**lookup(key as string) as boolean**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Checks whether an element with the given key exists in this set.

**Example:**

```
dim set as new StringOrderedSetMBS
```

```
set.insert "Hello"
set.insert "World"
```

```
MsgBox str(set.lookup("missed")) // shows false as value is missing
MsgBox str(set.lookup("Hello")) // shows true as value is found
```

**Notes:** Returns true if yes and false if no.

**LowerBound(key as string) as StringOrderedSetIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator for the first element whose key is not less than k.

**Remove(first as StringOrderedSetIteratorMBS, last as StringOrderedSetIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases all elements in a range.

See also:

- 3.45.1 Remove(key as string) as integer 268
- 3.45.1 Remove(pos as StringOrderedSetIteratorMBS) 268

**Remove(key as string) as integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element with the given key.

See also:

- 3.45.1 Remove(first as StringOrderedSetIteratorMBS, last as StringOrderedSetIteratorMBS) 267
- 3.45.1 Remove(pos as StringOrderedSetIteratorMBS) 268

**Remove(pos as StringOrderedSetIteratorMBS)**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Erases the element pointed to by the pos iterator.

See also:

- 3.45.1 Remove(first as StringOrderedSetIteratorMBS, last as StringOrderedSetIteratorMBS) 267
- 3.45.1 Remove(key as string) as integer 268

**UpperBound(key as string) as StringOrderedSetIteratorMBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns an iterator for the first element whose key is greater than k.

**3.45.2 Properties****Count as Integer**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items in this set.

**Example:**

```
dim set as new StringOrderedSetMBS
```

```
set.insert "a"
```

```
set.insert "b"
```

MsgBox str(set.Count)

**Notes:** (Read only property)

### Empty as Boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the size is zero.

**Notes:** (Read only property)

### MaxSize as Integer

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the largest possible size for this set.

**Notes:**

Value is -1 if no limit is defined.  
(Read only property)



# Chapter 4

## Math

### 4.1 class UnsignedInteger64MBS

`class UnsignedInteger64MBS`

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class to make an unsigned 64bit integer available to Realbasic.

**Example:**

```
dim s as new UnsignedInteger64MBS(1000)
MsgBox s.StringValue
```

**Notes:** For Realbasic 5.x the operator methods will be called automatically, but for Realbasic 4.5 you need to call them yourself to perform a given operation.

#### 4.1.1 Methods

**Constructor**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A constructor which sets the value to 0.

**Example:**

```
dim u as new UnsignedInteger64MBS
u = u + 5
MsgBox u.StringValue
```

See also:

- 4.1.1 Constructor(value as double) 272
- 4.1.1 Constructor(value as string) 272

### Constructor(value as double)

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A constructor which takes a double.

See also:

- 4.1.1 Constructor 271
- 4.1.1 Constructor(value as string) 272

### Constructor(value as string)

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A constructor which takes a string.

**Example:**

```
dim u as new UnsignedInteger64MBS("5")
u = u + 5
MsgBox u.StringValue
```

See also:

- 4.1.1 Constructor 271
- 4.1.1 Constructor(value as double) 272

### DoubleValue as double

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value as a double.

**Example:**

```
dim u as new UnsignedInteger64MBS("5")
MsgBox str(u.DoubleValue)
```

**Notes:**

You can lose precision using this function.  
(Read and Write computed property)

### IntegerValue as integer

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value as an integer.

**Example:**

```
dim u as new UnsignedInteger64MBS("5")
MsgBox str(u.IntegerValue)
```

**Notes:**

You can lose precision using this function.  
(Read and Write computed property)

### Is32bit as boolean

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True only if the higher 32bits of the value are all 0.

**Example:**

```
dim s as new UnsignedInteger64MBS(1000)
MsgBox str(s.Is32bit)
```

**Maximum as UnsignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The maximum 64bit unsigned integer value possible.

**Example:**

```
MsgBox UnsignedInteger64MBS.Maximum
```

**Notes:** returns & hffffffffffffff

**MemoryblockValue as Memoryblock**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value stored in an eight byte memoryblock.

