

REFINITIV MANAGEMENT CLASSES VERSION 2.2

Reference Manual

© Refinitiv 2008, 2012, 2020. All Rights Reserved.

Republication or redistribution of Refinitiv content, including by framing or similar means, is prohibited without the prior written consent of Refinitiv. 'Refinitiv' and the Refinitiv logo are registered trademarks and trademarks of Refinitiv.

Any software, including but not limited to: the code, screen, structure, sequence, and organization thereof, and its documentation are protected by national copyright laws and international treaty provisions. This manual is subject to U.S. and other national export regulations.

Refinitiv, by publishing this document, does not guarantee that any information contained herein is and will remain accurate or that use of the information will ensure correct and faultless operation of the relevant service or equipment. Refinitiv, its agents, and its employees, shall not be held liable to or through any user for any loss or damage whatsoever resulting from reliance on the information contained herein.

Contents

Chapter 1	Overview	32
	RMC Publisher Classes:.....	32
	RMC Consumer Classes:	32
	Other Utilities Classes:	33
	Other Type Definitions:	34
Chapter 2	Refinitiv Management Classes 2.1.0 Module Index.....	35
	Refinitiv Management Classes 2.1.0 Modules	35
Chapter 3	Refinitiv Management Classes Class Index.....	41
	Refinitiv Management Classes Class List	41
Chapter 4	Refinitiv Management Classes Module Documentation	46
	RTRConfigDb - State.....	46
	Function Documentation	46
	RTRConfigDb - Query	46
	Function Documentation	46
	RTRConfigDb - Access.....	46
	Function Documentation	46
	RTRConfigDb - OBSOLETE.....	47
	RTRConfigVariable - Attributes	47
	Function Documentation	47
	RTRConfigVariable - State	47
	Function Documentation	47
	RTRConfigVariable - Assignment.....	47
	Function Documentation	48
	RTRConfigVariable - Compatibility.....	48
	Function Documentation	48
	RTRCmdLineArg - Attributes.....	48
	Function Documentation	48
	RTRCmdLineArg - State.....	49
	RTRCmdLineArg - Transformation.....	49
	Function Documentation	49
	RTRCmdLineArg - Query	49
	RTRCmdLineArg - Operations	49

Function Documentation	49
RTRDefaultLogger - Attributes	49
Function Documentation	49
RTRDefaultLogger - Utilities	50
Function Documentation	50
RTREventNotifierImp - Access	50
Function Documentation	50
RTREventNotifierImp - Insert	51
Function Documentation	51
RTREventNotifierImp - Remove	51
Function Documentation	51
RTREventNotifierImp - Implementation	52
Function Documentation	52
RTREventNotifier - Query	52
Function Documentation	53
RTREventNotifier - Access	53
Function Documentation	53
RTREventNotifier - Insert	53
Function Documentation	54
RTREventNotifier - Remove	54
Function Documentation	54
RTREventNotifier - Control	55
Function Documentation	55
RTREventNotifier - Control - OBSOLETE	55
Function Documentation	55
RTRLockableObj - Operations	55
RTRLockableObj - State	56
Function Documentation	56
RTRManagedBooleanConfig - Assignment	56
Function Documentation	56
RTRManagedBooleanConfig - Attributes	56
Function Documentation	56
RTRManagedBoolean - Comparison	57
Function Documentation	57
RTRManagedBoolean - Access	57

Function Documentation	57
RTRManagedBoolean - Transformation.....	57
Function Documentation	57
RTRManagedBoolean - State	57
Function Documentation	57
RTRManagedBoolean - Assignment	57
Function Documentation	57
RTRManagedBoolean - Operations	58
Function Documentation	58
RTRManagedCounter - Comparison.....	58
Function Documentation	58
RTRManagedCounter - Access.....	58
Function Documentation	58
RTRManagedCounter - Transformation	58
Function Documentation	58
RTRManagedCounter - Operations.....	59
Function Documentation	59
RTRManagedGaugeConfig - Attributes.....	59
Function Documentation	59
RTRMgmtAction - Identity	59
Function Documentation	59
RTRMgmtAction - Attributes.....	60
Function Documentation	60
RTRMgmtAction - State.....	60
Function Documentation	60
RTRMgmtAction - Operations	60
Function Documentation	60
RTRMgmtEvent - Attributes.....	61
Function Documentation	61
RTRMgmtEvent - Comparison	61
Function Documentation	61
RTRMgmtEvent - Operation	62
Function Documentation	62
RTRMgmtEvent - Operations Assignment	62
RTRMgmtEvent - OBSOLETE	62

Function Documentation	63
RTRManagedGauge - Attributes	63
Function Documentation	63
RTRManagedGauge - Operations.....	64
Function Documentation	64
RTRManagedNumericConfig - Attributes	64
Function Documentation	64
RTRManagedNumericConfig - Assignment	65
Function Documentation	65
RTRManagedNumericRange - Attributes.....	66
Function Documentation	66
RTRManagedNumericRange - Assignment	66
Function Documentation	66
RTRManagedNumericRange - Operations	66
Function Documentation	66
RTRManagedLargeNumeric - Comparison.....	67
Function Documentation	67
RTRManagedLargeNumeric - Access.....	67
Function Documentation	67
RTRManagedLargeNumeric - Transformation	67
Function Documentation	67
RTRManagedNumeric - Comparison	67
Function Documentation	67
RTRManagedNumeric - Access	68
Function Documentation	68
RTRManagedNumeric - Transformation	68
Function Documentation	68
RTRManagedObject - Identity	68
Function Documentation	68
RTRManagedObject - Attributes	68
Function Documentation	69
RTRManagedObject - State	69
Function Documentation	69
RTRManagedObject - Query.....	70
Function Documentation	70

RTRManagedObject - Access Sequentially	70
Function Documentation	70
RTRManagedObject - Access Randomly	70
Function Documentation	71
RTRManagedObject - Client management	72
Function Documentation	72
RTRManagedObject - Operations from RTRLockableObj	72
Function Documentation	73
RTRManagedObject - Private Implementation	73
Function Documentation	73
RTRManagedObject - Private Event processing	73
Function Documentation	73
RTRManagedObjectClient - Event processing	74
Function Documentation	74
RTRManagedObjectIterator - Attributes	75
Function Documentation	75
RTRManagedObjectIterator - State	75
Function Documentation	75
RTRManagedObjectIterator - Access	75
Function Documentation	75
RTRManagedObjectIterator - Operations	75
Function Documentation	75
RTRManagedVariableIterator - Attributes	76
Function Documentation	76
RTRManagedVariableIterator - State	76
Function Documentation	76
RTRManagedVariableIterator - Access	76
Function Documentation	76
RTRManagedVariableIterator - Operations	77
Function Documentation	77
RTRManagedObjectDirectory - Attributes	77
Function Documentation	77
RTRManagedObjectDirectory - Access	77
Function Documentation	77
RTRManagedObjectDirectory - Insertion	78

Function Documentation	78
RTRManagedObjectDirectory - Deletion	78
Function Documentation	78
RTRManagedObjectDirectory - Client management	78
Function Documentation	78
RTRManagedStringConfig - Assignment	79
Function Documentation	79
RTRManagedStringConfig - Attributes	79
Function Documentation	79
RTRManagedString - Access	80
Function Documentation	80
RTRManagedString - Transformation	80
Function Documentation	80
RTRManagedString - Comparison	80
Function Documentation	80
RTRManagedString - Attributes	80
Function Documentation	80
RTRManagedString - Assignment	81
Function Documentation	81
RTRManagedString - Operations	81
Function Documentation	81
RTRManagedVariable - Identity	81
Function Documentation	81
RTRManagedVariable - Attributes	81
Function Documentation	81
RTRManagedVariable - Transformation	82
Function Documentation	82
RTRManagedVariable - Client management	83
Function Documentation	83
RTRManagedVariable - Operations	84
Function Documentation	84
RTRManagedVariable - Private Implementation	84
Function Documentation	84
RTRManagedVariableClient - Event processing	84
Function Documentation	84

RTRObjctId - Attributes	85
Function Documentation	85
RTRObjctId - State	85
Function Documentation	85
RTRObjctId - Access	85
Function Documentation	85
RTRObjctId - Query	86
Function Documentation	86
RTRObjctId - Comparison	86
Function Documentation	86
RTRObjctId - Transformation	86
Function Documentation	87
RTRObjctId - Assignment	87
Function Documentation	87
RTRObjctId - Modification	87
Function Documentation	87
RTRObjctId - OBSOLETE	88
Function Documentation	88
RTRPublicBooleanConfig - Assignment.....	88
Function Documentation	88
RTRPublicBooleanConfig - Operations	88
Function Documentation	88
RTRPublicBoolean - Assignment	89
Function Documentation	89
RTRPublicBoolean - Operations	89
Function Documentation	89
RTRPublicCounter - Operations	89
Function Documentation	89
RTRPublicGaugeConfig - Assignment	89
Function Documentation	89
RTRPublicGaugeConfig - Operations	90
Function Documentation	90
RTRPublicGauge - Assignment.....	91
Function Documentation	91
RTRPublicGauge - Operations	91

Function Documentation	92
RTRPublicLargeNumeric - Assignment.....	93
RTRPublicLargeNumeric - Operations	93
RTRPublicNumericConfig - Assignment.....	93
Function Documentation	93
RTRPublicNumericConfig - Operations.....	94
Function Documentation	94
RTRPublicNumericRange - Assignment	94
Function Documentation	94
RTRPublicNumericRange - Operations.....	94
Function Documentation	94
RTRPublicNumeric - Assignment	95
RTRPublicNumeric - Operations	95
RTRPublicObject - Operations	95
Function Documentation	95
RTRProxyManagedObjectClassDirectory - Attributes.....	96
RTRProxyManagedObjectClassDirectory - Query	96
Function Documentation	96
RTRProxyManagedObjectClassDirectory - Access Randomly	96
Function Documentation	96
RTRProxyManagedObjectClassDirectory - Access Sequentially	96
Function Documentation	97
RTRProxyManagedObjectClassDirectory - Event processing from RTRProxyManagedObjectServerPoolClient ...	97
Function Documentation	97
RTRProxyManagedObjectClassDirectory - Event client management	97
Function Documentation	97
RTRProxyManagedObjectClassDirectoryClient - Event processing	97
Function Documentation	98
RTRProxyManagedObjectPool - Query	98
Function Documentation	98
RTRProxyManagedObjectPool - Attributes	98
RTRProxyManagedObjectPool - Access Randomly	98
Function Documentation	98
RTRProxyManagedObjectPool - Access Sequentially	98
Function Documentation	98

RTRProxyManagedObjectPool - Event processing	99
Function Documentation	99
RTRProxyManagedObjectPool - Event client management	99
Function Documentation	99
RTRProxyManagedObjectPoolClient - Event processing	99
Function Documentation	100
RTRProxyManagedObjectServerClient - Event processing	100
Function Documentation	100
RTRProxyManagedObjectServerPool - Access Sequentially	100
Function Documentation	100
RTRProxyManagedObjectServerPool - Event client management	101
Function Documentation	101
RTRProxyManagedObjectServerPoolClient - Event processing	101
Function Documentation	101
RTRPublicStringConfig - Assignment	101
Function Documentation	102
RTRPublicStringConfig - Operations	102
Function Documentation	102
RTRPublicString - Assignment	102
Function Documentation	102
RTRPublicString - Operations	102
Function Documentation	103
RTRProxyManagedObjectHandle - Attributes	103
Function Documentation	103
RTRProxyManagedVariableHandle - Attributes	103
Function Documentation	103
RTRProxyManagedObjectHandleIterator - Attributes	104
Function Documentation	104
RTRProxyManagedObjectHandleIterator - State	104
Function Documentation	104
RTRProxyManagedObjectHandleIterator - Access	104
Function Documentation	104
RTRProxyManagedObjectHandleIterator - Operations	104
Function Documentation	104
RTRProxyManagedVarHandleIterator - Attributes	105

Function Documentation	105
RTRProxyManagedVarHandleIterator - State	105
Function Documentation	105
RTRProxyManagedVarHandleIterator - Access.....	105
Function Documentation	105
RTRProxyManagedVarHandleIterator - Operations.....	106
Function Documentation	106
RTRProxyManagedBoolean - Comparison	106
Function Documentation	106
RTRProxyManagedBoolean - Access	106
Function Documentation	106
RTRProxyManagedBoolean - Transformation	107
Function Documentation	107
RTRProxyManagedBoolean - Attributes	107
Function Documentation	107
RTRProxyManagedBoolean - Assignment.....	107
Function Documentation	107
RTRProxyManagedBoolean - Operations	108
Function Documentation	108
RTRProxyManagedBooleanConfig - Assignment	108
Function Documentation	108
RTRProxyManagedBooleanConfig - Attributes	108
Function Documentation	108
RTRProxyManagedCounter - Comparison	109
Function Documentation	109
RTRProxyManagedCounter - Access	109
Function Documentation	109
RTRProxyManagedCounter - Transformation.....	109
Function Documentation	109
RTRProxyManagedCounter - Operations	109
Function Documentation	109
RTRProxyManagedGauge - Transformation.....	110
Function Documentation	110
RTRProxyManagedGauge - Attributes.....	110
Function Documentation	110

RTRProxyManagedGauge - Operations	111
Function Documentation	111
RTRProxyManagedGaugeConfig - Attributes	111
Function Documentation	111
RTRProxyManagedLargeNumeric - Comparison.....	112
Function Documentation	112
RTRProxyManagedLargeNumeric - Access	112
Function Documentation	112
RTRProxyManagedLargeNumeric - Transformation	112
Function Documentation	112
RTRProxyManagedNumeric - Comparison	113
Function Documentation	113
RTRProxyManagedNumeric - Access.....	113
Function Documentation	113
RTRProxyManagedNumeric - Transformation	113
Function Documentation	113
RTRProxyManagedNumericConfig - Access	114
Function Documentation	114
RTRProxyManagedNumericConfig - Attributes.....	114
Function Documentation	115
RTRProxyManagedNumericConfig - Assignment	116
Function Documentation	116
RTRProxyManagedNumericConfig - Operations	116
Function Documentation	116
RTRProxyManagedNumericRange - Attributes	116
Function Documentation	116
RTRProxyManagedNumericRange - Assignment.....	117
Function Documentation	117
RTRProxyManagedNumericRange - Operations	117
Function Documentation	117
RTRProxyManagedObject - Identity	117
Function Documentation	117
RTRProxyManagedObject - Attributes	118
Function Documentation	118
RTRProxyManagedObject - State	118

Function Documentation	118
RTRProxyManagedObject - Query.....	118
Function Documentation	118
RTRProxyManagedObject - Access Sequentially	119
Function Documentation	119
RTRProxyManagedObject - Access Randomly	119
Function Documentation	119
RTRProxyManagedObject - Event client management.....	121
Function Documentation	121
RTRProxyManagedObject - Operations from RTRLockableObj.....	122
Function Documentation	122
RTRProxyManagedObjectClient - Event processing	122
Function Documentation	123
RTRProxyManagedObjectServer - Attributes	123
Function Documentation	124
RTRProxyManagedObjectServer - State	124
Function Documentation	124
RTRProxyManagedObjectServer - Access Sequentially	124
Function Documentation	124
RTRProxyManagedObjectServer - Access Randomly.....	124
Function Documentation	124
RTRProxyManagedObjectServer - Event client management.....	124
Function Documentation	125
RTRProxyManagedString - Access.....	125
Function Documentation	125
RTRProxyManagedString - Transformation	125
Function Documentation	125
RTRProxyManagedString - Comparison	126
Function Documentation	126
RTRProxyManagedString - Attributes	126
Function Documentation	126
RTRProxyManagedString - Assignment.....	126
Function Documentation	126
RTRProxyManagedString - Operations.....	126
Function Documentation	127

RTRProxyManagedStringConfig - Access	127
Function Documentation	127
RTRProxyManagedStringConfig - Attributes	127
Function Documentation	127
RTRProxyManagedStringConfig - Assignment	128
Function Documentation	128
RTRProxyManagedVariable - Identify	128
Function Documentation	128
RTRProxyManagedVariable - Attributes	129
Function Documentation	129
RTRProxyManagedVariable - State	129
Function Documentation	129
RTRProxyManagedVariable - Transformation	129
Function Documentation	130
RTRProxyManagedVariable - Event client management	132
Function Documentation	132
RTRProxyManagedVariable - Operations from RTRLockableObj	132
Function Documentation	133
RTRProxyManagedVariableClient - Event processing	133
Function Documentation	133
RTRExternalValue - Assignment	133
RTRExternalValue - Conversion Functions	134
Function Documentation	134
RTRListOfExternalValue - Assignment operator	134
RTRListOfExternalValue - Iteration	134
Function Documentation	134
RTRListOfExternalValue - Extraction	135
Function Documentation	135
RTRListOfExternalValue - Limits	135
Function Documentation	135
RTRString - Attributes	135
Function Documentation	135
RTRString - Modify in entirety	136
Function Documentation	136
RTRString - Modify in part	137

Function Documentation	137
RTRString - Truncate.....	138
Function Documentation	138
RTRString - Comparison	138
Function Documentation	138
RTRString - Access	138
Function Documentation	139
RTRString - Query	139
Function Documentation	139
RTRString - Transform	139
Function Documentation	140
RTRString - Operators.....	140
Function Documentation	140
RTRString - Operations	141
Function Documentation	141
RTRString - OBSOLETE	141
Function Documentation	142
RTRSelectNotifier - From RTREventNotifier	142
Function Documentation	142
RTRSelectNotifier - From RTREventNotifierImp	142
Function Documentation	143
RTRShmMOServerMemPool - State.....	143
RTRShmMOServerMemPool - Attributes.....	143
Function Documentation	143
RTRShmMOServerMemPool - Util.....	143
RTRShmMOServerMemPool - Event processing	143
Function Documentation	143
RTRShmMOServer - State	144
Function Documentation	144
RTRShmMOServer - Identity.....	144
RTRShmMOServer - Attributes	144
Function Documentation	144
RTRShmMOServer - Operations.....	144
Function Documentation	145
RTRShmProxyManagedObjectClassDirFactory - Operations from RTRProxyManagedObjectDirFactory	145

RTRShmProxyManagedObjectServerPool - Operations	145
Function Documentation	145
RTRShmServer - State	145
Function Documentation	145
RTRShmServer - Identity	146
RTRShmServer - Attributes	146
Function Documentation	146
RTRShmServer - Operations	146
Function Documentation	146
RTRServerSharedMemoryRoot - Identity	147
RTRServerSharedMemoryRoot - Attributes	147
Function Documentation	147
RTRServerSharedMemoryRoot - State	147
Function Documentation	147
RTRServerSharedMemoryRoot - Access	148
RTRServerSharedMemoryRoot - Event processing	148
Function Documentation	148
RTRTimerCmd - Attributes	148
Function Documentation	148
RTRTimerCmd - Comparison	148
RTRTimerCmd - State	149
Function Documentation	149
RTRTimerCmd - Operations	149
Function Documentation	149
RTRTimerCmd - Event processing	149
Function Documentation	149
RTRTimerCmd - Implementation	149
Function Documentation	149
RTRWindowsNotifier - Control from RTREventNotifier	150
Function Documentation	150
RTRWindowsNotifier - Control from RTREventNotifierImp	150
Function Documentation	150
RTRXFileDb - State	150
Function Documentation	151
RTRXFileDb - Query	151

Function Documentation	151
RTRXFileDb - Access	151
Function Documentation	151
RTRXFileDb - OBSOLETE	152
Function Documentation	152
RTRXFileDb - File features	152
Function Documentation	152
Chapter 5 Refinitiv Management Classes Class Documentation	153
RTRApplicationId Class Reference	153
Public Member Functions	153
Detailed Description	153
Constructor & Destructor Documentation	154
RTRCmdLine Class Reference	154
Public Member Functions	154
Static Public Attributes	154
Friends 154	
Detailed Description	155
Constructor & Destructor Documentation	155
Member Function Documentation	155
Member Data Documentation	156
RTRCmdLineArg Class Reference	156
Public Types	157
Public Member Functions	157
Static Public Attributes	158
Friends 158	
Detailed Description	158
Member Enumeration Documentation	158
Constructor & Destructor Documentation	158
RTRCmdLineString Class Reference	158
Public Member Functions	159
Detailed Description	159
Constructor & Destructor Documentation	160
Member Function Documentation	160
RTRConfig Class Reference	160
Static Public Member Functions	160

Detailed Description	160
Member Function Documentation	160
RTRConfigDb Class Reference	160
Public Member Functions	161
Detailed Description	161
Constructor & Destructor Documentation	162
RTRConfigVariable Class Reference	162
Public Member Functions	163
Detailed Description	163
Constructor & Destructor Documentation	164
RTRDefaultLogger Class Reference	164
Public Member Functions	165
Static Public Attributes	165
Detailed Description	166
Constructor & Destructor Documentation	166
RTREventNotifier Class Reference	166
Public Member Functions	167
Friends	167
Detailed Description	167
Constructor & Destructor Documentation	168
RTREventNotifierImp Class Reference	168
Public Member Functions	169
Detailed Description	170
Constructor & Destructor Documentation	170
RTRExternalValue Class Reference	170
Public Member Functions	171
Detailed Description	171
Constructor & Destructor Documentation	172
RTRListOfExternalValue Class Reference	172
Public Member Functions	172
Detailed Description	172
Constructor & Destructor Documentation	173
RTRLock Class Reference	173
Public Member Functions	173
Detailed Description	173

Constructor & Destructor Documentation	173
RTRLockableObj Class Reference.....	173
Public Member Functions.....	174
Static Public Attributes	175
Detailed Description	175
Constructor & Destructor Documentation	175
Member Data Documentation	175
RTRManagedBoolean Class Reference	175
Public Member Functions.....	177
Friends 177	
Detailed Description	177
Constructor & Destructor Documentation	178
RTRManagedBooleanConfig Class Reference	178
Public Member Functions.....	179
Friends 179	
Detailed Description	179
Constructor & Destructor Documentation	180
RTRManagedCounter Class Reference.....	180
Public Member Functions.....	181
Static Public Attributes	181
Friends 181	
Detailed Description	181
Constructor & Destructor Documentation	182
RTRManagedGauge Class Reference.....	182
Public Member Functions.....	183
Friends 183	
Detailed Description	184
Constructor & Destructor Documentation	184
RTRManagedGaugeConfig Class Reference	184
Public Member Functions.....	186
Friends 186	
Detailed Description	187
Constructor & Destructor Documentation	187
RTRManagedLargeNumeric Class Reference.....	187
Public Member Functions.....	188

Static Public Attributes	188
Friends 188	
Detailed Description	188
Constructor & Destructor Documentation	188
RTRManagedNumeric Class Reference	188
Public Member Functions.....	190
Static Public Attributes	190
Friends 190	
Detailed Description	190
Constructor & Destructor Documentation	191
RTRManagedNumericConfig Class Reference.....	191
Public Member Functions.....	192
Friends 193	
Detailed Description	193
Constructor & Destructor Documentation	193
RTRManagedNumericRange Class Reference	193
Public Member Functions.....	194
Friends 194	
Detailed Description	194
RTRManagedObjDirClient Class Reference	195
Public Member Functions.....	195
Detailed Description	195
Member Function Documentation	195
RTRManagedObjDirRootIterator Class Reference	196
Public Member Functions.....	196
Detailed Description	196
Constructor & Destructor Documentation	196
Member Function Documentation	196
RTRManagedObject Class Reference	197
Public Types.....	198
Public Member Functions.....	198
Static Public Attributes	200
Friends 200	
Detailed Description	200
Member Enumeration Documentation	201

Constructor & Destructor Documentation	201
Member Data Documentation	201
RTRManagedObjectClient Class Reference	201
Public Member Functions.....	201
Detailed Description	202
RTRManagedObjectDirectory Class Reference.....	202
Public Member Functions.....	202
Friends 203	
Detailed Description	203
Constructor & Destructor Documentation	203
RTRManagedObjectIterator Class Reference.....	203
Public Member Functions.....	203
Detailed Description	204
Constructor & Destructor Documentation	204
RTRManagedProcess Class Reference.....	204
Public Member Functions.....	205
Detailed Description	205
Constructor & Destructor Documentation	206
RTRManagedString Class Reference	206
Public Member Functions.....	207
Friends 207	
Detailed Description	207
Constructor & Destructor Documentation	208
RTRManagedStringConfig Class Reference.....	208
Public Member Functions.....	209
Friends 209	
Detailed Description	210
Constructor & Destructor Documentation	210
RTRManagedVariable Class Reference	210
Public Types.....	211
Public Member Functions.....	211
Static Public Attributes	211
Friends 211	
Detailed Description	212
Member Enumeration Documentation	212

Constructor & Destructor Documentation	212
Member Data Documentation	212
RTRManagedVariableClient Class Reference	212
Public Member Functions.....	212
Detailed Description	213
RTRManagedVariableIterator Class Reference	213
Public Member Functions.....	213
Detailed Description	213
Constructor & Destructor Documentation	213
RTRMgmtAction Class Reference.....	213
Public Member Functions.....	214
Detailed Description	214
Constructor & Destructor Documentation	214
RTRMgmtEvent Class Reference	215
Public Member Functions.....	216
Static Public Member Functions	217
Static Public Attributes	217
Detailed Description	217
Constructor & Destructor Documentation	218
Member Function Documentation	218
RTRObjectId Class Reference	218
Public Member Functions.....	219
Static Public Attributes	220
Friends 220	
Detailed Description	220
Constructor & Destructor Documentation	220
RTRProxyManagedBoolean Class Reference	220
Public Member Functions.....	222
Detailed Description	222
Constructor & Destructor Documentation	223
RTRProxyManagedBooleanConfig Class Reference.....	223
Public Member Functions.....	224
Detailed Description	224
Constructor & Destructor Documentation	225
RTRProxyManagedCounter Class Reference	225

Public Member Functions.....	226
Static Public Attributes	226
Detailed Description	226
Constructor & Destructor Documentation	227
RTRProxyManagedGauge Class Reference	227
Public Member Functions.....	228
Detailed Description	229
Constructor & Destructor Documentation	229
RTRProxyManagedGaugeConfig Class Reference	229
Public Member Functions.....	231
Detailed Description	231
Constructor & Destructor Documentation	232
RTRProxyManagedLargeNumeric Class Reference	232
Public Member Functions.....	233
Static Public Attributes	233
Detailed Description	233
Constructor & Destructor Documentation	233
RTRProxyManagedNumeric Class Reference	233
Public Member Functions.....	235
Static Public Attributes	235
Detailed Description	235
Constructor & Destructor Documentation	235
RTRProxyManagedNumericConfig Class Reference	235
Public Member Functions.....	237
Detailed Description	238
Constructor & Destructor Documentation	238
RTRProxyManagedNumericRange Class Reference	238
Public Member Functions.....	239
Detailed Description	239
Constructor & Destructor Documentation	240
RTRProxyManagedObject Class Reference	240
Public Types.....	241
Public Member Functions.....	241
Friends 242	
Detailed Description	242

Member Enumeration Documentation	243
Constructor & Destructor Documentation	243
RTRProxyManagedObjectClassDirectory Class Reference	243
Public Member Functions.....	244
Detailed Description	244
Constructor & Destructor Documentation	245
RTRProxyManagedObjectClassDirectoryClient Class Reference	245
Public Member Functions.....	245
Detailed Description	245
Constructor & Destructor Documentation	245
RTRProxyManagedObjectClient Class Reference.....	246
Public Member Functions.....	246
Detailed Description	246
Constructor & Destructor Documentation	247
RTRProxyManagedObjectHandle Class Reference	247
Public Member Functions.....	248
Friends 248	
Detailed Description	248
Constructor & Destructor Documentation	248
RTRProxyManagedObjectHandleIterator Class Reference	249
Public Member Functions.....	249
Detailed Description	249
Constructor & Destructor Documentation	249
RTRProxyManagedObjectPool Class Reference.....	249
Public Member Functions.....	250
Detailed Description	250
Constructor & Destructor Documentation	250
RTRProxyManagedObjectPoolClient Class Reference	250
Public Member Functions.....	251
Detailed Description	251
Constructor & Destructor Documentation	251
RTRProxyManagedObjectServer Class Reference	251
Public Member Functions.....	252
Friends 252	
Detailed Description	252

Constructor & Destructor Documentation	252
RTRProxyManagedObjectServerClient Class Reference	253
Public Member Functions	253
Detailed Description	253
Constructor & Destructor Documentation	253
RTRProxyManagedObjectServerPool Class Reference	253
Public Member Functions	254
Detailed Description	254
Constructor & Destructor Documentation	254
RTRProxyManagedObjectServerPoolClient Class Reference	255
Public Member Functions	255
Detailed Description	255
Constructor & Destructor Documentation	255
RTRProxyManagedString Class Reference	256
Public Member Functions	256
Detailed Description	257
Constructor & Destructor Documentation	257
RTRProxyManagedStringConfig Class Reference	257
Public Member Functions	259
Detailed Description	259
Constructor & Destructor Documentation	259
RTRProxyManagedVarHandleIterator Class Reference	259
Public Member Functions	259
Detailed Description	260
Constructor & Destructor Documentation	260
RTRProxyManagedVariable Class Reference	260
Public Member Functions	260
Friends 262	
Detailed Description	262
Constructor & Destructor Documentation	262
RTRProxyManagedVariableClient Class Reference	262
Public Member Functions	262
Detailed Description	263
Constructor & Destructor Documentation	263
RTRProxyManagedVariableHandle Class Reference	263

Public Types.....	264
Public Member Functions.....	264
Static Public Member Functions	264
Detailed Description	264
Member Enumeration Documentation	264
Constructor & Destructor Documentation	264
RTRPublicBoolean Class Reference.....	265
Public Member Functions.....	266
Detailed Description	266
Constructor & Destructor Documentation	267
RTRPublicBooleanConfig Class Reference	267
Public Member Functions.....	268
Detailed Description	269
Constructor & Destructor Documentation	269
RTRPublicCounter Class Reference	269
Public Member Functions.....	270
Detailed Description	270
Constructor & Destructor Documentation	271
RTRPublicGauge Class Reference	271
Public Member Functions.....	272
Detailed Description	273
Constructor & Destructor Documentation	273
RTRPublicGaugeConfig Class Reference.....	273
Public Member Functions.....	275
Detailed Description	276
Constructor & Destructor Documentation	276
RTRPublicLargeNumeric Class Reference	276
Public Member Functions.....	277
Detailed Description	278
Constructor & Destructor Documentation	278
RTRPublicNumeric Class Reference	278
Public Member Functions.....	279
Detailed Description	280
Constructor & Destructor Documentation	280
RTRPublicNumericConfig Class Reference.....	280

Public Member Functions.....	281
Detailed Description	282
Constructor & Destructor Documentation	282
RTRPublicNumericRange Class Reference.....	282
Public Member Functions.....	284
Detailed Description	285
Constructor & Destructor Documentation	285
RTRPublicObject Class Reference	285
Public Member Functions.....	287
Detailed Description	287
Constructor & Destructor Documentation	287
RTRPublicObjectLock Class Reference.....	288
Public Member Functions.....	288
Detailed Description	288
Constructor & Destructor Documentation	288
RTRPublicString Class Reference	288
Public Member Functions.....	290
Detailed Description	290
Constructor & Destructor Documentation	291
RTRPublicStringConfig Class Reference	291
Public Member Functions.....	292
Detailed Description	292
Constructor & Destructor Documentation	293
RTRSelectNotifier Class Reference	293
Public Member Functions.....	293
Static Public Member Functions	294
Static Public Attributes	294
Friends 294	
Detailed Description	294
Constructor & Destructor Documentation	294
Member Function Documentation.....	294
Member Data Documentation	295
RTRServerSharedMemoryRoot Class Reference.....	295
Public Member Functions.....	296
Static Public Attributes	296

Friends	297
Detailed Description	297
Constructor & Destructor Documentation	297
Member Data Documentation	297
RTRSharedMemoryStats Class Reference	297
Public Member Functions	299
Detailed Description	299
Constructor & Destructor Documentation	299
Member Function Documentation	299
RTRShmMOServer Class Reference	299
Public Member Functions	300
Static Public Attributes	300
Detailed Description	300
Constructor & Destructor Documentation	301
RTRShmMOServerMemPool Class Reference	301
Public Member Functions	302
Static Public Attributes	302
Friends	302
Detailed Description	303
Constructor & Destructor Documentation	303
Member Data Documentation	303
RTRShmProxyManagedObjectClassDirFactory Class Reference	303
Public Member Functions	304
Detailed Description	304
Constructor & Destructor Documentation	304
RTRShmProxyManagedObjectServerPool Class Reference	305
Public Member Functions	305
Detailed Description	306
Constructor & Destructor Documentation	306
RTRShmServer Class Reference	306
Public Member Functions	306
Static Public Attributes	307
Detailed Description	307
Constructor & Destructor Documentation	307
RTRString Class Reference	308

Public Member Functions.....	308
Static Public Attributes	311
Friends 311	
Detailed Description	311
Constructor & Destructor Documentation	311
Member Data Documentation	312
RTRTimerCmd Class Reference.....	312
Public Member Functions.....	313
Static Public Attributes	313
Friends 314	
Detailed Description	314
Constructor & Destructor Documentation	314
RTRWindowsNotifier Class Reference.....	314
Public Member Functions.....	315
Static Public Attributes	315
Friends 315	
Detailed Description	315
Constructor & Destructor Documentation	316
Member Function Documentation	316
Member Data Documentation	316
RTRXEventNotifier Class Reference.....	316
Public Member Functions.....	317
Static Public Member Functions	317
Static Public Attributes	317
Detailed Description	317
Constructor & Destructor Documentation	318
Member Data Documentation	318
RTRXFileDb Class Reference.....	318
Public Member Functions.....	318
Detailed Description	319
Constructor & Destructor Documentation	319
RTRXtNotifier Class Reference.....	319
Public Member Functions.....	320
Public Attributes	320
Static Public Attributes	321

Detailed Description	321
Constructor & Destructor Documentation	321
Member Function Documentation	321
Member Data Documentation	321
RTRXViewNotifier Class Reference	322
Public Member Functions.....	323
Static Public Attributes	323
Detailed Description	323
Constructor & Destructor Documentation	323
Member Function Documentation	323

Chapter 1 Overview

The Refinitiv Management Classes provides a stable implementation of managing and monitoring solutions: RMC Publisher and RMC Consumer API. RMC API uses the concept of shared memory which offers fast and efficient benefits. The publisher applications are able to use the API without affecting the performance of the application, while the consumer applications are able to use the API to monitor the publisher application and receive notifications as the information changes.

RMC Publisher Classes:

- [RTRPublicBoolean](#)
- [RTRPublicBooleanConfig](#)
- [RTRPublicCounter](#)
- [RTRPublicGauge](#)
- [RTRPublicGaugeConfig](#)
- [RTRPublicLargeNumeric](#)
- [RTRPublicNumeric](#)
- [RTRPublicNumericConfig](#)
- [RTRPublicNumericRange](#)
- [RTRPublicObject](#)
- [RTRPublicString](#)
- [RTRPublicStringConfig](#)

RMC Consumer Classes:

- [RTRProxyManagedBoolean](#)
- [RTRProxyManagedBooleanConfig](#)
- [RTRProxyManagedCounter](#)
- [RTRProxyManagedGauge](#)
- [RTRProxyManagedGaugeConfig](#)
- [RTRProxyManagedLargeNumeric](#)
- [RTRProxyManagedNumeric](#)
- [RTRProxyManagedNumericConfig](#)
- [RTRProxyManagedNumericRange](#)
- [RTRProxyManagedObject](#)
- [RTRProxyManagedObjectClient](#)
- [RTRProxyManagedObjectClassDirectory](#)
- [RTRProxyManagedObjectClassDirectoryClient](#)
- [RTRProxyManagedObjectHandle](#)

- [RTRProxyManagedObjectHandleIterator](#)
- [RTRProxyManagedObjectPool](#)
- [RTRProxyManagedObjectPoolClient](#)
- [RTRProxyManagedObjectServer](#)
- [RTRProxyManagedObjectServerClient](#)
- [RTRProxyManagedObjectServerPool](#)
- [RTRProxyManagedObjectServerPoolClient](#)
- [RTRProxyManagedString](#)
- [RTRProxyManagedStringConfig](#)
- [RTRProxyManagedVariable](#)
- [RTRProxyManagedVariableClient](#)
- [RTRProxyManagedVariableHandle](#)
- [RTRProxyManagedVarHandleIterator](#)

The following classes are specific to the shared memory implementation of the Publisher and Consumer API provided in RMC 2.1.0. They should be used as the entry points into the API.

- [RTRServerSharedMemoryRoot](#)
- [RTRShmMOServer](#) [Helper Class]
- [RTRShmMOServerMemPool](#)
- [RTRShmProxyManagedObjectClassDirFactory](#)
- [RTRShmProxyManagedObjectServerPool](#)
- [RTRShmServer](#) [Helper Class]

Other Utilities Classes:

The following classes are specific to the API and are recommended to use with the application.

- Main loop
 - [RTRSelectNotifier](#)
- Timer
 - [RTRTimerCmd](#)
- Logging
 - [RTRDefaultLogger](#)
 - [RTRMgmtEvent](#)
- Config
 - [RTRConfig](#)
 - [RTRConfigDb](#)
- Command Line
 - [RTRCmdLine](#)

- [RTRCmdLineString](#)
- Data Type
 - [RTRString](#)

Other Type Definitions:

- `typedef unsigned int` RTRBOOL
- `#define` RTRTRUE 1
- `#define` RTRFALSE 0

For other classes, please refer to [Class List index](#).

Chapter 2 Refinitiv Management Classes 2.1.0 Module Index

Refinitiv Management Classes 2.1.0 Modules

Here is a list of all modules:

RTRConfigDb - State	46
RTRConfigDb - Query	46
RTRConfigDb - Access	46
RTRConfigDb - OBSOLETE	47
RTRConfigVariable - Attributes	47
RTRConfigVariable - State	47
RTRConfigVariable - Assignment	47
RTRConfigVariable - Compatibility	48
RTRCmdLineArg - Attributes	48
RTRCmdLineArg - State	49
RTRCmdLineArg - Transformation	49
RTRCmdLineArg - Query	49
RTRCmdLineArg - Operations	49
RTRDefaultLogger - Attributes	49
RTRDefaultLogger - Utilities	50
RTREventNotifierImp - Access	50
RTREventNotifierImp - Insert	51
RTREventNotifierImp - Remove	51
RTREventNotifierImp - Implementation	52
RTREventNotifier - Query	52
RTREventNotifier - Access	53
RTREventNotifier - Insert	53
RTREventNotifier - Remove	54
RTREventNotifier - Control	55
RTREventNotifier - Control - OBSOLETE	55
RTRLockableObj - Operations	55
RTRLockableObj - State	56
RTRManagedBooleanConfig - Assignment	56
RTRManagedBooleanConfig - Attributes	56
RTRManagedBoolean - Comparison	57
RTRManagedBoolean - Access	57
RTRManagedBoolean - Transformation	57
RTRManagedBoolean - State	57
RTRManagedBoolean - Assignment	57
RTRManagedBoolean - Operations	58
RTRManagedCounter - Comparison	58
RTRManagedCounter - Access	58
RTRManagedCounter - Transformation	58
RTRManagedCounter - Operations	59
RTRManagedGaugeConfig - Attributes	59
RTRMgmtAction - Identity	59
RTRMgmtAction - Attributes	60
RTRMgmtAction - State	60
RTRMgmtAction - Operations	60
RTRMgmtEvent - Attributes	61
RTRMgmtEvent - Comparison	61

RTRMgmtEvent - Operation	62
RTRMgmtEvent - Operations Assignment	62
RTRMgmtEvent - OBSOLETE	62
RTRManagedGauge - Attributes	63
RTRManagedGauge - Operations	64
RTRManagedNumericConfig - Attributes	64
RTRManagedNumericConfig - Assignment	65
RTRManagedNumericRange - Attributes	66
RTRManagedNumericRange - Assignment	66
RTRManagedNumericRange - Operations	66
RTRManagedLargeNumeric - Comparison	67
RTRManagedLargeNumeric - Access	68
RTRManagedLargeNumeric - Transformation	68
RTRManagedNumeric - Comparison	67
RTRManagedNumeric - Access	68
RTRManagedNumeric - Transformation	68
RTRManagedObject - Identity	68
RTRManagedObject - Attributes	68
RTRManagedObject - State	69
RTRManagedObject - Query	70
RTRManagedObject - Access Sequentially	70
RTRManagedObject - Access Randomly	70
RTRManagedObject - Client management	72
RTRManagedObject - Operations from RTRLockableObj	72
RTRManagedObject - Private Implementation	73
RTRManagedObject - Private Event processing	73
RTRManagedObjectClient - Event processing	74
RTRManagedObjectIterator - Attributes	75
RTRManagedObjectIterator - State	75
RTRManagedObjectIterator - Access	75
RTRManagedObjectIterator - Operations	75
RTRManagedVariableIterator - Attributes	76
RTRManagedVariableIterator - State	76
RTRManagedVariableIterator - Access	76
RTRManagedVariableIterator - Operations	77
RTRManagedObjectDirectory - Attributes	77
RTRManagedObjectDirectory - Access	77
RTRManagedObjectDirectory - Insertion	78
RTRManagedObjectDirectory - Deletion	78
RTRManagedObjectDirectory - Client management	78
RTRManagedStringConfig - Assignment	79
RTRManagedStringConfig - Attributes	79
RTRManagedString - Access	80
RTRManagedString - Transformation	80
RTRManagedString - Comparison	80
RTRManagedString - Attributes	80
RTRManagedString - Assignment	81
RTRManagedString - Operations	81
RTRManagedVariable - Identity	81
RTRManagedVariable - Attributes	81
RTRManagedVariable - Transformation	82
RTRManagedVariable - Client management	83
RTRManagedVariable - Operations	84