**Example:**

```
dim s as new UnsignedInteger64MBS(1000)
dim m as MemoryBlock = s.MemoryblockValue
```

```
MsgBox str(m.Int64Value(0))
```

**Notes:**

You can set it to nil to set it to 0.  
(Read and Write computed property)

**Operator\_ Add(value as UnsignedInteger64MBS) as UnsignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Example:**

```
dim s as new UnsignedInteger64MBS(5)
dim t as new UnsignedInteger64MBS(3)
```

```
s = s + t
```

```
MsgBox s
```

**Notes:**

```
// for
x=y+5
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ AddRight(value as double) as UnsignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Example:**

```
dim s as new UnsignedInteger64MBS(5)
```

```
s = s + 3
```

```
MsgBox s
```

**Notes:**

```
// for
x=5 + y
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ Compare(value as UnsignedInteger64MBS) as integer**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Example:**

```
dim s as new UnsignedInteger64MBS(5)
```

```
if s = 5 then
  MsgBox "is 5"
else
  MsgBox "is not 5"
end if
```

**Notes:**

```
if x=y or x<>y or y<x or y>x then
end if
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ Convert as double**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
d=x
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

See also:

- 4.1.1 Operator\_ Convert as string 277
- 4.1.1 Operator\_ Convert(value as double) 277
- 4.1.1 Operator\_ Convert(value as string) 277

**Operator\_ Convert as string**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

s=x

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.  
See also:

- 4.1.1 Operator\_ Convert as double 276
- 4.1.1 Operator\_ Convert(value as double) 277
- 4.1.1 Operator\_ Convert(value as string) 277

**Operator\_ Convert(value as double)**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

x=5

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.  
See also:

- 4.1.1 Operator\_ Convert as double 276
- 4.1.1 Operator\_ Convert as string 277
- 4.1.1 Operator\_ Convert(value as string) 277

**Operator\_ Convert(value as string)**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
x="1234"
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.  
See also:

- 4.1.1 Operator\_ Convert as double 276
- 4.1.1 Operator\_ Convert as string 277
- 4.1.1 Operator\_ Convert(value as double) 277

#### **Operator\_ Divide(value as UnsignedInteger64MBS) as UnsignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for
x=y / 5
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

#### **Operator\_ DivideRight(value as double) as UnsignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for
x=5 / y
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ IntegerDivide(value as UnsignedInteger64MBS) as UnsignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x=y \5
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ IntegerDivideRight(value as integer) as UnsignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x=5 \y
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ Modulo(value as UnsignedInteger64MBS) as UnsignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x=y mod 5
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ ModuloRight(value as double) as UnsignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x=5 mod y
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ Multiply(value as UnsignedInteger64MBS) as UnsignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x=y * 5
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ MultiplyRight(value as double) as UnsignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x=5*y
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ Negate(value as UnsignedInteger64MBS) as UnsignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x=-y
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ Sub(value as UnsignedInteger64MBS) as UnsignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x= y - 5
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ SubRight(value as double) as UnsignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x=5 - y
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**SignedInteger64 as SignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Casts the value to a signed integer.

**Example:**

```
dim u as new UnsignedInteger64MBS(1000)
dim s as SignedInteger64MBS = u.SignedInteger64
MsgBox s.StringValue
```

**Notes:** (Read and Write computed property)

**StringValue as string**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value as a string.

**Example:**

```
dim u as new UnsignedInteger64MBS("5")
u = u + 5
MsgBox u.StringValue
```

**Notes:** (Read and Write computed property)

**4.1.2 Properties****HigherInteger as Integer**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The higher 32bit integer part of this 64bit value.

**Example:**

```
dim s as new UnsignedInteger64MBS(1000)
MsgBox str(s.HigherInteger)
```

**Notes:** (Read and Write property)

### LowerInteger as Integer

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The lower 32bit integer part of this 64bit value.

**Example:**

```
dim s as new UnsignedInteger64MBS(1000)
MsgBox str(s.LowerInteger)
```

**Notes:** (Read and Write property)

## 4.2 class SignedInteger64MBS

class SignedInteger64MBS

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class to make a signed 64bit integer available to Realbasic.

**Example:**