RTRManagedVariable - Private Implementation	84
RTRManagedVariableClient - Event processing	84
RTRObjectId - Attributes	85
RTRObjectId - State	85
RTRObjectId - Access	85
RTRObjectId - Query	86
RTRObjectId - Comparison	86
RTRObjectId - Transformation	86
RTRObjectId - Assignment	87
RTRObjectId - Modification	87
RTRObjectId - OBSOLETE	88
RTRPublicBooleanConfig - Assignment	88
RTRPublicBooleanConfig - Operations	88
RTRPublicBoolean - Assignment	89
RTRPublicBoolean - Operations	89
RTRPublicCounter - Operations	89
RTRPublicGaugeConfig - Assignment	89
RTRPublicGaugeConfig - Operations	90
RTRPublicGauge - Assignment	91
RTRPublicGauge - Operations	91
RTRPublicLargeNumeric - Assignment	93
RTRPublicLargeNumeric - Operations	94
RTRPublicNumericConfig - Assignment	93
RTRPublicNumericConfig - Operations	94
RTRPublicNumericRange - Assignment	94
RTRPublicNumericRange - Operations	94
RTRPublicNumeric - Assignment	95
RTRPublicNumeric - Operations	95
RTRPublicObject - Operations	95
RTRProxyManagedObjectClassDirectory - Attributes	96
RTRProxyManagedObjectClassDirectory - Query	96
RTRProxyManagedObjectClassDirectory - Access Randomly	96
RTRProxyManagedObjectClassDirectory - Access Sequentially	96
RTRProxyManagedObjectClassDirectory - Event processing from RTRProxyManagedObjectServerPoolClient	97
RTRProxyManagedObjectClassDirectory - Event client management	97
RTRProxyManagedObjectClassDirectoryClient - Event processing	97
RTRProxyManagedObjectPool - Query	98
RTRProxyManagedObjectPool - Attributes	98
RTRProxyManagedObjectPool - Access Randomly	98
RTRProxyManagedObjectPool - Access Sequentially	98
RTRProxyManagedObjectPool - Event processing	99
RTRProxyManagedObjectPool - Event client management	99
RTRProxyManagedObjectPoolClient - Event processing	99
RTRProxyManagedObjectServerClient - Event processing	100
RTRProxyManagedObjectServerPool - Access Sequentially	100
RTRProxyManagedObjectServerPool - Event client management	101
RTRProxyManagedObjectServerPoolClient - Event processing	101
RTRPublicStringConfig - Assignment	101
RTRPublicStringConfig - Operations	102
RTRPublicString - Assignment	102
RTRPublicString - Operations	102
RTRProxyManagedObjectHandle - Attributes	103

RTRProxyManagedVariableHandle - Attributes	103
RTRProxyManagedObjectHandleIterator - Attributes.....	104
RTRProxyManagedObjectHandleIterator - State	104
RTRProxyManagedObjectHandleIterator - Access	104
RTRProxyManagedObjectHandleIterator - Operations	104
RTRProxyManagedVarHandleIterator - Attributes	105
RTRProxyManagedVarHandleIterator - State	105
RTRProxyManagedVarHandleIterator - Access	105
RTRProxyManagedVarHandleIterator - Operations	106
RTRProxyManagedBoolean - Comparison	106
RTRProxyManagedBoolean - Access	106
RTRProxyManagedBoolean - Transformation	107
RTRProxyManagedBoolean - Attributes	107
RTRProxyManagedBoolean - Assignment.....	107
RTRProxyManagedBoolean - Operations.....	108
RTRProxyManagedBooleanConfig - Assignment	108
RTRProxyManagedBooleanConfig - Attributes	108
RTRProxyManagedCounter - Comparison.....	109
RTRProxyManagedCounter - Access	109
RTRProxyManagedCounter - Transformation	109
RTRProxyManagedCounter - Operations	109
RTRProxyManagedGauge - Transformation.....	110
RTRProxyManagedGauge - Attributes.....	110
RTRProxyManagedGauge - Operations	111
RTRProxyManagedGaugeConfig - Attributes	111
RTRProxyManagedLargeNumeric - Comparison.....	113
RTRProxyManagedLargeNumeric - Access.....	113
RTRProxyManagedLargeNumeric - Transformation	113
RTRProxyManagedNumeric - Comparison	113
RTRProxyManagedNumeric - Access.....	113
RTRProxyManagedNumeric - Transformation	113
RTRProxyManagedNumericConfig - Access	114
RTRProxyManagedNumericConfig - Attributes	114
RTRProxyManagedNumericConfig - Assignment	116
RTRProxyManagedNumericConfig - Operations	116
RTRProxyManagedNumericRange - Attributes.....	116
RTRProxyManagedNumericRange - Assignment.....	117
RTRProxyManagedNumericRange - Operations	117
RTRProxyManagedObject - Identity.....	117
RTRProxyManagedObject - Attributes	118
RTRProxyManagedObject - State	118
RTRProxyManagedObject - Query.....	118
RTRProxyManagedObject - Access Sequentially	119
RTRProxyManagedObject - Access Randomly.....	119
RTRProxyManagedObject - Event client management.....	121
RTRProxyManagedObject - Operations from RTRLockableObj	122
RTRProxyManagedObjectClient - Event processing.....	122
RTRProxyManagedObjectServer - Attributes.....	123
RTRProxyManagedObjectServer - State	124
RTRProxyManagedObjectServer - Access Sequentially.....	124
RTRProxyManagedObjectServer - Access Randomly	124
RTRProxyManagedObjectServer - Event client management	124
RTRProxyManagedString - Access.....	125

RTRProxyManagedString - Transformation	125
RTRProxyManagedString - Comparison	126
RTRProxyManagedString - Attributes	126
RTRProxyManagedString - Assignment.....	126
RTRProxyManagedString - Operations.....	126
RTRProxyManagedStringConfig - Access	127
RTRProxyManagedStringConfig - Attributes.....	127
RTRProxyManagedStringConfig - Assignment	128
RTRProxyManagedVariable - Identify	128
RTRProxyManagedVariable - Attributes	129
RTRProxyManagedVariable - State	129
RTRProxyManagedVariable - Transformation	129
RTRProxyManagedVariable - Event client management	132
RTRProxyManagedVariable - Operations from RTRLockableObj.....	132
RTRProxyManagedVariableClient - Event processing.....	133
RTRExternalValue - Assignment.....	133
RTRExternalValue - Conversion Functions.....	134
RTRLiStOfExternalValue - Assignment operator	134
RTRLiStOfExternalValue - Iteration	134
RTRLiStOfExternalValue - Extraction	135
RTRLiStOfExternalValue - Limits.....	135
RTRString - Attributes	135
RTRString - Modify in entirety	136
RTRString - Modify in part.....	137
RTRString - Truncate	138
RTRString - Comparison	138
RTRString - Access.....	138
RTRString - Query.....	139
RTRString - Transform	139
RTRString - Operators	140
RTRString - Operations.....	141
RTRString - OBSOLETE	141
RTRSelectNotifier - From RTREventNotifier	142
RTRSelectNotifier - From RTREventNotifierImp	142
RTRShmMOServerMemPool - State.....	143
RTRShmMOServerMemPool - Attributes.....	143
RTRShmMOServerMemPool - Util.....	143
RTRShmMOServerMemPool - Event processing	143
RTRShmMOServer - State.....	144
RTRShmMOServer - Identity	144
RTRShmMOServer - Attributes.....	144
RTRShmMOServer - Operations.....	144
RTRShmProxyManagedObjectClassDirFactory - Operations from	
RTRProxyManagedObjectDirFactory	145
RTRShmProxyManagedObjectServerPool - Operations.....	145
RTRShmServer - State	145
RTRShmServer - Identity	146
RTRShmServer - Attributes.....	146
RTRShmServer - Operations	146
RTRServerSharedMemoryRoot - Identity.....	147
RTRServerSharedMemoryRoot - Attributes	147
RTRServerSharedMemoryRoot - State.....	147
RTRServerSharedMemoryRoot - Access.....	148

RTRServerSharedMemoryRoot - Event processing.....	148
RTRTimerCmd - Attributes.....	148
RTRTimerCmd - Comparison.....	148
RTRTimerCmd - State.....	149
RTRTimerCmd - Operations	149
RTRTimerCmd - Event processing	149
RTRTimerCmd - Implementation.....	149
RTRWindowsNotifier - Control from RTREventNotifier	150
RTRWindowsNotifier - Control from RTREventNotifierImp	150
RTRXFileDb - State.....	150
RTRXFileDb - Query	151
RTRXFileDb - Access	151
RTRXFileDb - OBSOLETE.....	152
RTRXFileDb - File features	152

Chapter 3 Refinitiv Management Classes Class Index

Refinitiv Management Classes Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<u>RTRApplicationId</u> (The class RTRApplicationId is a descendant of <u>RTRObjectId</u> that set's itself up as <hostname>.<instance>.<appname> where appName is the name of this application (typically the name of the executable) and instance is a numeric identifier which uniquely identifies this executable from other similar executables on the same host)	153
<u>RTRCmdLine</u> (It is assumed that there is only one instance of RTRCmdLine in an application. That instance should be accessed from <u>RTRCmdLine::cmdLine</u> . RTRCmdLine provides built in help flag (tag is "?"). Unused elements of argv are available from <u>leftOvers()</u> (a list of RTRCmdLineData).)	154
<u>RTRCmdLineArg</u> (RTRCmdLineArg is the base class for command line arguments. This class includes the base constructor, accessor methods, state checking methods, and <u>stringValue()</u> for getting a <u>RTRString</u> for the argument's value.)	156
<u>RTRCmdLineString</u> (Descendant of <u>RTRCmdLineArg</u> which provides type checking. Strings provide cast operator to <u>RTRString</u> .)	158
<u>RTRConfig</u> (Provides "global" access to a configuration database via the static function <u>configDb()</u> . By default, the available database will be an instance of RTRDefaultConfigDb. The application can override this by instantiating some other type of <u>RTRConfigDb</u> and "installing" it using the <u>setConfigDb()</u> function.)	160
<u>RTRConfigDb</u> (This class provides an abstract definition of a configuration database from which configuration variables may be obtained.)	160
<u>RTRConfigVariable</u> (Class RTRConfigVariable offers clients a convenient way to access configuration parameter values. If a value is not available for the parameter, the config variable will be in an error state.)	162
<u>RTRDefaultLogger</u> (This class provides three types of management event actions (file, std err and system). This logger consults the configuration database, passed on construction, to determine wether it is enabled and wether it is supposed to display its configuration on stdout.)	164
<u>RTREventNotifier</u> (RTREventNotifier is the abstract base class for event managers which provides facilities whereby clients can register to receive I/O events. Timing events are also implemented by RTREventNotifier but are made available to clients by means of the <u>RTRTimerCmd</u> abstraction.)	166
<u>RTREventNotifierImp</u> (RTREventNotifierImp is an intermediate base class for implementations of the <u>RTREventNotifier</u> abstraction. It implements most of what is required and leaves some specific implementation details for descendants.)	168
<u>RTRExternalValue</u> (This class provides the same capabilities as the <u>RTRString</u> class plus the following:)	170
<u>RTRLiStOfExternalValue</u> (A simple list of <u>RTRExternalValue</u> instances)	172
<u>RTRLock</u> (An instance of RTRLock locks a <u>RTRLockableObj</u> object passed as an argument in constructor and unlocks it when the RTRLock instance is deleted.)	173
<u>RTRLockableObj</u> (RTRLockableObj is a base class representing application component which provides lock/unlock operations on itself perceiving that its states/values could be accessed from multiple threads in applications thus need to be synchronized. Any component that wants to be made MT-safe can be a decendent class of this.)	173
<u>RTRManagedBoolean</u> (The base class for boolean managed variables. Inherits from <u>RTRManagedVariable</u> and provides services for accessing and modifying a variable of type Boolean.)	175
<u>RTRManagedBooleanConfig</u> (The base class for boolean config managed variables. Inherits from <u>RTRManagedBoolean</u> and provides services for specifying configuration and default values for the boolean variable.)	178
<u>RTRManagedCounter</u> (RTRManagedCounter is a descendant of <u>RTRManagedVariable</u> . The RTRManagedCounter can be incremented or reset (to 0); it cannot be decremented.)	180
<u>RTRManagedGauge</u> (A RTRManagedGauge is a descendant of <u>RTRManagedNumeric</u> and provides services for min/max values and low/high water marks. The low/high water marks indicate the lowest/highest values assumed by a gauge since its creation.)	182
<u>RTRManagedGaugeConfig</u> (Inherits from <u>RTRManagedGauge</u> and provides services for providing a configuration and default min/max values for the gauge.)	184
<u>RTRManagedLargeNumeric</u> (The base class for large numeric managed variables. This class provides a read-only interface to the large numeric value.)	188
<u>RTRManagedNumeric</u> (The base class for numeric managed variables. This class provides a read-only interface to the numeric value.)	188

RTRManagedNumericConfig (Numeric configs inherit from Numeric and provides services for specifying min/max values, configuration value and a default value.)	191
RTRManagedNumericRange (Numeric Ranges inherit from Numeric and provides services for specifying a min/max value.)	193
RTRManagedObjDirClient (RTRManagedObjDirClient is the base class for application components which wish to register to receive events from an instance of RTRManagedObjectDirectory)	195
RTRManagedObjDirRootIterator (An iterator for directory root object)	196
RTRManagedObject (RTRManagedObject is an abstract base class representing application components which can be accessed and managed by external management entities. Management is effected by monitoring and possibly modifying variables made available by the application component to be managed.)	197
RTRManagedObjectClient (The abstract base class for components which wish to receive object level events)	201
RTRManagedObjectDirectory (A descendant of the template class RTRDirectory which is specific to managed objects, instances of this class provide access to a list of so-called root objects, i.e. managed objects with no parent.)	202
RTRManagedObjectIterator (Stateless iteration on an object's children. Multiple instances of this can be used (in a multi-thread environment) for read access.)	203
RTRManagedProcess (RTRManagedProcess is a descendant of RTRPublicObject which provides a minimum set of variables relating to process state.)	204
RTRManagedString (The base class for string variables and provides services for accessing and modifying the value of the ManagedString.)	206
RTRManagedStringConfig (String config inherited from String and provides services for specifying configuration and default values.)	208
RTRManagedVariable (This is the base class for more specific types of managed variables. A managed variable has a name, type, and is contained by an instance of RTRManagedObject . The name of the variable must be unique within the context of the containing object.)	210
RTRManagedVariableClient (The base class for components which can register with a variable to receive changed events from that variable)	212
RTRManagedVariableIterator (Stateless iteration on an object's variables. Multiple instances of this can be used (in a multi-thread environment) for read access.)	213
RTRMgmtAction (RTRMgmtAction is the abstract base class for components which can be installed with an instance of RTRMgmtEventRouter in order to process application generated management events)	213
RTRMgmtEvent (RTRMgmtEvent provides the means for managed applications to generate events for processing by managing applications. Events have an identifier (component), text, severity, and a timestamp. The identifier is that of the component generating the event. Text is descriptive information about the event. Severity is a value between RTRMgmtEvent::Emergency and RTRMgmtEvent::Debug. A timestamp will be generated automatically if none is set.)	215
RTRObjectId (An object identifier. Both instance identifiers and class identifiers can be represented by instances of RTRObjectId)	218
RTRProxyManagedBoolean (A cloned (proxy) representation of a Boolean variable. The base class for proxy boolean managed variables. Inherits from RTRProxyManagedVariable and provides services for accessing and (conditionally) modifying a managed variable of type boolean. The managed application will accept modifications to this variable if the modifyEnabled() attribute is true. Accepted modifications are limited to setting the value of the variable to true or false.)	220
RTRProxyManagedBooleanConfig (A cloned (proxy) representation of a BooleanConfig variable. The base class for proxy boolean managed configuration variables. Inherits from RTRProxyManagedBoolean and provides additional services for accessing the stored and default values of a managed variable of type boolean config. The managed application will accept modifications to this variable if the modifyEnabled() attribute is true. Accepted modifications are limited to setting the active value of the variable to true or false (the stored and default values cannot be modified).)	223
RTRProxyManagedCounter (A cloned (proxy) representation of a Counter variable. The base class for proxy counter managed variables. Inherits from RTRProxyManagedVariable and provides services for accessing and resetting (to 0) the value of a managed variable of type counter. The managed application will always accept reset modification requests)	225
RTRProxyManagedGauge (A cloned (proxy) representation of a Gauge variable. The base class for proxy gauge managed variables. Inherits from RTRProxyManagedNumeric and provides additional services for accessing and modifying a managed variable of type gauge.)	227
RTRProxyManagedGaugeConfig (A cloned (proxy) representation of a GaugeConfig variable. The base class for proxy gauge managed configuration variables. Inherits from RTRProxyManagedGauge and provides additional services for accessing the stored and default values for the minimum and maximum values.)	229
RTRProxyManagedLargeNumeric (A cloned (proxy) representation of a Large Numeric variable. The base class for large proxy numeric managed variables. Inherits from RTRProxyManagedVariable and provides services for accessing the current value of the variable.)	233

<u>RTRProxyManagedNumeric</u> (A cloned (proxy) representation of a Numeric variable. The base class for proxy numeric managed variables. Inherits from <u>RTRProxyManagedVariable</u> and provides services for accessing the current value of the variable.)	233
<u>RTRProxyManagedNumericConfig</u> (A cloned (proxy) representation of a NumericConfig variable. The base class for proxy numeric managed configuration variables. Inherits from <u>RTRProxyManagedNumeric</u> and provides additional services for accessing the stored and default values and for modifying (conditionally) the active value.)	235
<u>RTRProxyManagedNumericRange</u> (A cloned (proxy) representation of a NumericRange variable. The base class for proxy numeric range managed variables. Inherits from <u>RTRProxyManagedNumeric</u> and provides additional services for accessing and modifying (conditionally) the current value. It also provides a fixed range of values that the variable can assume.)	238
<u>RTRProxyManagedObject</u> (A cloned (proxy) representation of a managed object. RTRProxyManagedObject is an abstract base class representing application components which can be accessed and managed by external management entities. Management is accomplished by monitoring and possibly modifying variables made available by the application component to be managed.)	240
<u>RTRProxyManagedObjectClassDirectory</u> (RTRProxyManagedObjectClassDirectory is a directory of all object handles for all managed objects of a particular type as published by a pool of <u>RTRProxyManagedObjectServer</u> . The handles can be used to retrieve instances of <u>RTRProxyManagedObject</u> from a <u>RTRProxyManagedObjectServer</u> . The pool of servers which contribute to a directory is specified when the directory is constructed. The directory will dynamically adjust its contents according to changes in the server pool and changes in the set of objects published by the servers.)	243
<u>RTRProxyManagedObjectClassDirectoryClient</u> (RTRProxyManagedObjectClassDirectoryClient is the abstract base class for application components which can register to receive events from one or more instances of <u>RTRProxyManagedObjectClassDirectory</u>)	245
<u>RTRProxyManagedObjectClient</u> (The base class for components which can register with a proxy managed object to receive change events from that proxy managed object. The notifications are grouped into five categories: (1) proxy managed object state changes, (2) the managed object has been deleted by the producer, (3) the state attribute has changed, (4) a child managed object has been added/removed, and (5) a contained managed variable has been added/removed.)	246
<u>RTRProxyManagedObjectHandle</u> (A RTRProxyManagedObjectHandle uniquely identifies a Proxy Managed Object. The handle is used to request a clone (proxy) of a particular managed object.)	247
<u>RTRProxyManagedObjectHandleIterator</u> (A RTRProxyManagedObjectHandleIterator is used to sequentially traverse a set of Proxy Managed Object Handles.)	249
<u>RTRProxyManagedObjectPool</u> (RTRProxyManagedObjectPool is a pool of objects matching the contents of a directory provided on the constructor. The directory in turn matches the contents of a pool of object servers.)	249
<u>RTRProxyManagedObjectPoolClient</u> (RTRProxyManagedObjectPoolClient is the abstract base class for application components which wish to register with one or more instances of <u>RTRProxyManagedObjectPool</u> in order to be notified when objects are added to or removed from a pool.)	250
<u>RTRProxyManagedObjectServer</u> (Provides access to the managed objects of a particular RTRManagedObjectServer. The set of available root proxy managed objects is also maintained.)	251
<u>RTRProxyManagedObjectServerClient</u> (RTRProxyManagedObjectServerClient is the abstract base class for application components which wish to register with one or more instances of <u>RTRProxyManagedObjectServer</u> in order to be notified when root managed objects are added to or removed from a server.)	253
<u>RTRProxyManagedObjectServerPool</u> (RTRProxyManagedObjectServerPool acts as a factory for instances of the class <u>RTRProxyManagedObjectServer</u>)	253
<u>RTRProxyManagedObjectServerPoolClient</u> (The base class for components which can register with a proxy managed object server pool to receive change events from that server pool. The notifications are grouped into a single category. (1) A proxy managed object server has been added/removed from the pool.)	255
<u>RTRProxyManagedString</u> (A cloned (proxy) representation of a String variable. The base class for proxy string managed variables. Inherits from <u>RTRProxyManagedVariable</u> and provides additional services for accessing and modifying (conditionally) the current value.)	256
<u>RTRProxyManagedStringConfig</u> (A cloned (proxy) representation of a StringConfig variable. The base class for proxy string managed configuration variables. Inherits from <u>RTRProxyManagedString</u> and provides additional services for accessing the stored and default values.)	257
<u>RTRProxyManagedVarHandleIterator</u> (A RTRProxyManagedVarHandleIterator is used to sequentially traverse a set of Proxy Managed Variable Handles.)	259

<u>RTRProxyManagedVariable</u> (A cloned (proxy) representation of an <u>RTRManagedVariable</u> . The base class for all of the proxy managed variable types. The cloning process could be an asynchronous process and so the state of the proxy variable must be checked before using many of the available operations (methods).)	260
<u>RTRProxyManagedVariableClient</u> (The base class for components which can register with a variable to receive change events from that variable. The notifications are grouped into three categories: (1) proxy variable state changes, (2) the variable has been updated, and (3) the variable has been deleted by the managed application.)	262
<u>RTRProxyManagedVariableHandle</u> (A <u>RTRProxyManagedVariableHandle</u> uniquely identifies a Proxy Managed Variable. The handle is used to request a clone (proxy) of a particular managed variable.)	263
<u>RTRPublicBoolean</u> (An implementation of the <u>RTRManagedBoolean</u> base class which provides modification operations and uses the global instance of <u>RTRMOServerMemPool</u> for storage allocation.)	265
<u>RTRPublicBooleanConfig</u> (An implementation of the <u>RTRManagedBooleanConfig</u> base class which provides modification operations and uses the global instance of <u>RTRMOServerMemPool</u> for storage allocation.)	267
<u>RTRPublicCounter</u> (An implementation of the <u>ManagedCounter</u> base class which provides increment capability and uses the class <u>RTRMNumericImpl</u> for storage allocation. Note: counters can be reset to 0 and incremented, but not decremented)	269
<u>RTRPublicGauge</u> (An implementation of the <u>RTRManagedGauge</u> base class which provides modification operations and uses the global instance of <u>RTRMOServerMemPool</u> for storage allocation.)	271
<u>RTRPublicGaugeConfig</u> (An implementation of the <u>RTRManagedGaugeConfig</u> base class which provides modification operations and uses the global instance of <u>RTRMOServerMemPool</u> for storage allocation.)	273
<u>RTRPublicLargeNumeric</u> (An implementation of the <u>RTRManagedLargeNumeric</u> base class which provides modification operations and uses the class <u>RTRMNumericImpl</u> for storage allocation.)	278
<u>RTRPublicNumeric</u> (An implementation of the <u>RTRManagedNumeric</u> base class which provides modification operations and uses the class <u>RTRMNumericImpl</u> for storage allocation.)	278
<u>RTRPublicNumericConfig</u> (An implementation of the <u>RTRManagedNumericConfig</u> base class which uses the class <u>RTRMNumConfigImpl</u> for storage allocation.)	280
<u>RTRPublicNumericRange</u> (An implementation of the <u>RTRManagedNumericRange</u> base class which uses the class <u>RTRMNumRangeImpl</u> for storage allocation.)	282
<u>RTRPublicObject</u> (Typically, application components which wish to be managed or become "public" are descendants of <u>RTRPublicObject</u> . They in turn may instantiate other public objects which will be their children in the managed object tree.)	285
<u>RTRPublicObjectLock</u> (A construct that is convenient in a multi-thread application where synchronization is needed for accessing managed object directory (MOD) and parent managed object. For example, when constructing/desconstructing instances of <u>RTRPublicObject</u> in multiple threads, instance of this can be constructed on stack to lock the global object tree, and when this instance is out of scope, its destructor is called to unlock the global object tree.)	288
<u>RTRPublicString</u> (An implementation of the <u>RTRManagedString</u> base class which provides set operations and uses the class <u>RTRMStringImpl</u> for storage allocation.)	288
<u>RTRPublicStringConfig</u> (An implementation of the <u>RTRManagedStringConfig</u> base class which uses the class <u>RTRMStrConfigImpl</u> for storage allocation.)	291
<u>RTRSelectNotifier</u> (This implementation of <u>RTREventNotifierImp</u> implements a main loop based on the select() system call.)	293
<u>RTRServerSharedMemoryRoot</u> (The encapsulation of the server side of a server/client shared memory relationship. An instance of <u>RTRServerSharedMemoryRoot</u> is constructed with a key and will then attempt to allocate the shared memory using with that key. If memory already exists with that key then the memory server will examine that memory to determine whether or not it can safely be reinitialized. If the memory header matches that which the server would create (version, size etc) and the memory appears to no longer be in use, then the server will re-initialize the existing memory. If the memory could be used but has not yet timed-out (based on data extracting from the existing memory) then the server will periodically retry the allocation process.)	295
<u>RTRSharedMemoryStats</u>	297
<u>RTRShmMOServer</u> (<u>RTRShmMOServer</u> is a helper class which instantiates an instance of <u>RTRServerSharedMemoryRoot</u> (and conditionally <u>RTRSharedMemoryStats</u>) as indicated by either the config db or information passed in on the constructor.)	299
<u>RTRShmMOServerMemPool</u> (<u>RTRShmMOServerMemPool</u> is an implementation of the abstract base class <u>RTRMOServerMemPool</u> which uses shared memory to allocate storage for managed objects and variables allocated by the application(server).)	301

<u>RTRShmProxyManagedObjectClassDirFactory</u> (A utility class used to obtain instances of <u>RTRProxyManagedObjectClassDirectory</u>)	303
<u>RTRShmProxyManagedObjectServerPool</u> (A shared memory based implementation of a <u>RTRProxyManagedObjectServerPool</u>)	305
<u>RTRShmServer</u> (RTRShmServer is a helper class which instantiates an <u>RTRServerSharedMemoryRoot</u> . The configuration of the <u>RTRServerSharedMemoryRoot</u> is obtained from either a config file or passed in from the constructor (with a minimal number of required arguments).)	306
<u>RTRString</u> (A representation for a sequence of characters. The sequence may contain embedded null characters.)	308
<u>RTRTimerCmd</u> (RTRTimerCmd is an abstract base class for components that will receive timer events.)	312
<u>RTRWindowsNotifier</u> (This implementation of <u>RTREventNotifierImp</u> (<u>RTREventNotifier</u>) is based on the Windows library. The implementation allocates a window (WNDCLASS) which is used to register for I/O and timing events as needed.) ..	314
<u>RTRXEventNotifier</u>	316
<u>RTRXFileDb</u> (This descendant of a RTRFileConfigDb implements an "X" version of a file based configuration database. The X11 library configuration utilites are used to parse and maintain config variables retrieved from a disk file.)	318
<u>RTRXtNotifier</u> (An implementation of an <u>RTREventNotifierImp</u> (<u>RTREventNotifier</u>) based on the Xt library. The application must initialize the static class member appContext. It is of type XtAppContext. The initialition must occur before any methods of the notifier are invoked by any part of the system.)	319
<u>RTRXViewNotifier</u> (An implementation of an <u>RTREventNotifierImp</u> (<u>RTREventNotifier</u>) based on the XView library. The application must initialize the static class member appContext. It is of type Frame. The initialition must occur before any methods of the notifier are invoked by any part of the system.)	322

Chapter 4 Refinitiv Management Classes Module Documentation

RTRConfigDb - State

Functions

- virtual RTRBOOL [RTRConfigDb::error](#) () const =0
- virtual const char * [RTRConfigDb::errorText](#) () const =0

Function Documentation

virtual RTRBOOL RTRConfigDb::error () const [pure virtual, inherited]
Is the config db in an error state?

Implemented in [RTRXFileDb](#). **virtual const char* RTRConfigDb::errorText () const** [pure virtual, inherited]
Explanation for the error.

Implemented in [RTRXFileDb](#).

RTRConfigDb - Query

Functions

- virtual RTRBOOL [RTRConfigDb::has](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &varName) const =0

Function Documentation

virtual RTRBOOL RTRConfigDb::has (const RTRObjectId & classId, const RTRObjectId & instanceId, const [RTRString](#) & varName) const [pure virtual, inherited]
Does db contain a variable corresponding to the class identifier and instance identifier with the given variable name?

REQUIRE: !error()

Implemented in [RTRXFileDb](#).

RTRConfigDb - Access

Functions

- virtual [RTRConfigVariable](#) [RTRConfigDb::variable](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &varName, const [RTRString](#) &dflt) const =0
- virtual [RTRConfigVariable](#) [RTRConfigDb::variable](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &varName) const =0

Function Documentation

virtual [RTRConfigVariable](#) RTRConfigDb::variable (const [RTRObjectId](#) & classId, const [RTRObjectId](#) & instanceId, const [RTRString](#) & varName, const [RTRString](#) & dflt) const [pure virtual, inherited]

The variable corresponding to the given class identifier and instance identifier along with the given variable name. If no value is available, the returned config variable will use the default value.

REQUIRE: !error()

ENSURE: !Result.[error\(\)](#)

Implemented in [RTRXFileDb](#).virtual [RTRConfigVariable](#) RTRConfigDb::variable (const [RTRObjectld](#) & classld, const [RTRObjectld](#) & instancelld, const [RTRString](#) & varName) const [pure virtual, inherited]

The variable corresponding to the given class identifier and instance identifier along with the given variable name. If no value is available, the returned config variable will have its error set.

REQUIRE: !error()

ENSURE: has(classld, instancelld, varName) == !Result.[error\(\)](#)

Implemented in [RTRXFileDb](#).

RTRConfigDb - OBSOLETE

Functions

- virtual [RTRConfigVariable](#) value (const [RTRObjectld](#) &classld, const [RTRObjectld](#) &instancelld, const [RTRString](#) &varName, const [RTRString](#) &dflt) const =0
- virtual [RTRConfigVariable](#) value (const [RTRObjectld](#) &classld, const [RTRObjectld](#) &instancelld, const [RTRString](#) &varName) const =0

RTRConfigVariable - Attributes

Functions

- const [RTRString](#) & [RTRConfigVariable::defaultValue](#) () const
- RTRBOOL [RTRConfigVariable::isDefaultValue](#) () const

Function Documentation

const [RTRString](#) & RTRConfigVariable::defaultValue () const [inline, inherited]
Default value.

RTRBOOL RTRConfigVariable::isDefaultValue () const [inline, inherited]
Was this variable assigned a default value?

RTRConfigVariable - State

Functions

- RTRBOOL [RTRConfigVariable::error](#) () const

Function Documentation

RTRBOOL RTRConfigVariable::error () const [inline, inherited]
Does the config variable exist? Note that a TRUE response from this function indicates non-existence, it does *not* indicate an empty/null value.

RTRConfigVariable - Assignment

Functions

- [RTRConfigVariable](#) & [RTRConfigVariable::operator=](#) (const [RTRConfigVariable](#) &other)

Function Documentation

[RTRConfigVariable](#) & RTRConfigVariable::operator= (const [RTRConfigVariable](#) & *other*) [inherited]

Initialize this variable as a copy of other

RTRConfigVariable - Compatibility

Functions

- [RTRExternalValue](#) [RTRConfigVariable::value](#) ()

Function Documentation

[RTRExternalValue](#) [RTRConfigVariable::value](#) () [inherited]

The value of this configuration variable. This may be an empty value.

REQUIRE: !error()

RTRCmdLineArg - Attributes

Functions

- Required [RTRCmdLineArg::required](#) () const
- RTRBOOL [RTRCmdLineArg::hidden](#) () const
- const [RTRString](#) & [RTRCmdLineArg::tag](#) () const
- const [RTRString](#) & [RTRCmdLineArg::name](#) () const
- const [RTRString](#) & [RTRCmdLineArg::defaultValue](#) () const
- const [RTRString](#) & [RTRCmdLineArg::purpose](#) () const

Function Documentation

[RTRCmdLineArg::Required](#) [RTRCmdLineArg::required](#) () const [inline, inherited]

Is the argument required to be on the command line?

RTRBOOL [RTRCmdLineArg::hidden](#) () const [inline, inherited]

Is the argument hidden (not visible in usage text)?

const [RTRString](#) & [RTRCmdLineArg::tag](#) () const [inline, inherited]

The command line tag. Standard string comparisons are used to compare elements of argv with [tag\(\)](#).

e.g. the "config" part of "-config file_name"

const [RTRString](#) & [RTRCmdLineArg::name](#) () const [inline, inherited]

The name, if any, of the value to be specified. If [name\(\)](#) is null then this arg will be interpreted as a boolean flag.

e.g. the "file_name" part of "-config file_name"

const [RTRString](#) & [RTRCmdLineArg::defaultValue](#) () const [inline, inherited]

The default, if any, to be used.

const [RTRString](#) & [RTRCmdLineArg::purpose](#) () const [inline, inherited]

The description to be used in the long part of the usage text.

e.g. The "load ..." part of "Usage: foo -config file_name -config load the given file name"

RTRCmdLineArg - State

Functions

- RTRBOOL **valid** () const
- RTRBOOL **error** () const

RTRCmdLineArg - Transformation

Functions

- const [RTRString](#) & [RTRCmdLineArg::stringValue](#) () const

Function Documentation

const [RTRString](#) & RTRCmdLineArg::stringValue () const [inline, inherited]

REQUIRE: valid()

RTRCmdLineArg - Query

Functions

- RTRBOOL **hasDefault** () const

RTRCmdLineArg - Operations

Functions

- virtual void **printShortUsage** (std::ostream &) const
- virtual void **printLongUsage** (std::ostream &) const
- virtual void **printLongUsage** (std::ostream &, int) const
- virtual void **resolve** (RTRDLinkList< RTRCmdLineData, RTRDLink0 > &)
- virtual void [RTRCmdLineArg::hide](#) ()

Function Documentation

virtual void RTRCmdLineArg::hide () [virtual, inherited]

Make argument hidden (it won't show up in usage text).

RTRDefaultLogger - Attributes

Functions

- RTRDefaultFileAction * [RTRDefaultLogger::defaultFileAction](#) ()
- RTRDefaultStdErrAction * [RTRDefaultLogger::defaultStdErrorAction](#) ()
- RTRDefaultSystemAction * [RTRDefaultLogger::defaultSystemAction](#) ()

Function Documentation

RTRDefaultFileAction* RTRDefaultLogger::defaultFileAction () [inherited]

The file action used by the logger.

Creation of this action can be determined by configuration.

The default is to create this action.

RTRDefaultStdErrAction* RTRDefaultLogger::defaultStdErrorAction () [inherited]

The standard error output action used by the logger.

Creation of this action can be determined by configuration.

The default is to *not* create this action.

RTRDefaultSystemAction* RTRDefaultLogger::defaultSystemAction () [inherited]

The system logger action used by the logger.

Creation of this action can be determined by configuration.

The default is to *not* create this action.

RTRDefaultLogger - Utilities

Functions

- void [RTRDefaultLogger::displayConfiguration](#) () const
- void [RTRDefaultLogger::setSelector](#) ([RTRString](#) &name)

Function Documentation

void RTRDefaultLogger::displayConfiguration () const [inherited]

displays logger on stdout

void RTRDefaultLogger::setSelector ([RTRString](#) & name) [inherited]

sets the logger selector

RTREventNotifierImp - Access

Functions

- virtual [RTRIOClient](#)* [RTREventNotifierImp::registeredReadClient](#) (int fd) const
- virtual [RTRIOClient](#)* [RTREventNotifierImp::registeredWriteClient](#) (int fd) const
- virtual [RTRIOClient](#)* [RTREventNotifierImp::registeredExceptionClient](#) (int fd) const

Function Documentation

virtual [RTRIOClient](#)* RTREventNotifierImp::registeredReadClient (int fd) const [virtual, inherited]

The client registered for read events on file descriptor fd? MT-unsafe interface

Implements [RTREventNotifier](#). **virtual [RTRIOClient](#)* RTREventNotifierImp::registeredWriteClient (int fd) const** [virtual, inherited]

The client registered for write events on file descriptor fd? MT-unsafe interface

Implements [RTREventNotifier](#). **virtual [RTRIOClient](#)* RTREventNotifierImp::registeredExceptionClient (int fd) const** [virtual, inherited]

The client registered for exception events on file descriptor fd? MT-unsafe interface

Implements [RTREventNotifier](#).

RTEventNotifierImp - Insert

Functions

- virtual void [RTEventNotifierImp::addReadClient](#) (RTRIOClient &client, int fd)
- virtual void [RTEventNotifierImp::addWriteClient](#) (RTRIOClient &client, int fd)
- virtual void [RTEventNotifierImp::addExceptionClient](#) (RTRIOClient &client, int fd)

Function Documentation

virtual void RTEventNotifierImp::addReadClient (RTRIOClient & *client*, int *fd*) [virtual, inherited]

Register the given client for read events on file descriptor fd. Synchronized

REQUIRE : !hasReadClient(fd)

ENSURE : hasReadClient(fd)

ENSURE : registeredReadClient(fd) == &client

Implements [RTEventNotifier](#). **virtual void RTEventNotifierImp::addWriteClient (RTRIOClient & *client*, int *fd*)** [virtual, inherited]

Register the given client for write events on file descriptor fd. Synchronized

REQUIRE : !hasWriteClient(fd)

ENSURE : hasWriteClient(fd)

ENSURE : registeredWriteClient(fd) == &client

Implements [RTEventNotifier](#). **virtual void RTEventNotifierImp::addExceptionClient (RTRIOClient & *client*, int *fd*)** [virtual, inherited]

Register the given client for exception events on file descriptor fd. Synchronized

REQUIRE : !hasExceptionClient(fd)

ENSURE : hasExceptionClient(fd)

ENSURE : registeredExceptionClient(fd) == &client

Implements [RTEventNotifier](#).

RTEventNotifierImp - Remove

Functions

- virtual void [RTEventNotifierImp::dropReadClient](#) (int fd)
- virtual void [RTEventNotifierImp::dropWriteClient](#) (int fd)
- virtual void [RTEventNotifierImp::dropExceptionClient](#) (int fd)

Function Documentation

virtual void RTEventNotifierImp::dropReadClient (int *fd*) [virtual, inherited]

De-register the client current registered for read events on file descriptor fd.

Synchronized

ENSURE : !hasReadClient(fd)

Implements [RTEventNotifier](#). **virtual void RTEventNotifierImp::dropWriteClient (int *fd*)** [virtual, inherited]

De-register the client current registered for write events on file descriptor fd.

Synchronized

ENSURE : !hasWriteClient(fd)

Implements [RTREventNotifier](#). **virtual void RTREventNotifierImp::dropExceptionClient (int fd) [virtual, inherited]**

De-register the client current registered for exception events on file descriptor fd.

Synchronized

ENSURE : !hasExceptionClient(fd)

Implements [RTREventNotifier](#).

RTREventNotifierImp - Implementation

Functions

- void [RTREventNotifierImp::notifyReadPending](#) (int fd)
- void [RTREventNotifierImp::notifyWritePending](#) (int fd)
- void [RTREventNotifierImp::notifyExceptPending](#) (int fd)
- void [RTREventNotifierImp::expireEvents](#) ()

Function Documentation

void RTREventNotifierImp::notifyReadPending (int fd) [inherited]

Called by descendants when an I/O read event is pending on the given file descriptor. This method will invoke the appropriate method of the RTRIOClient instance which is registered for the given descriptor.

void RTREventNotifierImp::notifyWritePending (int fd) [inherited]

Called by descendants when an I/O write event is pending on the given file descriptor. This method will invoke the appropriate method of the RTRIOClient instance which is registered for the given descriptor.

void RTREventNotifierImp::notifyExceptPending (int fd) [inherited]

Called by descendants when an I/O exception event is pending on the given file descriptor. This method will invoke the appropriate method of the RTRIOClient instance which is registered for the given descriptor.

void RTREventNotifierImp::expireEvents () [inherited]

Called by descendants when the current timer has expired.

RTREventNotifier - Query

Functions

- RTRBOOL [RTREventNotifier::isReadClient](#) (RTRIOClient &client, int fd) const
- RTRBOOL [RTREventNotifier::isWriteClient](#) (RTRIOClient &client, int fd) const
- RTRBOOL [RTREventNotifier::isExceptionClient](#) (RTRIOClient &client, int fd) const
- RTRBOOL [RTREventNotifier::hasReadClient](#) (int fd) const
- RTRBOOL [RTREventNotifier::hasWriteClient](#) (int fd) const
- RTRBOOL [RTREventNotifier::hasExceptionClient](#) (int fd) const

Function Documentation

RTRBOOL RTREventNotifier::isReadClient (RTRIOClient & *client*, int *fd*) const [inline, inherited]

Is the given client registered for read events on file descriptor fd?