```
dim s as new SignedInteger64MBS(1000)
MsgBox s.StringValue
```

**Notes:** For Realbasic 5.x the operator methods will be called automatically, but for Realbasic 4.5 you need to call them yourself to perform a given operation.

### 4.2.1 Methods

#### Abs as SignedInteger64MBS

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the absolute value.

**Example:**

```
dim s as new SignedInteger64MBS(-1000)
dim t as SignedInteger64MBS = s.Abs
MsgBox t.StringValue
```

#### Constructor

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A constructor which sets the value to 0.

**Example:**

```
dim u as new SignedInteger64MBS
u = u + 5
MsgBox u.StringValue
```

See also:

- 4.2.1 Constructor(value as double) 284
- 4.2.1 Constructor(value as string) 285

#### Constructor(value as double)

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A constructor which takes a double.

**Example:**

```
dim u as new SignedInteger64MBS(5)
u = u + 5
MsgBox u.StringValue
```

See also:

- 4.2.1 Constructor 284
- 4.2.1 Constructor(value as string) 285

### Constructor(value as string)

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A constructor which takes a string.

#### Example:

```
dim u as new SignedInteger64MBS("5")
u = u + 5
MsgBox u.StringValue
```

See also:

- 4.2.1 Constructor 284
- 4.2.1 Constructor(value as double) 284

### DoubleValue as double

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value as a double.

#### Example:

```
dim u as new SignedInteger64MBS("5")
MsgBox str(u.DoubleValue)
```

#### Notes:

You can loose precision using this function.  
(Read and Write computed property)

**IntegerValue as integer**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value as an integer.

**Example:**

```
dim u as new SignedInteger64MBS("5")
MsgBox str(u.IntegerValue)
```

**Notes:**

You can loose precision using this function.  
(Read and Write computed property)

**Is32bit as boolean**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True only if the higher 32bits of the value are all 0.

**Example:**

```
dim s as new SignedInteger64MBS(1000)
MsgBox str(s.Is32bit)
```

**Maximum as SignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The maximum 64bit signed integer value possible.

**Example:**

```
MsgBox SignedInteger64MBS.Maximum
```

**Notes:** returns: 9223372036854775807

### MemoryblockValue as Memoryblock

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value stored in an eight byte memoryblock.

**Example:**

```
dim s as new SignedInteger64MBS(1000)
dim m as MemoryBlock = s.MemoryblockValue
```

```
MsgBox str(m.Int64Value(0))
```

**Notes:**

You can set it to nil to set it to 0.  
(Read and Write computed property)

### Minimum as SignedInteger64MBS

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The maximum 64bit signed integer value possible.

**Example:**

```
MsgBox SignedInteger64MBS.Minimum
```

**Notes:** returns: -9223372036854775808

### Operator \_ Add(value as SignedInteger64MBS) as SignedInteger64MBS

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Example:**

```
dim s as new SignedInteger64MBS(5)
dim t as new SignedInteger64MBS(3)
```

```
s = s + t
```

```
MsgBox s
```

**Notes:**

```
// for
x=y + 5
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ AddRight(doublevalue as double) as SignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Example:**

```
dim s as new SignedInteger64MBS(5)
```

```
s = s + 3
```

```
MsgBox s
```

**Notes:**

```
// for
x=5 + y
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ Compare(value as SignedInteger64MBS) as integer**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Example:**

```
dim s as new SignedInteger64MBS(5)
```

```
if s = 5 then
  MsgBox "is 5"
else
  MsgBox "is not 5"
end if
```

**Notes:**

```
if x=y or x<>y or y<x or y>x then
end if
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ Convert as double**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
d=x
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

See also:

- 4.2.1 Operator\_ Convert as string 290
- 4.2.1 Operator\_ Convert(value as double) 290
- 4.2.1 Operator\_ Convert(value as string) 290

**Operator\_ Convert as string**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

s=x

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.  
See also:

- 4.2.1 Operator\_ Convert as double 289
- 4.2.1 Operator\_ Convert(value as double) 290
- 4.2.1 Operator\_ Convert(value as string) 290

**Operator\_ Convert(value as double)**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

x=5

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.  
See also:

- 4.2.1 Operator\_ Convert as double 289
- 4.2.1 Operator\_ Convert as string 290
- 4.2.1 Operator\_ Convert(value as string) 290

**Operator\_ Convert(value as string)**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
x="1234"
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.  
See also:

- 4.2.1 Operator\_ Convert as double 289
- 4.2.1 Operator\_ Convert as string 290
- 4.2.1 Operator\_ Convert(value as double) 290

#### **Operator\_ Divide(value as SignedInteger64MBS) as SignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x=y / 5
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

#### **Operator\_ DivideRight(doublevalue as double) as SignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x=5 / y
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ IntegerDivide(value as SignedInteger64MBS) as SignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x=y \5
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ IntegerDivideRight(integervalue as integer) as SignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x=5 \y
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ Modulo(value as SignedInteger64MBS) as SignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x=y mod 5
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ ModuloRight(doublevalue as double) as SignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x=5 mod y
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ Multiply(value as SignedInteger64MBS) as SignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x=y * 5
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ MultiplyRight(doublevalue as double) as SignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x=5*y
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ Negate(value as SignedInteger64MBS) as SignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x=-y
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ Sub(value as SignedInteger64MBS) as SignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x=y - 5
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

**Operator\_ SubRight(doublevalue as double) as SignedInteger64MBS**

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the math operators.

**Notes:**

```
// for  
x=5 - y
```

Realbasic 5.x and newer calls this method whenever you do the operation showed in the example.

### StringValue as string

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value as a string.

**Example:**

```
dim u as new SignedInteger64MBS("5")
u = u + 5
MsgBox u.StringValue
```

**Notes:** (Read and Write computed property)

### UnsignedInteger64 as UnsignedInteger64MBS

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Casts the value to an unsigned integer.

**Example:**

```
dim u as new SignedInteger64MBS(1000)
dim s as UnsignedInteger64MBS = u.UnsignedInteger64
MsgBox s.StringValue
```

**Notes:** (Read and Write computed property)

## 4.2.2 Properties

### HigherInteger as Integer

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The higher 32bit integer part of this 64bit value.

**Example:**

```
dim s as new SignedInteger64MBS(1000)
MsgBox str(s.HigherInteger)
```

**Notes:** (Read and Write property)

### LowerInteger as Integer

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The lower 32bit integer part of this 64bit value.

**Example:**

```
dim s as new SignedInteger64MBS(1000)
MsgBox str(s.LowerInteger)
```

**Notes:** (Read and Write property)

### Sign as Integer

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The sign of the value.

**Example:**

```
dim s as new SignedInteger64MBS(1000)
MsgBox str(s.Sign)
dim t as new SignedInteger64MBS(-1000)
MsgBox str(t.Sign)
```

**Notes:**

1 for positive, -1 for negative and 0 for zero.  
(Read only property)

# Chapter 5

## Notifications

### 5.1 class NotificationObserverMBS

class NotificationObserverMBS

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for receiving notifications.

#### 5.1.1 Methods

Constructor(name as string = "", ref as object = nil, tag as variant = nil)

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new notification observer.

**Notes:** If you register with name = "" and ref = nil, you receive all notifications. If you have a name, you get only notifications matching the name (case sensitive compare). If you have a reference object, you receive only objects for that object. And you can use both name and object.

### 5.1.2 Events

**ReceivedNotification**(name as string, ref as variant, tag as variant, notification as NotificationMBS)

Plugin Version: 10.4 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The event called if a notification was received.

**Notes:**

This event is registered automatically, so do not call RegisterReceiver with the NotificationObserverMBS object.

If you register with name = "" and ref = nil, you receive all notifications. If you have a name, you get only notifications matching the name (case sensitive compare). If you have a reference object, you receive only objects for that object. And you can use both name and object.