ENSURE: Result == (registeredClient(fd) == &client)

RTRBOOL RTREventNotifier::isWriteClient (RTRIOClient & *client*, int *fd*) const [inline, inherited]

Is the given client registered for write events on file descriptor fd?

ENSURE: Result == (registeredWriteClient(fd) == &client)

RTRBOOL RTREventNotifier::isExceptionClient (RTRIOClient & *client*, int *fd*) const [inline, inherited]

Is the given client registered for exception events on file descriptor fd?

ENSURE: Result == (registeredExceptionClient(fd) == &client)

RTRBOOL RTREventNotifier::hasReadClient (int *fd*) const [inline, inherited]

Is any client registered for read events on file descriptor fd?

RTRBOOL RTREventNotifier::hasWriteClient (int *fd*) const [inline, inherited]

Is any client registered for write events on file descriptor fd?

RTRBOOL RTREventNotifier::hasExceptionClient (int *fd*) const [inline, inherited]

Is any client registered for exception events on file descriptor fd?

RTREventNotifier - Access

Functions

- virtual RTRIOClient * [RTREventNotifier::registeredReadClient](#) (int fd) const =0
- virtual RTRIOClient * [RTREventNotifier::registeredWriteClient](#) (int fd) const =0
- virtual RTRIOClient * [RTREventNotifier::registeredExceptionClient](#) (int fd) const =0

Function Documentation

virtual RTRIOClient* RTREventNotifier::registeredReadClient (int *fd*) const [pure virtual, inherited]

The client registered for read events on file descriptor fd?

Implemented in [RTREventNotifierImp](#). **virtual RTRIOClient* RTREventNotifier::registeredWriteClient (int *fd*) const** [pure virtual, inherited]

The client registered for write events on file descriptor fd?

Implemented in [RTREventNotifierImp](#). **virtual RTRIOClient* RTREventNotifier::registeredExceptionClient (int *fd*) const** [pure virtual, inherited]

The client registered for exception events on file descriptor fd?

Implemented in [RTREventNotifierImp](#).

RTREventNotifier - Insert

Functions

- virtual void [RTREventNotifier::addReadClient](#) (RTRIOClient &client, int fd)=0
- virtual void [RTREventNotifier::addWriteClient](#) (RTRIOClient &client, int fd)=0
- virtual void [RTREventNotifier::addExceptionClient](#) (RTRIOClient &client, int fd)=0

Function Documentation

virtual void RTREventNotifier::addReadClient (RTRIOClient & *client*, int *fd*) [pure virtual, inherited]

Register the given client for read events on file descriptor fd.

REQUIRE : !hasReadClient(fd)

ENSURE : hasReadClient(fd)

ENSURE : registeredReadClient(fd) == &client

Implemented in [RTREventNotifierImp](#). **virtual void RTREventNotifier::addWriteClient (RTRIOClient & *client*, int *fd*)** [pure virtual, inherited]

Register the given client for write events on file descriptor fd.

REQUIRE : !hasWriteClient(fd)

ENSURE : hasWriteClient(fd)

ENSURE : registeredWriteClient(fd) == &client

Implemented in [RTREventNotifierImp](#). **virtual void RTREventNotifier::addExceptionClient (RTRIOClient & *client*, int *fd*)** [pure virtual, inherited]

Register the given client for exception events on file descriptor fd.

REQUIRE : !hasExceptionClient(fd)

ENSURE : hasExceptionClient(fd)

ENSURE : registeredExceptionClient(fd) == &client

Implemented in [RTREventNotifierImp](#).

RTREventNotifier - Remove

Functions

- virtual void [RTREventNotifier::dropReadClient](#) (int fd)=0
- virtual void [RTREventNotifier::dropWriteClient](#) (int fd)=0
- virtual void [RTREventNotifier::dropExceptionClient](#) (int fd)=0

Function Documentation

virtual void RTREventNotifier::dropReadClient (int *fd*) [pure virtual, inherited]

De-register the client current registered for read events on file descriptor fd.

ENSURE : !hasReadClient(fd)

Implemented in [RTREventNotifierImp](#). **virtual void RTREventNotifier::dropWriteClient (int *fd*)** [pure virtual, inherited]

De-register the client current registered for write events on file descriptor fd.

ENSURE : !hasWriteClient(fd)

Implemented in [RTREventNotifierImp](#). **virtual void RTREventNotifier::dropExceptionClient (int *fd*)** [pure virtual, inherited]

De-register the client current registered for exception events on file descriptor fd.

ENSURE : !hasExceptionClient(fd)

Implemented in [RTREventNotifierImp](#).

RTREventNotifier - Control

Functions

- virtual void [RTREventNotifier::disable](#) ()=0

Function Documentation

virtual void RTREventNotifier::disable () [pure virtual, inherited]

Stop dispatching events. NOTE: This causes control to return to the context which "started" the notifier. Exact behaviour of this is implementation specific. This feature provided as a convenience for simple programs and for debugging. In general components should not "stop" an application. Descendant implementations should, in general, cease operation when there are not IO clients and no timers pending.

Implemented in [RTRSelectNotifier](#), [RTRWindowsNotifier](#), [RTRXtNotifier](#), and [RTRXViewNotifier](#).

RTREventNotifier - Control - OBSOLETE

Functions

- virtual void [RTREventNotifier::enable](#) ()=0

Function Documentation

virtual void RTREventNotifier::enable () [pure virtual, inherited]

Enable the notifier. Left here for compatibility. The "enable" of a notifier is implementation specific.

Implemented in [RTRSelectNotifier](#), [RTRWindowsNotifier](#), [RTRXtNotifier](#), and [RTRXViewNotifier](#).

RTRLockableObj - Opearations

Functions

- virtual void **lock** ()
- virtual void **unlock** ()

RTRLockableObj - State

Functions

- virtual RTRBOOL [RTRLockableObj::locked](#) () const

Function Documentation

virtual RTRBOOL RTRLockableObj::locked () const [virtual, inherited]

Is this locked by calling thread? It is used in PRECONDITION of application class to ensure that an instance of RTRLockableObj must be locked before access.

NOTE: Note: this only serves as necessary-but-not-sufficient condition;

i.e., ([locked\(\)](#)== RTRTRUE) == (possibly right);

([locked\(\)](#)== RTRFALSE) == (definitely wrong);

By default, this call will always return RTRTRUE unless static member alwaysLocked is set RTRFALSE.

Reimplemented in [RTRManagedObject](#), [RTRManagedVariable](#), [RTRProxyManagedObject](#), and [RTRProxyManagedVariable](#).

RTRManagedBooleanConfig - Assignment

Functions

- [RTRManagedBooleanConfig](#) & [RTRManagedBooleanConfig::operator=](#) (RTRBOOL rhs)

Function Documentation

[RTRManagedBooleanConfig](#) & RTRManagedBooleanConfig::operator= (RTRBOOL rhs) [inline, inherited]

Set the active value to rhs

REQUIRE: [modifyEnabled\(\)](#)

Reimplemented from [RTRManagedBoolean](#). Reimplemented in [RTRPublicBooleanConfig](#).

RTRManagedBooleanConfig - Attributes

Functions

- RTRBOOL [RTRManagedBooleanConfig::activeValue](#) () const
- RTRBOOL [RTRManagedBooleanConfig::storeValue](#) () const
- RTRBOOL [RTRManagedBooleanConfig::factoryDefault](#) () const

Function Documentation

RTRBOOL RTRManagedBooleanConfig::activeValue () const [inline, inherited]

A synonym for [value\(\)](#)

RTRBOOL RTRManagedBooleanConfig::storeValue () const [inline, inherited]

The store value

RTRBOOL RTRManagedBooleanConfig::factoryDefault () const [inline, inherited]

The factory default value

RTRManagedBoolean - Comparison

Functions

- RTRBOOL [RTRManagedBoolean::operator==](#) (RTRBOOL) const

Function Documentation

RTRBOOL RTRManagedBoolean::operator== (RTRBOOL) const [inline, inherited]
Comparison

RTRManagedBoolean - Access

Functions

- RTRBOOL [RTRManagedBoolean::value](#) () const

Function Documentation

RTRBOOL RTRManagedBoolean::value () const [inline, inherited]
The current value of this variable.

RTRManagedBoolean - Transformation

Functions

- virtual [RTRString RTRManagedBoolean::toString](#) () const

Function Documentation

virtual [RTRString](#) RTRManagedBoolean::toString () const [virtual, inherited]
A copy of this variable value, represented as a string

Implements [RTRManagedVariable](#).

RTRManagedBoolean - State

Functions

- RTRBOOL [RTRManagedBoolean::modifyEnabled](#) ()

Function Documentation

RTRBOOL RTRManagedBoolean::modifyEnabled () [inline, inherited]
Is the managing application permitted to modify this variable?

RTRManagedBoolean - Assignment

Functions

- [RTRManagedBoolean](#) & [RTRManagedBoolean::operator=](#) (RTRBOOL rhs)

Function Documentation

[RTRManagedBoolean](#) & RTRManagedBoolean::operator= (RTRBOOL rhs) [inline, inherited]
REQUIRE: [modifyEnabled\(\)](#)

Set the current value to rhs, notify clients and notify context (containing managed object).

Reimplemented in [RTRManagedBooleanConfig](#), [RTRPublicBooleanConfig](#), and [RTRPublicBoolean](#).

RTRManagedBoolean - Operations

Functions

- virtual void [RTRManagedBoolean::set](#) ()
- virtual void [RTRManagedBoolean::clear](#) ()

Function Documentation

virtual void [RTRManagedBoolean::set](#) () [virtual, inherited]

REQUIRE: [modifyEnabled\(\)](#)

Set the current value to RTRTRUE, notify clients and notify context (containing managed object).

virtual void [RTRManagedBoolean::clear](#) () [virtual, inherited]

REQUIRE: [modifyEnabled\(\)](#)

Set the current value to RTRFALSE, notify clients and notify context (containing managed object).

RTRManagedCounter - Comparison

Functions

- RTRBOOL [RTRManagedCounter::operator==](#) (unsigned long) const

Function Documentation

RTRBOOL [RTRManagedCounter::operator==](#) (unsigned *long*) const [inline, inherited]

Comparison

RTRManagedCounter - Access

Functions

- unsigned long [RTRManagedCounter::value](#) () const

Function Documentation

unsigned long [RTRManagedCounter::value](#) () const [inline, inherited]

The current value of this variable.

RTRManagedCounter - Transformation

Functions

- [RTRManagedCounter::operator unsigned long](#) () const
- virtual [RTRString RTRManagedCounter::toString](#) () const

Function Documentation

[RTRManagedCounter::operator unsigned long](#) () const [inline, inherited]

This variable as an unsigned long.

virtual [RTRString](#) RTRManagedCounter::toString () const [virtual, inherited]
 A copy of this variable's value, represented as a string.

Implements [RTRManagedVariable](#).

RTRManagedCounter - Operations

Functions

- virtual void [RTRManagedCounter::reset](#) ()=0

Function Documentation

virtual void RTRManagedCounter::reset () [pure virtual, inherited]
 Reset this counter to 0.

Implemented in [RTRPublicCounter](#).

RTRManagedGaugeConfig - Attributes

Functions

- long [RTRManagedGaugeConfig::minStoreValue](#) () const
- long [RTRManagedGaugeConfig::minFactoryDefault](#) () const
- long [RTRManagedGaugeConfig::maxStoreValue](#) () const
- long [RTRManagedGaugeConfig::maxFactoryDefault](#) () const

Function Documentation

long RTRManagedGaugeConfig::minStoreValue () const [inline, inherited]
 The store minimum value.

long RTRManagedGaugeConfig::minFactoryDefault () const [inline, inherited]
 The factory default minimum value

long RTRManagedGaugeConfig::maxStoreValue () const [inline, inherited]
 The store maximum value

long RTRManagedGaugeConfig::maxFactoryDefault () const [inline, inherited]
 The factory default maximum value

RTRMgmtAction - Identity

Functions

- const [RTRObjectId](#) & [RTRMgmtAction::classId](#) () const
- const [RTRObjectId](#) & [RTRMgmtAction::instanceId](#) () const

Function Documentation

const [RTRObjectId](#) & RTRMgmtAction::classId () const [inline, inherited]
 Type of this action.

`const RTRObjectid & RTRMgmtAction::instanceId () const [inline, inherited]`
 Identifier for this action.

RTRMgmtAction - Attributes

Functions

- `const RTRMgmtEventFilter & RTRMgmtAction::filter () const`

Function Documentation

`const RTRMgmtEventFilter & RTRMgmtAction::filter () const [inline, inherited]`
 The current filter.

RTRMgmtAction - State

Functions

- `RTRBOOL RTRMgmtAction::installed () const`

Function Documentation

`RTRBOOL RTRMgmtAction::installed () const [inline, inherited]`
 Is this action installed with its event router?

RTRMgmtAction - Operations

Functions

- `virtual void RTRMgmtAction::processMgmtEvent (const RTRMgmtEvent &)`
- `virtual void RTRMgmtAction::processFilteredMgmtEvent (const RTRMgmtEvent &)=0`
- `void RTRMgmtAction::install ()`
- `void RTRMgmtAction::deinstall ()`
- `void RTRMgmtAction::setFilter (const RTRMgmtEventFilter &)`

Function Documentation

`virtual void RTRMgmtAction::processMgmtEvent (const RTRMgmtEvent &) [virtual, inherited]`
 Filter the given event with [filter\(\)](#). If passed, invoke [processFilteredMgmtEvent\(\)](#)

`virtual void RTRMgmtAction::processFilteredMgmtEvent (const RTRMgmtEvent &) [pure virtual, inherited]`
 Take action on the given event.

`void RTRMgmtAction::install () [inherited]`
 Install this action its router.

REQUIRE: `!installed()`

ENSURE: [installed\(\)](#)

`void RTRMgmtAction::deinstall () [inherited]`
 Deinstall this action.

REQUIRE: [installed\(\)](#)

ENSURE: !installed()

void RTRMgmtAction::setFilter (const RTRMgmtEventFilter &) [inherited]
Use the given filter (copied).

RTRMgmtEvent - Attributes

Functions

- const [RTRObjectId](#) & [RTRMgmtEvent::instanceId](#) () const
- const [RTRString](#) & [RTRMgmtEvent::text](#) () const
- int [RTRMgmtEvent::severity](#) () const
- const RTRDateTime & [timestamp](#) () const

Function Documentation

const [RTRObjectId](#) & RTRMgmtEvent::instanceId () const [inline, inherited]
The unique instance identifier for this event.

const [RTRString](#) & RTRMgmtEvent::text () const [inline, inherited]
The descriptive text for this event.

int RTRMgmtEvent::severity () const [inline, inherited]
Severity of the event. Valid values are:
Emergency, Alert, Critical, Error, Warning, Notice, Info, Debug, or None

RTRMgmtEvent - Comparison

Functions

- RTRBOOL [RTRMgmtEvent::operator<](#) (const [RTRMgmtEvent](#) &) const
- RTRBOOL [RTRMgmtEvent::operator>](#) (const [RTRMgmtEvent](#) &) const
- RTRBOOL [RTRMgmtEvent::operator==](#) (const [RTRMgmtEvent](#) &) const
- RTRBOOL [RTRMgmtEvent::operator<=](#) (const [RTRMgmtEvent](#) &) const
- RTRBOOL [RTRMgmtEvent::operator>=](#) (const [RTRMgmtEvent](#) &) const

Function Documentation

RTRBOOL RTRMgmtEvent::operator< (const [RTRMgmtEvent](#) &) const [inline, inherited]
Time based (newer is greater than older).

RTRBOOL RTRMgmtEvent::operator> (const [RTRMgmtEvent](#) &) const [inline, inherited]
Time based (newer is greater than older).

RTRBOOL RTRMgmtEvent::operator== (const [RTRMgmtEvent](#) &) const [inline, inherited]
Time based (newer is greater than older).

RTRBOOL RTRMgmtEvent::operator<= (const [RTRMgmtEvent](#) &) const [inline, inherited]
Time based (newer is greater than older).

RTRBOOL RTRMgmtEvent::operator>= (const [RTRMgmtEvent](#) &) const [inline, inherited]

Time based (newer is greater than older).

RTRMgmtEvent - Operation

Functions

- void [RTRMgmtEvent::setIdentifier](#) (const [RTRObjectld](#) &)
- void [RTRMgmtEvent::setSeverity](#) (int)
- void [RTRMgmtEvent::setSeverity](#) (const [RTRString](#) &)
- void [RTRMgmtEvent::setText](#) (const char *)
- void [RTRMgmtEvent::setText](#) (const [RTRString](#) &)
- void [RTRMgmtEvent::setTimestamp](#) ()
- void [RTRMgmtEvent::log](#) ()

Function Documentation

void RTRMgmtEvent::setIdentifier (const [RTRObjectld](#) &) [inherited]

Set the unique identifier for this event.

void RTRMgmtEvent::setSeverity (int) [inherited]

Set the severity level for this event by integer. Valid values must be in the range from RTRMgmtEvent::Emergency to RTRMgmtEvent::Debug.

ENSURE: [severity\(\)](#) <= Emergency && [severity\(\)](#) >= Debug

void RTRMgmtEvent::setSeverity (const [RTRString](#) &) [inherited]

Set the severity level for this event by string. Valid values must be one of the following: "Emergency", "Alert", "Critical", "Error", "Warning", "Notice", "Info" (default), "Debug".

ENSURE: [severity\(\)](#) <= Emergency && [severity\(\)](#) >= None

void RTRMgmtEvent::setText (const char *) [inherited]

Set the descriptive text for this event.

void RTRMgmtEvent::setText (const [RTRString](#) &) [inherited]

Set the descriptive text for this event.

void RTRMgmtEvent::setTimestamp () [inherited]

Set the time at which the event occurred.

void RTRMgmtEvent::log () [inherited]

Pass this event to the event routing mechanism.

RTRMgmtEvent - Operations Assignment

Functions

- [RTRMgmtEvent](#) & operator= (const [RTRMgmtEvent](#) &)

RTRMgmtEvent - OBSOLETE

Functions

- void [RTRMgmtEvent::setComponent](#) (const char *c)
- void [RTRMgmtEvent::setComponent](#) (const [RTRString](#) &c)

Function Documentation

void [RTRMgmtEvent::setComponent](#) (const char * c) [inherited]
Use [setIdentifier\(\)](#) instead.

void [RTRMgmtEvent::setComponent](#) (const [RTRString](#) &c) [inherited]
Use [setIdentifier\(\)](#) instead.

RTRManagedGauge - Attributes

Functions

- long [RTRManagedGauge::minValue](#) () const
- long [RTRManagedGauge::maxValue](#) () const
- long [RTRManagedGauge::lowWaterMark](#) () const
- long [RTRManagedGauge::highWaterMark](#) () const
- RTRBOOL [RTRManagedGauge::modifyEnabled](#) () const

Function Documentation

long [RTRManagedGauge::minValue](#) () const [inline, inherited]
The minimum value which may be assigned to this parameter.

ENSURE: [minValue\(\)](#) <= [maxValue\(\)](#)

ENSURE: ([minValue\(\)](#) <= [value\(\)](#)) || [modifyEnabled\(\)](#)

ENSURE: ([minValue\(\)](#) <= [highWaterMark\(\)](#)) || [modifyEnabled\(\)](#)

long [RTRManagedGauge::maxValue](#) () const [inline, inherited]
The maximum value which may be assigned to this parameter.

ENSURE: [maxValue\(\)](#) >= [minValue\(\)](#)

ENSURE: ([maxValue\(\)](#) >= [value\(\)](#)) || [modifyEnabled\(\)](#)

ENSURE: ([maxValue\(\)](#) >= [lowWaterMark\(\)](#)) || [modifyEnabled\(\)](#)

long [RTRManagedGauge::lowWaterMark](#) () const [inline, inherited]
The lowest value assumed by this gauge since its creation.

ENSURE: [lowWaterMark\(\)](#) <= [highWaterMark\(\)](#)

ENSURE: [lowWaterMark\(\)](#) <= [value\(\)](#)

ENSURE: ([lowWaterMark\(\)](#) <= [maxValue\(\)](#)) || [modifyEnabled\(\)](#)

long [RTRManagedGauge::highWaterMark](#) () const [inline, inherited]
The highest value assumed by this gauge since its creation.

ENSURE: [highWaterMark\(\)](#) >= [lowWaterMark\(\)](#)

ENSURE: [highWaterMark\(\)](#) >= [value\(\)](#)

ENSURE: ([highWaterMark\(\)](#) >= [minValue\(\)](#)) || [modifyEnabled\(\)](#)

RTRBOOL RTRManagedGauge::modifyEnabled () const [inline, inherited]

Is the consumer permitted to modify this variable?

RTRManagedGauge - Operations

Functions

- virtual void [RTRManagedGauge::setRange](#) (long newMin, long newMax)

Function Documentation

virtual void RTRManagedGauge::setRange (long newMin, long newMax) [virtual, inherited]

Sets the min and max values.

REQUIRE: [modifyEnabled\(\)](#)

REQUIRE: (newMin <= newValue) || [modifyEnabled\(\)](#)

REQUIRE: (newMax >= newValue) || [modifyEnabled\(\)](#)

REQUIRE: newMin <= newMax

RTRManagedNumericConfig - Attributes

Functions

- long [RTRManagedNumericConfig::minValue](#) () const
- long [RTRManagedNumericConfig::maxValue](#) () const
- long [RTRManagedNumericConfig::activeValue](#) () const
- long [RTRManagedNumericConfig::storeValue](#) () const
- int [RTRManagedNumericConfig::storeState](#) () const
- long [RTRManagedNumericConfig::factoryDefault](#) () const
- RTRBOOL [RTRManagedNumericConfig::modifyEnabled](#) () const
- RTRBOOL [RTRManagedNumericConfig::hasStore](#) () const
- RTRBOOL [RTRManagedNumericConfig::isStoreActive](#) () const
- RTRBOOL [RTRManagedNumericConfig::isStoreClassConfig](#) () const
- RTRBOOL [RTRManagedNumericConfig::isStoreInstanceConfig](#) () const

Function Documentation

long RTRManagedNumericConfig::minValue () const [inline, inherited]

The minimum value which may be assigned to the active variable.

ENSURE: ([minValue\(\)](#) <= [maxValue\(\)](#))

ENSURE: ([minValue\(\)](#) <= [value\(\)](#))

long RTRManagedNumericConfig::maxValue () const [inline, inherited]

The maximum value which may be assigned to the active variable.

ENSURE: ([maxValue\(\)](#) >= [minValue\(\)](#))

ENSURE: ([maxValue\(\)](#) >= [value\(\)](#))

long RTRManagedNumericConfig::activeValue () const [inline, inherited]

A synonym for [value\(\)](#)

ENSURE: ([activeValue\(\)](#) >= [minValue\(\)](#))

ENSURE: ([activeValue\(\)](#) <= [maxValue\(\)](#))

long RTRManagedNumericConfig::storeValue () const [inline, inherited]

The store value

int RTRManagedNumericConfig::storeState () const [inline, inherited]

The store state

long RTRManagedNumericConfig::factoryDefault () const [inline, inherited]

The factory default value

RTRBOOL RTRManagedNumericConfig::modifyEnabled () const [inline, inherited]

Is the consumer permitted to modify this variable?

RTRBOOL RTRManagedNumericConfig::hasStore () const [inherited]

Is the variable a client of a RTRVariableConfig?

RTRBOOL RTRManagedNumericConfig::isStoreActive () const [inherited]

Is the RTRVariableConfig in an active state?

REQUIRE: [hasStore\(\)](#)

RTRBOOL RTRManagedNumericConfig::isStoreClassConfig () const [inherited]

Is the RTRVariableConfig's context a RTRClassConfig? class config?

REQUIRE: [hasStore\(\)](#)

RTRBOOL RTRManagedNumericConfig::isStoreInstanceConfig () const [inherited]

Is the RTRVariableConfig's context a RTRInstanceConfig? instance config?

REQUIRE: [hasStore\(\)](#)

RTRManagedNumericConfig - Assignment

Functions

- [RTRManagedNumericConfig](#) & [RTRManagedNumericConfig::operator=](#) (long rhs)
- virtual void [RTRManagedNumericConfig::set](#) (long newValue)

Function Documentation

[RTRManagedNumericConfig](#) & [RTRManagedNumericConfig::operator=](#) (long rhs) [inline, inherited]

REQUIRE: [modifyEnabled\(\)](#)

REQUIRE: rhs >= [minValue\(\)](#)

REQUIRE: rhs <= [maxValue\(\)](#)

Set the active value.

Synchronized

Reimplemented in [RTRPublicNumericConfig](#). virtual void [RTRManagedNumericConfig::set](#) (long *newValue*) [virtual, inherited]

REQUIRE: [modifyEnabled\(\)](#)

REQUIRE: `newValue >= minValue\(\)`

REQUIRE: `newValue <= maxValue\(\)`

A synonym for [operator=\(\)](#)

Synchronized

RTRManagedNumericRange - Attributes

Functions

- long [RTRManagedNumericRange::minValue](#) () const
- long [RTRManagedNumericRange::maxValue](#) () const

Function Documentation

long [RTRManagedNumericRange::minValue](#) () const [inline, inherited]

The minimum value which may be assigned to this parameter.

ENSURE: ([minValue\(\)](#) <= [maxValue\(\)](#))

ENSURE: ([minValue\(\)](#) <= [value\(\)](#))

long [RTRManagedNumericRange::maxValue](#) () const [inline, inherited]

The maximum value which may be assigned to this parameter.

ENSURE: ([maxValue\(\)](#) >= [minValue\(\)](#))

ENSURE: ([maxValue\(\)](#) >= [value\(\)](#))

RTRManagedNumericRange - Assignment

Functions

- [RTRManagedNumericRange](#) & [RTRManagedNumericRange::operator=](#) (long rhs)

Function Documentation

[RTRManagedNumericRange](#) & [RTRManagedNumericRange::operator=](#) (long *rhs*) [inline, inherited]

EQUIRE: (rhs >= [minValue\(\)](#))

REQUIRE: (rhs <= [maxValue\(\)](#))

Reimplemented in [RTRPublicNumericRange](#).

RTRManagedNumericRange - Operations

Functions

- virtual void [RTRManagedNumericRange::set](#) (long newValue)

Function Documentation

virtual void [RTRManagedNumericRange::set](#) (long *newValue*) [virtual, inherited]

A synonym for [operator=\(\)](#)

REQUIRE: (newValue >= [minValue\(\)](#))

REQUIRE: (newValue <= [maxValue\(\)](#))

Reimplemented in [RTRPublicNumericRange](#).

RTRManagedLargeNumeric - Comparison

Functions

- RTRBOOL [RTRManagedLargeNumeric::operator==](#) (RTR_I64) const

Function Documentation

RTRBOOL RTRManagedLargeNumeric::operator== (RTR_I64) const [inline, inherited]
Comparison

RTRManagedLargeNumeric - Access

Functions

- RTR_I64 [RTRManagedLargeNumeric::value](#) () const

Function Documentation

RTR_I64 RTRManagedLargeNumeric::value () const [inline, inherited]
The current value of this variable.

RTRManagedLargeNumeric - Transformation

Functions

- [RTRManagedNumeric::operator](#) RTR_I64 () const
- virtual [RTRString RTRManagedNumeric::toString](#) () const

Function Documentation

RTRManagedLargeNumeric::operator RTR_I64 () const [inline, inherited]
This variable as an RTR_I64 (long long for Unix or __int64 for Windows).

virtual [RTRString](#) RTRManagedLargeNumeric::toString () const [virtual, inherited]
A copy of this variable value, represented as a string

Implements [RTRManagedVariable](#).

RTRManagedNumeric - Comparison

Functions

- RTRBOOL [RTRManagedNumeric::operator==](#) (long) const

Function Documentation

RTRBOOL RTRManagedNumeric::operator== (long) const [inline, inherited]
Comparison

RTRManagedNumeric - Access

Functions

- long [RTRManagedNumeric::value](#) () const

Function Documentation

long [RTRManagedNumeric::value](#) () const [inline, inherited]

The current value of this variable.

RTRManagedNumeric - Transformation

Functions

- [RTRManagedNumeric::operator long](#) () const
- virtual [RTRString RTRManagedNumeric::toString](#) () const

Function Documentation

[RTRManagedNumeric::operator long](#) () const [inline, inherited]

This variable as an long.

virtual [RTRString RTRManagedNumeric::toString](#) () const [virtual, inherited]

A copy of this variable value, represented as a string

Implements [RTRManagedVariable](#).

RTRManagedObject - Identity

Functions

- const [RTRObjectId](#) & [RTRManagedObject::classId](#) () const
- const [RTRObjectId](#) & [RTRManagedObject::instanceId](#) () const
- const [RTRString](#) & [RTRManagedObject::name](#) () const

Function Documentation

const [RTRObjectId](#) & [RTRManagedObject::classId](#) () const [inline, inherited]

The class (type) identifier of this object.

const [RTRObjectId](#) & [RTRManagedObject::instanceId](#) () const [inline, inherited]

The instance identifier of this object.

const [RTRString](#) & [RTRManagedObject::name](#) () const [inline, inherited]

The name of this object.

RTRManagedObject - Attributes

Functions

- [RTRManagedObject](#) * [RTRManagedObject::parent](#) () const
- MOState [RTRManagedObject::state](#) () const

- MOState [RTRManagedObject::previousState](#) () const
- const char * [RTRManagedObject::text](#) () const
- const [RTRString](#) & [RTRManagedObject::description](#) () const

Function Documentation

[RTRManagedObject](#) * [RTRManagedObject::parent](#) () const [inline, inherited]

The parent (if any) of this object.

[RTRManagedObject::MOState](#) [RTRManagedObject::state](#) () const [inline, inherited]

The state value of this object.

[RTRManagedObject::MOState](#) [RTRManagedObject::previousState](#) () const [inline, inherited]

The state value of this object.

const char * [RTRManagedObject::text](#) () const [inline, inherited]

Textual information regarding the state of this object.

const [RTRString](#) & [RTRManagedObject::description](#) () const [inline, inherited]

The definition of this class

RTRManagedObject - State

Functions

- RTRBOOL [RTRManagedObject::isInitializing](#) () const
- RTRBOOL [RTRManagedObject::isNormal](#) () const
- RTRBOOL [RTRManagedObject::isRecovering](#) () const
- RTRBOOL [RTRManagedObject::isWaiting](#) () const
- RTRBOOL [RTRManagedObject::isInterrupted](#) () const
- RTRBOOL [RTRManagedObject::isDead](#) () const

Function Documentation

RTRBOOL [RTRManagedObject::isInitializing](#) () const [inline, inherited]

Is this object in an initialization state?

RTRBOOL [RTRManagedObject::isNormal](#) () const [inline, inherited]

Is this object in a normal state?

RTRBOOL [RTRManagedObject::isRecovering](#) () const [inline, inherited]

Is this object recovering service automatically?

RTRBOOL [RTRManagedObject::isWaiting](#) () const [inline, inherited]

Is this object waiting for manual recovery?

RTRBOOL [RTRManagedObject::isInterrupted](#) () const [inline, inherited]

Is this object in a service interrupted state?

ENSURE: result == [isRecovering\(\)](#) || [isWaiting\(\)](#)

RTRBOOL [RTRManagedObject::isDead](#) () const [inline, inherited]

Is this object in an unrecoverable error condition representing non-graceful exit?

RTRManagedObject - Query

Functions

- RTRBOOL [RTRManagedObject::hasChild](#) (const [RTRString](#) &) const
- RTRBOOL [RTRManagedObject::hasVariable](#) (const [RTRString](#) &) const

Function Documentation

RTRBOOL [RTRManagedObject::hasChild](#) (const [RTRString](#) &) const [inherited]

Does this object have a child with the given name?

RTRBOOL [RTRManagedObject::hasVariable](#) (const [RTRString](#) &) const [inherited]

Does this object have a variable with the given name?

RTRManagedObject - Access Sequentially

Functions

- [RTRManagedObjectIterator](#) [RTRManagedObject::childIterator](#) () const
- [RTRManagedVariableIterator](#) [RTRManagedObject::variableIterator](#) () const

Function Documentation

[RTRManagedObjectIterator](#) [RTRManagedObject::childIterator](#) () const [inherited]

The manageable children of this instance.

[RTRManagedVariableIterator](#) [RTRManagedObject::variableIterator](#) () const [inherited]

The manageable variables of this instance.

RTRManagedObject - Access Randomly

Functions

- [RTRManagedObject](#) * [RTRManagedObject::childByName](#) (const char *) const
- [RTRManagedVariable](#) * [RTRManagedObject::variableByName](#) (const char *) const
- [RTRManagedBoolean](#) * [RTRManagedObject::booleanByName](#) (const char *) const
- [RTRManagedBooleanConfig](#) * [RTRManagedObject::booleanConfigByName](#) (const char *) const
- [RTRManagedCounter](#) * [RTRManagedObject::counterByName](#) (const char *) const
- [RTRManagedGauge](#) * [RTRManagedObject::gaugeByName](#) (const char *) const
- [RTRManagedGaugeConfig](#) * [RTRManagedObject::gaugeConfigByName](#) (const char *) const
- [RTRManagedNumeric](#) * [RTRManagedObject::numericByName](#) (const char *) const
- [RTRManagedNumericConfig](#) * [RTRManagedObject::numericConfigByName](#) (const char *) const
- [RTRManagedNumericRange](#) * [RTRManagedObject::numericRangeByName](#) (const char *) const
- [RTRManagedString](#) * [RTRManagedObject::stringByName](#) (const char *) const
- [RTRManagedStringConfig](#) * [RTRManagedObject::stringConfigByName](#) (const char *) const

Function Documentation

RTRManagedObject* **RTRManagedObject::childByName (const char *) const** [inherited]

The child, if any, with the given name.

RTRManagedVariable* **RTRManagedObject::variableByName (const char *) const** [inherited]

The variable, if any, with the given name.

ENSURE: result == null implies !hasVariable(name)

RTRManagedBoolean* **RTRManagedObject::booleanByName (const char *) const** [inherited]

The boolean, if any, with the given name.

ENSURE: result == null implies !hasVariable(name) || (

variableByName(name)->type != Boolean &&

variableByName(name)->type != BooleanConfig)

RTRManagedBooleanConfig* **RTRManagedObject::booleanConfigByName (const char *) const** [inherited]

The boolean config, if any, with the given name.

ENSURE: result == null implies !hasVariable(name) ||

variableByName(name)->type != BooleanConfig

RTRManagedCounter* **RTRManagedObject::counterByName (const char *) const** [inherited]

The counter, if any, with the given name.

ENSURE: result == null implies !hasVariable(name) ||

variableByName(name)->type != Counter

RTRManagedGauge* **RTRManagedObject::gaugeByName (const char *) const** [inherited]

The gauge, if any, with the given name.

ENSURE: result == null implies !hasVariable(name) || (

variableByName(name)->type != Gauge &&

variableByName(name)->type != GaugeConfig)

RTRManagedGaugeConfig* **RTRManagedObject::gaugeConfigByName (const char *) const** [inherited]

The gauge config, if any, with the given name.

ENSURE: result == null implies !hasVariable(name) ||

variableByName(name)->type != GaugeConfig

RTRManagedNumeric* **RTRManagedObject::numericByName (const char *) const** [inherited]

The numeric, if any, with the given name.

ENSURE: result == null implies !hasVariable(name) || (

variableByName(name)->type != Numeric &&

variableByName(name)->type != NumericConfig &&

variableByName(name)->type != NumericRange &&

variableByName(name)->type != Gauge &&

variableByName(name)->type != GaugeConfig)

RTRManagedNumericConfig* **RTRManagedObject::numericConfigByName (const char *) const** [inherited]

The numeric config, if any, with the given name.

ENSURE: result == null implies !hasVariable(name) ||

variableByName(name)->type != NumericConfig

[RTRManagedNumericRange](#)* **RTRManagedObject::numericRangeByName (const char *) const** [inherited]

The numeric range, if any, with the given name.

ENSURE: result == null implies !hasVariable(name) ||

variableByName(name)->type != NumericRange

[RTRManagedString](#)* **RTRManagedObject::stringByName (const char *) const** [inherited]

The string, if any, with the given name.

ENSURE: result == null implies !hasVariable(name) || (<>BR variableByName(name)->type != String &&

variableByName(name)->type != StringConfig)

[RTRManagedStringConfig](#)* **RTRManagedObject::stringConfigByName (const char *) const** [inherited]

The string config, if any, with the given name.

ENSURE: result == null implies !hasVariable(name) ||

variableByName(name)->type != StringConfig

RTRManagedObject - Client management

Functions

- void [RTRManagedObject::addClient](#) ([RTRManagedObjectClient](#) &client)
- void [RTRManagedObject::dropClient](#) ([RTRManagedObjectClient](#) &client)
- RTRBOOL [RTRManagedObject::hasClient](#) ([RTRManagedObjectClient](#) &client) const

Function Documentation

void **RTRManagedObject::addClient** ([RTRManagedObjectClient](#) & *client*) [inherited]

EQUIRE : !hasClient(client);

ENSURE : hasClient(client);

void **RTRManagedObject::dropClient** ([RTRManagedObjectClient](#) & *client*) [inherited]

EQUIRE : hasClient(client);

ENSURE : !hasClient(client);

RTRBOOL **RTRManagedObject::hasClient** ([RTRManagedObjectClient](#) & *client*) const [inherited]

Is the given client registered to receive events from this managed object?

RTRManagedObject - Operations from RTRLockableObj

Functions

- virtual void [RTRManagedObject::lock](#) ()
- virtual void [RTRManagedObject::unlock](#) ()
- virtual RTRBOOL [RTRManagedObject::locked](#) () const

Function Documentation

virtual void RTRManagedObject::lock () [virtual, inherited]

Operations - From [RTRLockableObj](#)

Reimplemented from [RTRLockableObj](#). **virtual void RTRManagedObject::unlock ()** [virtual, inherited]

Operations - From [RTRLockableObj](#)

Reimplemented from [RTRLockableObj](#). **virtual RTRBOOL RTRManagedObject::locked () const** [virtual, inherited]

Operations - From [RTRLockableObj](#)

Reimplemented from [RTRLockableObj](#).

RTRManagedObject - Private Implementation

Functions

- **RTRMOImpl * RTRManagedObject::storeImpl () const**
- **RTRMOImplPub * RTRManagedObject::storeImplPub () const**
- **void RTRManagedObject::cleanUpImplPub ()**
- **RTRBOOL RTRManagedObject::initImplPub (RTRManagedMemAllocator &, RTRBOOL=RTRTRUE)**

Function Documentation

RTRMOImpl * RTRManagedObject::storeImpl () const [inline, inherited]

Do not use.

RTRMOImplPub * RTRManagedObject::storeImplPub () const [inline, inherited]

Do not use.

void RTRManagedObject::cleanUpImplPub () [inherited]

Do not use.

RTRBOOL RTRManagedObject::initImplPub (RTRManagedMemAllocator &, RTRBOOL = RTRTRUE) [inherited]

Do not use.

RTRManagedObject - Private Event processing

Functions

- **virtual void RTRManagedObject::processParameterChange (RTRManagedVariable &)**
- **virtual void RTRManagedObject::processConfigChange (RTRManagedVariable &)**

Function Documentation

virtual void RTRManagedObject::processParameterChange (RTRManagedVariable &) [virtual, inherited]

One of the parameter variables contained by this object has been changed.

virtual void RTRManagedObject::processConfigChange (RTRManagedVariable &) [virtual, inherited]

One of the configuration variables contained by this object has been changed.

RTRManagedObjectClient - Event processing

Functions

- virtual void [RTRManagedObjectClient::processObjectDeleted](#) ([RTRManagedObject](#) &)=0
- virtual void [RTRManagedObjectClient::processObjectInService](#) ([RTRManagedObject](#) &)
- virtual void [RTRManagedObjectClient::processObjectRecovering](#) ([RTRManagedObject](#) &)
- virtual void [RTRManagedObjectClient::processObjectWaiting](#) ([RTRManagedObject](#) &)
- virtual void [RTRManagedObjectClient::processObjectDead](#) ([RTRManagedObject](#) &)
- virtual void [RTRManagedObjectClient::processObjectInfo](#) ([RTRManagedObject](#) &)
- virtual void [RTRManagedObjectClient::processChildAdded](#) ([RTRManagedObject](#) &, [RTRManagedObject](#) &ch)
- virtual void [RTRManagedObjectClient::processChildRemoved](#) ([RTRManagedObject](#) &, [RTRManagedObject](#) &ch)
- virtual void [RTRManagedObjectClient::processVariableAdded](#) ([RTRManagedObject](#) &, [RTRManagedVariable](#) &)
- virtual void [RTRManagedObjectClient::processVariableRemoved](#) ([RTRManagedObject](#) &, [RTRManagedVariable](#) &)

Function Documentation

virtual void RTRManagedObjectClient::processObjectDeleted ([RTRManagedObject](#) &) [pure virtual, inherited]
The given managed object has been deleted.

virtual void RTRManagedObjectClient::processObjectInService ([RTRManagedObject](#) &) [virtual, inherited]
The given managed object is now in a normal service state.

virtual void RTRManagedObjectClient::processObjectRecovering ([RTRManagedObject](#) &) [virtual, inherited]
The given managed object is in a service interrupted state but is attempting to recover normal service automatically.

virtual void RTRManagedObjectClient::processObjectWaiting ([RTRManagedObject](#) &) [virtual, inherited]
The given managed object is in a service interrupted state and is waiting for manual intervention to restore normal service.

virtual void RTRManagedObjectClient::processObjectDead ([RTRManagedObject](#) &) [virtual, inherited]
The given managed object has entered an unrecoverable error state.

virtual void RTRManagedObjectClient::processObjectInfo ([RTRManagedObject](#) &) [virtual, inherited]
The given managed object has changed its informational text.

virtual void RTRManagedObjectClient::processChildAdded ([RTRManagedObject](#) &, [RTRManagedObject](#) &ch) [virtual, inherited]
The given managed object has a new child.

virtual void RTRManagedObjectClient::processChildRemoved ([RTRManagedObject](#) &, [RTRManagedObject](#) &ch) [virtual, inherited]
The given managed object has had a child removed.

virtual void RTRManagedObjectClient::processVariableAdded ([RTRManagedObject](#) &, [RTRManagedVariable](#) &) [virtual, inherited]
The given managed object has a new variable.

virtual void RTRManagedObjectClient::processVariableRemoved ([RTRManagedObject](#) &, [RTRManagedVariable](#) &) [virtual, inherited]
The given managed object has had a variable removed.