## 5.2 class NotificationMBS

**class NotificationMBS**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for an application internal notification.

**Example:**

```
dim n as new NotificationMBS("DatabaseChangedNotification")
NotificationMBS.send(n)
```

**Notes:**

The point of notifications is to inform some other part of your application about something. For example a chart updating if the user enters data in textfields.

So you register for notifications in existing classes/windows with the NotificationReceiverMBS interface. Or you create a subclass of the NotificationObserverMBS class to receive notifications

All notifications are delivered on the same thread as the send method. If needed we could have an asynchron notification system. Please email for that.

Other notifications:

NSNotification: Notifications send from the Cocoa frameworks within your application over NSNotificationCenterMBS class or send across all applications with NSDistributedNotificationCenterMBS.

MacNotificationMBS: A notification message to the user which may have a sound, a message box and/or a jumping dock icon.

### 5.2.1 Methods

**Constructor**(name as string = "", ref as variant = nil, tag as variant = nil)

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new notification with the given values.

**Example:**

```
dim n as new NotificationMBS("DatabaseChangedNotification")
NotificationMBS.send(n)
```

**RegisterReceiver**(target as NotificationReceiverMBS, name as string = "", ref as variant = nil)

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Registers a receiver.  
**Notes:**

You have a class/window and you add NotificationReceiverMBS to the interfaces. Real Studio will add a ReceivedNotification method which looks like this:

```
ReceivedNotification(name as string, ref as variant, tag as variant, notification as NotificationMBS)
```

Don't forget to call `UnregisterReceiver` later in the `Close` event or destructor.

If you register with `name = ""` and `ref = nil`, you receive all notifications. If you have a name, you get only notifications matching the name (case sensitive compare). If you have a reference object, you receive only objects for that object. And you can use both name and object.

### **Send(name as string, ref as object = nil, tag as variant = nil)**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sends a notification.

#### **Example:**

```
NotificationMBS.Send("DatabaseChangedNotification")
```

#### **Notes:**

This is a convenience method which creates a new `NSNotificationMBS` object and sends it.

All registered receivers will get the `ReceivedNotification` method called as well as all registered observers will get an `ReceivedNotification` event.

Of course notifications are filtered by name and/or referenced object.

See also:

- 5.2.1 `Send(notification as NotificationMBS)`

300

### **Send(notification as NotificationMBS)**

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sends a notification.

#### **Example:**

```
dim n as new NotificationMBS("DatabaseChangedNotification")
NotificationMBS.send(n)
```

#### **Notes:**

All registered receivers will get the `ReceivedNotification` method called as well as all registered observers will

get an ReceivedNotification event.

Of course notifications are filtered by name and/or referenced object.

See also:

- 5.2.1 Send(name as string, ref as object = nil, tag as variant = nil) 300

### SendDelayed(name as string, ref as object = nil, tag as variant = nil)

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sends a notification.

**Example:**

```
NotificationMBS.SendDelayed("DatabaseChangedNotification")
```

**Notes:** Same as Send method, but the notification will be delivered later on the main thread. The notification is queued and will wait until there is free CPU time.

See also:

- 5.2.1 SendDelayed(notification as NotificationMBS) 301

### SendDelayed(notification as NotificationMBS)

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sends a notification.

**Example:**

```
dim n as new NotificationMBS("DatabaseChangedNotification", window1, "test")
NotificationMBS.SendDelayed(n)
```

**Notes:** Same as Send method, but the notification will be delivered later on the main thread. The notification is queued and will wait until there is free CPU time.

See also:

- 5.2.1 SendDelayed(name as string, ref as object = nil, tag as variant = nil) 301

## SendNotification

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sends this notification.

### Example:

```
dim n as new NotificationMBS("DatabaseChangedNotification")
n.SendNotification
```

### Notes:

All registered receivers will get the ReceivedNotification method called as well as all registered observers will get an ReceivedNotification event.

Of course notifications are filtered by name and/or referenced object.

## SendNotificationDelayed

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sends a notification.

### Example:

```
dim n as new NotificationMBS("DatabaseChangedNotification")
n.SendNotificationDelayed
```

**Notes:** Same as SendNotification method, but the notification will be delivered later on the main thread. The notification is queued and will wait until there is free CPU time.