RTRManagedObjectIterator - Attributes

Functions

- int [RTRManagedObjectIterator::count](#) () const

Function Documentation

int RTRManagedObjectIterator::count () const [inherited]

The number of children available via this iterator.

RTRManagedObjectIterator - State

Functions

- RTRBOOL [RTRManagedObjectIterator::off](#) () const
- RTRBOOL [RTRManagedObjectIterator::empty](#) () const

Function Documentation

RTRBOOL RTRManagedObjectIterator::off () const [inherited]

Is this iteration complete?

RTRBOOL RTRManagedObjectIterator::empty () const [inherited]

Are there no children available via this iterator?

ENSURE: result implies [count\(\)](#) == 0

RTRManagedObjectIterator - Access

Functions

- [RTRManagedObject](#) & [RTRManagedObjectIterator::item](#) () const

Function Documentation

[RTRManagedObject](#) & RTRManagedObjectIterator::item () const [inherited]

The current item in the current iteration.

RTRManagedObjectIterator - Operations

Functions

- void [RTRManagedObjectIterator::start](#) ()
- void [RTRManagedObjectIterator::finish](#) ()
- void [RTRManagedObjectIterator::forth](#) ()
- void [RTRManagedObjectIterator::back](#) ()

Function Documentation

void RTRManagedObjectIterator::start () [inherited]

Start a new iteration.

ENSURE: [off\(\)](#) implies [empty\(\)](#)

void RTRManagedObjectIterator::finish () [inherited]

Start an iteration from the last available child.

ENSURE: [off\(\)](#) implies [empty\(\)](#)

void RTRManagedObjectIterator::forth () [inherited]

Continue the current iteration from [start\(\)](#) to [finish\(\)](#).

REQUIRE: !off()

void RTRManagedObjectIterator::back () [inherited]

Continue the current iteration from [finish\(\)](#) to [start\(\)](#).

REQUIRE: !off()

RTRManagedVariableIterator - Attributes

Functions

- int [RTRManagedVariableIterator::count](#) () const

Function Documentation

int RTRManagedVariableIterator::count () const [inherited]

The number of variables available via this iterator.

RTRManagedVariableIterator - State

Functions

- RTRBOOL [RTRManagedVariableIterator::off](#) () const
- RTRBOOL [RTRManagedVariableIterator::empty](#) () const

Function Documentation

RTRBOOL RTRManagedVariableIterator::off () const [inherited]

Is this iteration complete?

RTRBOOL RTRManagedVariableIterator::empty () const [inherited]

Are there no variable available via this iterator?

ENSURE: result implies [count\(\)](#) == 0

RTRManagedVariableIterator - Access

Functions

- [RTRManagedVariable](#) & [RTRManagedVariableIterator::item](#) () const

Function Documentation

[RTRManagedVariable](#) & RTRManagedVariableIterator::item () const [inherited]

The current item in the current iteration.

RTRManagedVariableIterator - Operations

Functions

- void [RTRManagedVariableIterator::start](#) ()
- void [RTRManagedVariableIterator::finish](#) ()
- void [RTRManagedVariableIterator::forth](#) ()
- void [RTRManagedVariableIterator::back](#) ()

Function Documentation

void RTRManagedVariableIterator::start () [inherited]

Start a new iteration.

ENSURE: [off\(\)](#) implies [empty\(\)](#)

void RTRManagedVariableIterator::finish () [inherited]

Start an iteration from the last available variable.

ENSURE: [off\(\)](#) implies [empty\(\)](#)

void RTRManagedVariableIterator::forth () [inherited]

Continue the current iteration from [start\(\)](#) to [finish\(\)](#).

REQUIRE: ![off\(\)](#)

void RTRManagedVariableIterator::back () [inherited]

Continue the current iteration from [finish\(\)](#) to [start\(\)](#).

REQUIRE: ![off\(\)](#)

RTRManagedObjectDirectory - Attributes

Functions

- [RTRManagedObjDirRootIterator](#) [RTRManagedObjectDirectory::rootIterator](#) ()

Function Documentation

[RTRManagedObjDirRootIterator](#) RTRManagedObjectDirectory::rootIterator () [inherited]

Sequential access to the list of managed object instances which have no parent object.

RTRManagedObjectDirectory - Access

Functions

- const RTRClassCategory< [RTRManagedObject](#) > & [RTRManagedObjectDirectory::automaticCategory](#) (const [RTRObjectId](#) &cid)
const

Function Documentation

const RTRClassCategory<[RTRManagedObject](#)>& RTRManagedObjectDirectory::automaticCategory (const [RTRObjectId](#) & cid)
const [inherited]

The category with the given class (category is allocated if it doesn't currently exist in the directory).

ENSURE : result.classCategory() == cid

RTRManagedObjectDirectory - Insertion

Functions

- virtual void [RTRManagedObjectDirectory::put](#) ([RTRManagedObject](#) &mo)

Function Documentation

virtual void [RTRManagedObjectDirectory::put](#) ([RTRManagedObject](#) & mo) [virtual, inherited]

Add the given interface instance to the directory.

REQUIRE : !has(mo.instanceId());

ENSURE : has(mo.instanceId());

ENSURE : category(mo.classId()) != NULL

RTRManagedObjectDirectory - Deletion

Functions

- virtual void [RTRManagedObjectDirectory::remove](#) ([RTRManagedObject](#) &mo)

Function Documentation

virtual void [RTRManagedObjectDirectory::remove](#) ([RTRManagedObject](#) & mo) [virtual, inherited]

Remove the given interface instance from the directory.

REQUIRE : has(mo.instanceId());

ENSURE : !has(mo.instanceId());

RTRManagedObjectDirectory - Client management

Functions

- void [RTRManagedObjectDirectory::addClient](#) ([RTRManagedObjDirClient](#) &newClient)
- void [RTRManagedObjectDirectory::dropClient](#) ([RTRManagedObjDirClient](#) &oldClient)
- RTRBOOL [RTRManagedObjectDirectory::hasClient](#) ([RTRManagedObjDirClient](#) &client) const

Function Documentation

void [RTRManagedObjectDirectory::addClient](#) ([RTRManagedObjDirClient](#) & newClient) [inherited]

Register the given client to receive events from this directory.

REQUIRE : !hasClient(newClient)

ENSURE : hasClient(newClient)

void [RTRManagedObjectDirectory::dropClient](#) ([RTRManagedObjDirClient](#) & oldClient) [inherited]

Un-register the given client to receive events from this directory.

REQUIRE : hasClient(newClient)

ENSURE : !hasClient(oldClient)

RTRBOOL [RTRManagedObjectDirectory::hasClient](#) ([RTRManagedObjDirClient](#) & client) const [inherited]

Is the given client registered to receive events from this directory?

RTRManagedStringConfig - Assignment

Functions

- [RTRManagedStringConfig](#) & [RTRManagedStringConfig::operator=](#) (const char *rhs)

Function Documentation

[RTRManagedStringConfig](#) & [RTRManagedStringConfig::operator=](#) (const char * *rhs*) [inline, inherited]

Set the active value to rhs

REQUIRE: [modifyEnabled\(\)](#)

Synchronized

Reimplemented from [RTRManagedString](#). Reimplemented in [RTRPublicStringConfig](#).

RTRManagedStringConfig - Attributes

Functions

- [RTRString](#) [RTRManagedStringConfig::activeValue](#) () const
- [RTRString](#) [RTRManagedStringConfig::storeValue](#) () const
- [RTRString](#) [RTRManagedStringConfig::factoryDefault](#) () const
- RTRBOOL [RTRManagedStringConfig::hasStore](#) () const
- RTRBOOL [RTRManagedStringConfig::isStoreActive](#) () const
- RTRBOOL [RTRManagedStringConfig::isStoreClassConfig](#) () const
- RTRBOOL [RTRManagedStringConfig::isStoreInstanceConfig](#) () const

Function Documentation

[RTRString](#) [RTRManagedStringConfig::activeValue](#) () const [inline, inherited]

A synonym for [value\(\)](#)

[RTRString](#) [RTRManagedStringConfig::storeValue](#) () const [inline, inherited]

The store value

[RTRString](#) [RTRManagedStringConfig::factoryDefault](#) () const [inline, inherited]

The factory default value

RTRBOOL [RTRManagedStringConfig::hasStore](#) () const [inherited]

Is the variable a client of a RTRVariableConfig?

RTRBOOL [RTRManagedStringConfig::isStoreActive](#) () const [inherited]

Is the RTRVariableConfig in an active state?

REQUIRE: [hasStore\(\)](#)

RTRBOOL [RTRManagedStringConfig::isStoreClassConfig](#) () const [inherited]

Is the RTRVariableConfig's context a RTRClassConfig? class config?

REQUIRE: [hasStore\(\)](#)

RTRBOOL [RTRManagedStringConfig::isStoreInstanceConfig](#) () const [inherited]

Is the RTRVariableConfig's context a RTRInstanceConfig? instance config?

REQUIRE: [hasStore\(\)](#)

RTRManagedString - Access

Functions

- [RTRString RTRManagedString::value](#) () const

Function Documentation

[RTRString](#) [RTRManagedString::value](#) () const [inline, inherited]

A copy of the current value of this variable.

RTRManagedString - Transformation

Functions

- [RTRManagedString::operator const char *](#) () const
- virtual [RTRString RTRManagedString::toString](#) () const

Function Documentation

[RTRManagedString::operator const char *](#) () const [inline, inherited]

This variable as a C string (null terminated).

virtual [RTRString](#) [RTRManagedString::toString](#) () const [virtual, inherited]

A copy of this variables value, represented as a string.

Implements [RTRManagedVariable](#).

RTRManagedString - Comparison

Functions

- RTRBOOL [RTRManagedString::operator==](#) (const char *) const

Function Documentation

RTRBOOL [RTRManagedString::operator==](#) (const char *) const [inline, inherited]

Comparison

RTRManagedString - Attributes

Functions

- RTRBOOL [RTRManagedString::modifyEnabled](#) () const

Function Documentation

RTRBOOL [RTRManagedString::modifyEnabled](#) () const [inline, inherited]

Is the managing application permitted to modify this variable?

RTRManagedString - Assignment

Functions

- [RTRManagedString](#) & [RTRManagedString::operator=](#) (const char *rhs)

Function Documentation

[RTRManagedString](#) & [RTRManagedString::operator=](#) (const char * *rhs*) [inline, inherited]

Assigns the value of this variable to rhs

REQUIRE: [modifyEnabled\(\)](#)

Synchronized

Reimplemented in [RTRManagedStringConfig](#), [RTRPublicStringConfig](#), and [RTRPublicString](#).

RTRManagedString - Operations

Functions

- virtual void [RTRManagedString::set](#) (const char *newValue)

Function Documentation

virtual void [RTRManagedString::set](#) (const char * *newValue*) [virtual, inherited]

A synonym for [operator=\(\)](#)

REQUIRE: [modifyEnabled\(\)](#)

Synchronized

Reimplemented in [RTRPublicStringConfig](#), and [RTRPublicString](#).

RTRManagedVariable - Identity

Functions

- const [RTRString](#) & [RTRManagedVariable::name](#) () const

Function Documentation

const [RTRString](#) & [RTRManagedVariable::name](#) () const [inline, inherited]

Identity

RTRManagedVariable - Attributes

Functions

- MVType [RTRManagedVariable::type](#) () const
- [RTRManagedObject](#) & [RTRManagedVariable::context](#) () const
- const [RTRString](#) & [RTRManagedVariable::description](#) () const

Function Documentation

[RTRManagedVariable::MVType](#) [RTRManagedVariable::type](#) () const [inline, inherited]

The type of this variable.

[RTRManagedObject](#) & RTRManagedVariable::context () const [inline, inherited]

The managed object which contains this variable.

const [RTRString](#) & RTRManagedVariable::description () const [inline, inherited]

The definition of this variable.

RTRManagedVariable - Transformation

Functions

- RTRManagedVariable::operator RTRManagedBoolean & () const
- RTRManagedVariable::operator RTRManagedBooleanConfig & () const
- RTRManagedVariable::operator RTRManagedCounter & () const
- RTRManagedVariable::operator RTRManagedGauge & () const
- RTRManagedVariable::operator RTRManagedGaugeConfig & () const
- RTRManagedVariable::operator RTRManagedNumeric & () const
- RTRManagedVariable::operator RTRManagedLargeNumeric & () const
- RTRManagedVariable::operator RTRManagedNumericRange & () const
- RTRManagedVariable::operator RTRManagedNumericConfig & () const
- RTRManagedVariable::operator RTRManagedStringConfig & () const
- RTRManagedVariable::operator RTRManagedString & () const
- virtual [RTRString](#) RTRManagedVariable::toString () const =0

Function Documentation

RTRManagedVariable::operator [RTRManagedBoolean](#) & () const [inherited]

REQUIRE : [type\(\)](#) == Boolean ||

[type\(\)](#) == BooleanConfig

RTRManagedVariable::operator [RTRManagedBooleanConfig](#) & () const [inherited]

REQUIRE : [type\(\)](#) == BooleanConfig

RTRManagedVariable::operator [RTRManagedCounter](#) & () const [inherited]

REQUIRE : [type\(\)](#) == Counter

RTRManagedVariable::operator [RTRManagedGauge](#) & () const [inherited]

REQUIRE : [type\(\)](#) == Gauge ||

[type\(\)](#) == GaugeConfig

RTRManagedVariable::operator [RTRManagedGaugeConfig](#) & () const [inherited]

REQUIRE : [type\(\)](#) == GaugeConfig

RTRManagedVariable::operator [RTRManagedLargeNumeric](#) & () const [inherited]

REQUIRE : [type\(\)](#) == LargeNumeric

RTRManagedVariable::operator [RTRManagedNumeric](#) & () const [inherited]

REQUIRE : [type\(\)](#) == Numeric ||

```

type() == Gauge ||
type() == GaugeConfig ||
type() == NumericRange ||
type() == NumericConfig

```

RTRManagedVariable::operator [RTRManagedNumericRange](#) & () const [inherited]
REQUIRE : [type\(\)](#) == NumericRange

RTRManagedVariable::operator [RTRManagedNumericConfig](#) & () const [inherited]
REQUIRE : [type\(\)](#) == NumericConfig

RTRManagedVariable::operator [RTRManagedStringConfig](#) & () const [inherited]
REQUIRE : [type\(\)](#) == StringConfig

RTRManagedVariable::operator [RTRManagedString](#) & () const [inherited]
REQUIRE : [type\(\)](#) == String ||
[type\(\)](#) == StringConfig

virtual [RTRString](#) RTRManagedVariable::toString () const [pure virtual, inherited]
A copy of this variable's value, represented as a string.

Implemented in [RTRManagedBoolean](#), [RTRManagedCounter](#), [RTRManagedLargeNumeric](#), [RTRManagedNumeric](#), and [RTRManagedString](#).

RTRManagedVariable - Client management

Functions

- virtual void [RTRManagedVariable::addClient](#) ([RTRManagedVariableClient](#) &newClient)
- virtual void [RTRManagedVariable::dropClient](#) ([RTRManagedVariableClient](#) &oldClient)
- RTRBOOL [RTRManagedVariable::hasClient](#) ([RTRManagedVariableClient](#) &client) const
- RTRBOOL [RTRManagedVariable::hasClients](#) () const

Function Documentation

virtual void RTRManagedVariable::addClient ([RTRManagedVariableClient](#) & newClient) [virtual, inherited]
Register the given client to receive events from this variable.
ENSURE : hasClient(newClient)

virtual void RTRManagedVariable::dropClient ([RTRManagedVariableClient](#) & oldClient) [virtual, inherited]
Un-register the given client to receive events from this variable.
ENSURE : !hasClient(oldClient)

RTRBOOL RTRManagedVariable::hasClient ([RTRManagedVariableClient](#) & client) const [inherited]
Is the given client registered to receive events from this variable?

RTRBOOL RTRManagedVariable::hasClients () const [inline, inherited]
Does this variable have any clients?

RTRManagedVariable - Operations

Functions

- virtual void [RTRManagedVariable::lock](#) ()
- virtual void [RTRManagedVariable::unlock](#) ()
- virtual RTRBOOL [RTRManagedVariable::locked](#) () const

Function Documentation

virtual void RTRManagedVariable::lock () [virtual, inherited]

Operations from [RTRLockableObj](#)

Reimplemented from [RTRLockableObj](#). virtual void RTRManagedVariable::unlock () [virtual, inherited]

Operations from [RTRLockableObj](#)

Reimplemented from [RTRLockableObj](#). virtual RTRBOOL RTRManagedVariable::locked () const [virtual, inherited]

Operations from [RTRLockableObj](#)

Reimplemented from [RTRLockableObj](#).

RTRManagedVariable - Private Implementation

Functions

- RTRMVImp * [RTRManagedVariable::storeImpl](#) () const
- RTRMVImpIPub * [RTRManagedVariable::storeImplPub](#) () const

Function Documentation

RTRMVImp * RTRManagedVariable::storeImpl () const [inline, inherited]

Do not use.

RTRMVImpIPub * RTRManagedVariable::storeImplPub () const [inline, inherited]

Do not use.

RTRManagedVariableClient - Event processing

Functions

- virtual void [RTRManagedVariableClient::processVariableChange](#) ([RTRManagedVariable](#) &)=0
- virtual void [RTRManagedVariableClient::processVariableDelete](#) ([RTRManagedVariable](#) &)=0

Function Documentation

virtual void RTRManagedVariableClient::processVariableChange ([RTRManagedVariable](#) &) [pure virtual, inherited]

The given variable has changed.

virtual void RTRManagedVariableClient::processVariableDelete ([RTRManagedVariable](#) &) [pure virtual, inherited]

The given variable has been deleted.

RTRObjectId - Attributes

Functions

- [RTRString RTRObjectId::name](#) () const
- [RTRString RTRObjectId::base](#) () const
- int [RTRObjectId::numberOfElements](#) () const
- int [RTRObjectId::count](#) () const
- unsigned long [RTRObjectId::hash](#) () const

Function Documentation

[RTRString RTRObjectId::name](#) () const [inherited]

The name portion of this object id.

ENSURE: RTRObjectId(result) == lastN(1)

[RTRString RTRObjectId::base](#) () const [inherited]

The base portion of this object id.

ENSURE: RTRObjectId(result) == firstN(1)

int RTRObjectId::numberOfElements () const [inherited]

The number of names which comprise this object id.

int RTRObjectId::count () const [inline, inherited]

The number of characters in this object id.

unsigned long RTRObjectId::hash () const [inherited]

Hash code of this id.

RTRObjectId - State

Functions

- RTRBOOL [RTRObjectId::isEmpty](#) () const

Function Documentation

RTRBOOL RTRObjectId::isEmpty () const [inherited]

Is there no value for this id?

RTRObjectId - Access

Functions

- RTRLinkedList< [RTRString](#) > * [RTRObjectId::lineage](#) () const
- [RTRString RTRObjectId::iTh](#) (int i) const

Function Documentation

RTRLinkedList<[RTRString](#)>* RTRObjectId::lineage () const [inherited]

Return a list of names representing the lineage of the object id.

[RTRString](#) **RTRObjectId::iTh** (int *i*) const [inherited]

A new object id comprised of the *i*Th element of this id.

REQUIRE: *i* <= [numberOfElements\(\)](#)

RTRObjectId - Query

Functions

- **RTRBOOL** [RTRObjectId::isDescendant](#) (const [RTRObjectId](#) &*other*) const
- **RTRBOOL** [RTRObjectId::conformsTo](#) (const [RTRObjectId](#) &*other*) const

Function Documentation

RTRBOOL **RTRObjectId::isDescendant** (const [RTRObjectId](#) &*other*) const [inherited]

Is this instance id a "contained" by other?

ENSURE: !commonRoot(*other*).[isEmpty\(\)](#)

RTRBOOL **RTRObjectId::conformsTo** (const [RTRObjectId](#) &*other*) const [inherited]

Is this class id a "descendant" of other?