## UnregisterReceiver(target as NotificationReceiverMBS)

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unregisters an receiver.

## 5.2.2 Properties

### Name as String

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The name for the notification.

**Example:**

```
dim n as new NotificationMBS("DatabaseChangedNotification", window1, "test")
MsgBox n.Name
```

**Notes:** (Read and Write property)

### Ref as Variant

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The reference object.

**Notes:**

Defines which object the notification references. If nil, you target all objects.  
(Read and Write property)

### Tag as Variant

Plugin Version: 10.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The attached value.

**Example:**

```
dim n as new NotificationMBS
n.Tag = "Hello World"
```

**Notes:**

You can use this property as you like.

This value is sent to the receivers. It allows you to pass an additional value without needing to write a

subclass of the NotificationMBS class.  
(Read and Write property)

## Chapter 6

### List of all classes

• ComplexDoubleMBS	101
• ComplexSingleMBS	155
• IntegerHashSetIteratorMBS	200
• IntegerHashSetMBS	81
• IntegerOrderedSetIteratorMBS	122
• IntegerOrderedSetMBS	75
• IntegerToIntegerHashMapIteratorMBS	63
• IntegerToIntegerHashMapMBS	230
• IntegerToIntegerOrderedMapIteratorMBS	182
• IntegerToIntegerOrderedMapMBS	144
• IntegerToStringHashMapIteratorMBS	53
• IntegerToStringHashMapMBS	115
• IntegerToStringOrderedMapIteratorMBS	56
• IntegerToStringOrderedMapMBS	209
• IntegerToVariantHashMapIteratorMBS	92
• IntegerToVariantHashMapMBS	39
• IntegerToVariantOrderedMapIteratorMBS	172
• IntegerToVariantOrderedMapMBS	46

• NotificationMBS	298
• NotificationObserverMBS	297
• SignedInteger64MBS	283
• StackDoubleMBS	237
• StackIntegerMBS	87
• StackObjectMBS	151
• StackSingleMBS	219
• StackStringMBS	59
• StackVariantMBS	178
• StringHandleMBS	23
• StringHashSetIteratorMBS	141
• StringHashSetMBS	135
• StringOrderedSetIteratorMBS	242
• StringOrderedSetMBS	263
• StringToStringHashMapIteratorMBS	260
• StringToStringHashMapMBS	223
• StringToStringOrderedMapIteratorMBS	169
• StringToStringOrderedMapMBS	185
• StringToVariantHashMapIteratorMBS	258
• StringToVariantHashMapMBS	68
• StringToVariantOrderedMapIteratorMBS	216
• StringToVariantOrderedMapMBS	247
• StrongRefMBS	35
• UnsignedInteger64MBS	271
• VariantHashSetIteratorMBS	125
• VariantHashSetMBS	202
• VariantOrderedSetIteratorMBS	175
• VariantOrderedSetMBS	95
• VariantToVariantHashMapIteratorMBS	245

- VariantToVariantHashMapMBS 128
- VariantToVariantMapIteratorMBS 255
- VariantToVariantOrderedMapMBS 193
- WeakRefMBS 33



# Chapter 7

## List of all global methods

- 3.7 FFTDoubleAbsMBS(x() as ComplexDoubleMBS, N as integer = -1) as double() 66
- 3.7 FFTDoubleAbsMBS(x() as double, N as integer = -1) as double() 66
- 3.7 FFTDoubleMBS(x() as ComplexDoubleMBS, N as integer = -1) as ComplexDoubleMBS() 67
- 3.7 FFTDoubleMBS(x() as double, N as integer = -1) as ComplexDoubleMBS() 67
- 3.7 FFTSingleAbsMBS(x() as ComplexSingleMBS, N as integer = -1) as single() 67
- 3.7 FFTSingleAbsMBS(x() as single, N as integer = -1) as single() 67
- 3.7 FFTSingleMBS(x() as ComplexSingleMBS, N as integer = -1) as ComplexSingleMBS() 68
- 3.7 FFTSingleMBS(x() as single, N as integer = -1) as ComplexSingleMBS() 68