Synonym for [isDescendant\(\)](#)

RTRObjectId - Comparison

Functions

- **RTRBOOL** [RTRObjectId::operator==](#) (const [RTRObjectId](#) &*other*) const
- **RTRBOOL** [RTRObjectId::operator!=](#) (const [RTRObjectId](#) &*other*) const
- **RTRBOOL** [RTRObjectId::operator==](#) (const [RTRString](#) &*other*) const
- **RTRBOOL** [RTRObjectId::operator==](#) (const char **other*) const

Function Documentation

RTRBOOL **RTRObjectId::operator==** (const [RTRObjectId](#) &*other*) const [inherited]

Is this id exactly like other?

RTRBOOL **RTRObjectId::operator!=** (const [RTRObjectId](#) &*other*) const [inherited]

Is this id not equal to other?

RTRBOOL **RTRObjectId::operator==** (const [RTRString](#) &*other*) const [inherited]

Is this id exactly equal to other?

RTRBOOL **RTRObjectId::operator==** (const char **other*) const [inherited]

Is this id exactly equal to other?

RTRObjectId - Transformation

Functions

- operator const char * () const
- [RTRString](#) [RTRObjectId::string](#) () const

- [RTRString RTRObjectId::delimitedString](#) (char delimiter) const
- [RTRObjectId RTRObjectId::firstN](#) (int n) const
- [RTRObjectId RTRObjectId::lastN](#) (int n) const
- [RTRObjectId RTRObjectId::parent](#) () const
- [RTRObjectId RTRObjectId::commonRoot](#) (const [RTRObjectId](#) &other) const

Function Documentation

[RTRString](#) [RTRObjectId::string](#) () const [inherited]

Return the string representation of this id.

[RTRString](#) [RTRObjectId::delimitedString](#) (char *delimiter*) const [inherited]

Return the string representation of this id using given delimiter in place of the default delimiter.

[RTRObjectId](#) [RTRObjectId::firstN](#) (int *n*) const [inherited]

A new object id comprised of the first n elements of this id.

REQUIRE: $n \leq \text{numberOfElements}()$

[RTRObjectId](#) [RTRObjectId::lastN](#) (int *n*) const [inherited]

A new object id comprised of the last n elements of this id.

REQUIRE: $n \leq \text{numberOfElements}()$

[RTRObjectId](#) [RTRObjectId::parent](#) () const [inherited]

Return the object id for the parent object (empty if no parent)

ENSURE: $\text{result} == \text{firstN}(\text{numberOfElements}() - 1)$

[RTRObjectId](#) [RTRObjectId::commonRoot](#) (const [RTRObjectId](#) &other) const [inherited]

A new object id comprised of the portion of this object id that is in common with the other object id.

RTRObjectId - Assignment

Functions

- [RTRObjectId](#) & [RTRObjectId::operator=](#) (const [RTRObjectId](#) &rhs)

Function Documentation

[RTRObjectId](#) & [RTRObjectId::operator=](#) (const [RTRObjectId](#) &rhs) [inherited]

ENSURE: $\text{firstN}(\text{numberOfElements}()) == \text{rhs.firstN}(\text{rhs.numberOfElements}())$

RTRObjectId - Modification

Functions

- void [RTRObjectId::set](#) ([RTRString](#) &s, int n1, int n2)

Function Documentation

void [RTRObjectId::set](#) ([RTRString](#) &s, int *n1*, int *n2*) [inherited]

Set this id to the value of "s" from index n1 to index n2 of "s".

RTRObjectId - OBSOLETE

Functions

- `const char * RTRObjectId::to_c () const`
- `RTRBOOL RTRObjectId::equivalent (const RTRObjectId &other) const`

Function Documentation

`const char* RTRObjectId::to_c () const` *[inherited]*

Char * representation of this string. Return up to first null value in id.

`RTRBOOL RTRObjectId::equivalent (const RTRObjectId & other) const` *[inherited]*

For equality use `operator==(const RTRObjectId&)` for equality

To compare instance id with instance context `conformsTo(const RTRObjectId&)`

To compare class id with base class id

RTRPublicBooleanConfig - Assignment

Functions

- `RTRPublicBooleanConfig & RTRPublicBooleanConfig::operator= (RTRBOOL rhs)`

Function Documentation

`RTRPublicBooleanConfig & RTRPublicBooleanConfig::operator= (RTRBOOL rhs)` *[inline, inherited]*

Set the value to rhs. The [modifyEnabled\(\)](#) REQUIRE from the base class has been removed.

Reimplemented from [RTRManagedBooleanConfig](#).

RTRPublicBooleanConfig - Operations

Functions

- `void RTRPublicBooleanConfig::internalSet ()`
- `void RTRPublicBooleanConfig::internalClear ()`
- `void RTRPublicBooleanConfig::setStore ()`
- `void RTRPublicBooleanConfig::clearStore ()`

Function Documentation

`void RTRPublicBooleanConfig::internalSet ()` *[inline, inherited]*

Set the value to newVValue

`void RTRPublicBooleanConfig::internalClear ()` *[inline, inherited]*

Set the value to newVValue

`void RTRPublicBooleanConfig::setStore ()` *[inherited]*

Set the store value to RTRTRUE. The new value is not persistent.

`void RTRPublicBooleanConfig::clearStore ()` *[inherited]*

Set the store value to RTRFALSE. The new value is not persistent.

RTRPublicBoolean - Assignment

Functions

- [RTRPublicBoolean](#) & [RTRPublicBoolean::operator=](#) (RTRBOOL rhs)

Function Documentation

[RTRPublicBoolean](#) & [RTRPublicBoolean::operator=](#) (RTRBOOL *rhs*) [inline, inherited]

Set the current value to rhs, notify clients The [modifyEnabled\(\)](#) precondition in the base class is removed.

Reimplemented from [RTRManagedBoolean](#).

RTRPublicBoolean - Operations

Functions

- void [RTRPublicBoolean::internalSet](#) ()
- void [RTRPublicBoolean::internalClear](#) ()

Function Documentation

void [RTRPublicBoolean::internalSet](#) () [inline, inherited]

Set the value to RTRTRUE.

void [RTRPublicBoolean::internalClear](#) () [inline, inherited]

Set the value to RTRFALSE.

RTRPublicCounter - Operations

Functions

- virtual void [RTRPublicCounter::reset](#) ()
- void **operator+=** (unsigned long)
- [RTRPublicCounter](#) & **operator++** ()
- [RTRPublicCounter](#) & **operator++** (int)

Function Documentation

virtual void [RTRPublicCounter::reset](#) () [virtual, inherited]

Reset this counter to 0.

Implements [RTRManagedCounter](#).

RTRPublicGaugeConfig - Assignment

Functions

- [RTRPublicGaugeConfig](#) & [RTRPublicGaugeConfig::operator=](#) (long rhs)

Function Documentation

[RTRPublicGaugeConfig](#) & [RTRPublicGaugeConfig::operator=](#) (long *rhs*) [inline, inherited]

REQUIRE: (rhs >= [minValue\(\)](#)) || [modifyEnabled\(\)](#)

REQUIRE: (rhs <= [maxValue\(\)](#)) || [modifyEnabled\(\)](#)

ENSURE: [lowWaterMark\(\)](#) <= [value\(\)](#)

ENSURE: [highWaterMark\(\)](#) >= [value\(\)](#)

RTRPublicGaugeConfig - Operations

Functions

- void [RTRPublicGaugeConfig::operator+=](#) (long)
- void [RTRPublicGaugeConfig::operator-=](#) (long)
- [RTRPublicGaugeConfig](#) & [RTRPublicGaugeConfig::operator++](#) ()
- [RTRPublicGaugeConfig](#) & [RTRPublicGaugeConfig::operator++](#) (int)
- [RTRPublicGaugeConfig](#) & [RTRPublicGaugeConfig::operator--](#) ()
- [RTRPublicGaugeConfig](#) & [RTRPublicGaugeConfig::operator--](#) (int)
- void [RTRPublicGaugeConfig::set](#) (long newValue)
- void [RTRPublicGaugeConfig::set](#) (long newMin, long newMax, long newValue)
- void [RTRPublicGaugeConfig::internalSetRange](#) (long newMin, long newMax)
- void [RTRPublicGaugeConfig::setStore](#) (long newMin, long newMax)

Function Documentation

void RTRPublicGaugeConfig::operator+= (long) [inherited]

- **REQUIRE:** (newValue >= [minValue\(\)](#)) || [modifyEnabled\(\)](#)
- **REQUIRE:** (newValue <= [maxValue\(\)](#)) || [modifyEnabled\(\)](#)
- **ENSURE:** [lowWaterMark\(\)](#) <= [value\(\)](#)
- **ENSURE:** [highWaterMark\(\)](#) >= [value\(\)](#)

void RTRPublicGaugeConfig::operator-= (long) [inherited]

- REQUIRE:** (newValue >= [minValue\(\)](#)) || [modifyEnabled\(\)](#)
- REQUIRE:** (newValue <= [maxValue\(\)](#)) || [modifyEnabled\(\)](#)
- ENSURE:** [lowWaterMark\(\)](#) <= [value\(\)](#)
- ENSURE:** [highWaterMark\(\)](#) >= [value\(\)](#)

[RTRPublicGaugeConfig](#) & [RTRPublicGaugeConfig::operator++](#) () [inherited]

- REQUIRE:** (newValue <= [maxValue\(\)](#)) || [modifyEnabled\(\)](#)
- ENSURE:** [highWaterMark\(\)](#) >= [value\(\)](#)

[RTRPublicGaugeConfig](#) & [RTRPublicGaugeConfig::operator++](#) (int) [inherited]

- REQUIRE:** (newValue <= [maxValue\(\)](#)) || [modifyEnabled\(\)](#)
- ENSURE:** [highWaterMark\(\)](#) >= [value\(\)](#)

[RTRPublicGaugeConfig](#) & [RTRPublicGaugeConfig::operator--](#) () [inherited]

- REQUIRE:** (newValue >= [minValue\(\)](#)) || [modifyEnabled\(\)](#)

ENSURE: [lowWaterMark\(\)](#) <= [value\(\)](#)

[RTRPublicGaugeConfig](#) & **[RTRPublicGaugeConfig::operator--](#)** (int) [inherited]

REQUIRE: (newValue >= [minValue\(\)](#)) || [modifyEnabled\(\)](#)

ENSURE: [lowWaterMark\(\)](#) <= [value\(\)](#)

[void RTRPublicGaugeConfig::set](#) (long *newValue*) [inherited]

REQUIRE: (newValue >= [minValue\(\)](#)) || [modifyEnabled\(\)](#)

REQUIRE: (newValue <= [maxValue\(\)](#)) || [modifyEnabled\(\)](#)

ENSURE: [lowWaterMark\(\)](#) <= [value\(\)](#)

ENSURE: [highWaterMark\(\)](#) >= [value\(\)](#)

[void RTRPublicGaugeConfig::set](#) (long *newMin*, long *newMax*, long *newValue*) [inherited]

Set the minimum assignment value for this gauge to newMin, the maximum assignment value for this gauge to newMax, and the current value of this gauge to newValue.

REQUIRE: (newValue >= [minValue\(\)](#)) || [modifyEnabled\(\)](#)

REQUIRE: (newValue <= [maxValue\(\)](#)) || [modifyEnabled\(\)](#)

REQUIRE: newMin <= newMax

ENSURE: [lowWaterMark\(\)](#) <= [value\(\)](#)

ENSURE: [highWaterMark\(\)](#) >= [value\(\)](#)

[void RTRPublicGaugeConfig::internalSetRange](#) (long *newMin*, long *newMax*) [inherited]

Set the values for min and max. Clients are notified but the context is not notified.

REQUIRE: (newValue >= [minValue\(\)](#)) || [modifyEnabled\(\)](#)

REQUIRE: (newValue <= [maxValue\(\)](#)) || [modifyEnabled\(\)](#)

REQUIRE: newMin <= newMax

[void RTRPublicGaugeConfig::setStore](#) (long *newMin*, long *newMax*) [inherited]

REQUIRE: newMin <= newMax

RTRPublicGauge - Assignment

Functions

- [RTRPublicGauge](#) & [RTRPublicGauge::operator=](#) (long rhs)

Function Documentation

[RTRPublicGauge](#) & **[RTRPublicGauge::operator=](#)** (long *rhs*) [inline, inherited]

REQUIRE: (rhs >= [minValue\(\)](#)) || [modifyEnabled\(\)](#)

REQUIRE: (rhs <= [maxValue\(\)](#)) || [modifyEnabled\(\)](#)

ENSURE: [lowWaterMark\(\)](#) <= [value\(\)](#)

ENSURE: [highWaterMark\(\)](#) >= [value\(\)](#)

RTRPublicGauge - Operations

Functions

- [void RTRPublicGauge::operator+=](#) (long)

- void [RTRPublicGauge::operator-=](#) (long)
- [RTRPublicGauge](#) & [RTRPublicGauge::operator++](#) ()
- [RTRPublicGauge](#) & [RTRPublicGauge::operator++](#) (int)
- [RTRPublicGauge](#) & [RTRPublicGauge::operator--](#) ()
- [RTRPublicGauge](#) & [RTRPublicGauge::operator--](#) (int)
- void [RTRPublicGauge::set](#) (long newValue)
- void [RTRPublicGauge::set](#) (long newMin, long newMax, long newValue)
- virtual void [RTRPublicGauge::internalSetRange](#) (long newMin, long newMax)

Function Documentation

void RTRPublicGauge::operator+= (long) [inherited]
 REQUIRE: (newValue >= [minValue\(\)](#)) || [modifyEnabled\(\)](#)
 REQUIRE: (newValue <= [maxValue\(\)](#)) || [modifyEnabled\(\)](#)
 ENSURE: [lowWaterMark\(\)](#) <= [value\(\)](#)
 ENSURE: [highWaterMark\(\)](#) >= [value\(\)](#)

void RTRPublicGauge::operator-= (long) [inherited]
 REQUIRE: (newValue >= [minValue\(\)](#)) || [modifyEnabled\(\)](#)
 REQUIRE: (newValue <= [maxValue\(\)](#)) || [modifyEnabled\(\)](#)
 ENSURE: [lowWaterMark\(\)](#) <= [value\(\)](#)
 ENSURE: [highWaterMark\(\)](#) >= [value\(\)](#)

[RTRPublicGauge](#)& RTRPublicGauge::operator++ () [inherited]
 REQUIRE: (newValue <= [maxValue\(\)](#)) || [modifyEnabled\(\)](#)
 ENSURE: [highWaterMark\(\)](#) >= [value\(\)](#)

[RTRPublicGauge](#)& RTRPublicGauge::operator++ (int) [inherited]
 REQUIRE: (newValue <= [maxValue\(\)](#)) || [modifyEnabled\(\)](#)
 ENSURE: [highWaterMark\(\)](#) >= [value\(\)](#)

[RTRPublicGauge](#)& RTRPublicGauge::operator-- () [inherited]
 REQUIRE: (newValue >= [minValue\(\)](#)) || [modifyEnabled\(\)](#)
 ENSURE: [lowWaterMark\(\)](#) <= [value\(\)](#)

[RTRPublicGauge](#)& RTRPublicGauge::operator-- (int) [inherited]
 REQUIRE: (newValue >= [minValue\(\)](#)) || [modifyEnabled\(\)](#)
 ENSURE: [lowWaterMark\(\)](#) <= [value\(\)](#)

void RTRPublicGauge::set (long *newValue*) [inherited]
 REQUIRE: (newValue >= [minValue\(\)](#)) || [modifyEnabled\(\)](#)
 REQUIRE: (newValue <= [maxValue\(\)](#)) || [modifyEnabled\(\)](#)
 ENSURE: [lowWaterMark\(\)](#) <= [value\(\)](#)
 ENSURE: [highWaterMark\(\)](#) >= [value\(\)](#)

void RTRPublicGauge::set (long *newMin*, long *newMax*, long *newValue*) [inherited]

Set the minimum assignment value for this gauge to newMin, the maximum assignment value for this gauge to newMax, and the current value of this gauge to newValue.

REQUIRE: (newMin <= newValue) || [modifyEnabled\(\)](#)

REQUIRE: (newMax >= newValue) || [modifyEnabled\(\)](#)

REQUIRE: newMin <= newMax

ENSURE: [lowWaterMark\(\)](#) <= [value\(\)](#)

ENSURE: [highWaterMark\(\)](#) >= [value\(\)](#)

virtual void RTRPublicGauge::internalSetRange (long *newMin*, long *newMax*) [virtual, inherited]

Set the minimum assignment for this gauge to newMin and the maximum assignment value for this gauge to newMax.

REQUIRE: (newMin <= [value\(\)](#)) || [modifyEnabled\(\)](#)

REQUIRE: (newMax >= [value\(\)](#)) || [modifyEnabled\(\)](#)

REQUIRE: newMin <= newMax

RTRPublicLargeNumeric - Assignment

Functions

- [RTRPublicLargeNumeric](#) & **operator=** (RTR_I64)

RTRPublicLargeNumeric - Operations

Functions

- void **operator+=** (RTR_I64)
- void **operator-=** (RTR_I64)
- [RTRPublicLargeNumeric](#) & **operator++** ()
- [RTRPublicLargeNumeric](#) & **operator++** (int)
- [RTRPublicLargeNumeric](#) & **operator--** ()
- [RTRPublicLargeNumeric](#) & **operator--** (int)
- void **set** (RTR_I64)

RTRPublicNumericConfig - Assignment

Functions

- [RTRPublicNumericConfig](#) & [RTRPublicNumericConfig::operator=](#) (long rhs)

Function Documentation

[RTRPublicNumericConfig](#) & RTRPublicNumericConfig::operator= (long *rhs*) [inline, inherited]

Set the active value to newValue. Note: The [modifyEnabled\(\)](#) REQUIRE from the base class is removed here.

REQUIRE: newValue >= [minValue\(\)](#)

REQUIRE: newValue <= [maxValue\(\)](#)

Synchronized

Re-implemented from [RTRManagedNumericConfig](#).

RTRPublicNumericConfig - Operations

Functions

- void [RTRPublicNumericConfig::internalSet](#) (long newValue)
- void [RTRPublicNumericConfig::setStore](#) (long newStore)

Function Documentation

void RTRPublicNumericConfig::internalSet (long *newValue*) [inline, inherited]

Set the active value to newValue. It does not notify the context MO.

REQUIRE: newValue >= [minValue\(\)](#)

REQUIRE: newValue <= [maxValue\(\)](#)

Synchronized

void RTRPublicNumericConfig::setStore (long *newStore*) [inherited]

Set the store value to newStore. The value is not persistent. Synchronized

RTRPublicNumericRange - Assignment

Functions

- [RTRPublicNumericRange](#) & [RTRPublicNumericRange::operator=](#) (long rhs)

Function Documentation

[RTRPublicNumericRange](#) & RTRPublicNumericRange::operator= (long *rhs*) [inline, inherited]

Sets the current value to rhs.

REQUIRE: rhs >= [minValue\(\)](#)

REQUIRE: rhs <= [maxValue\(\)](#)

Reimplemented from [RTRManagedNumericRange](#).

RTRPublicNumericRange - Operations

Functions

- void [RTRPublicNumericRange::set](#) (long newValue)
- void [RTRPublicNumericRange::internalSet](#) (long newValue)
- void [RTRPublicNumericRange::set](#) (long newMin, long newMax, long newValue)

Function Documentation

void RTRPublicNumericRange::set (long *newValue*) [inline, virtual, inherited]

Sets the current value of this parameter to newValue.

REQUIRE: modifyEnabled()

REQUIRE: newValue >= [minValue\(\)](#)

REQUIRE: newValue <= [maxValue\(\)](#)

Reimplemented from [RTRManagedNumericRange](#). `void RTRPublicNumericRange::internalSet (long newValue)` [*inline*, *inherited*]

Sets the current value of this parameter to *newValue*.

REQUIRE: *newValue* >= [minValue\(\)](#)

REQUIRE: *newValue* <= [maxValue\(\)](#)

`void RTRPublicNumericRange::set (long newMin, long newMax, long newValue)` [*inherited*]

Sets the minimum assignment value for this parameter to *newMin*, the maximum assignment value for this parameter to *newMax*, and the current value of this parameter to *newValue*

REQUIRE: *newValue* >= *newMin*

REQUIRE: *newValue* <= *newMax*

REQUIRE: *newMin* <= *newMax*

RTRPublicNumeric - Assignment

Functions

- [RTRPublicNumeric](#) & `operator=` (long)

RTRPublicNumeric - Operations

Functions

- void `operator+=` (long)
- void `operator-=` (long)
- [RTRPublicNumeric](#) & `operator++` ()
- [RTRPublicNumeric](#) & `operator++` (int)
- [RTRPublicNumeric](#) & `operator--` ()
- [RTRPublicNumeric](#) & `operator--` (int)
- void `set` (long)

RTRPublicObject - Operations

Functions

- void [RTRPublicObject::markNormal](#) (const char *)
- void [RTRPublicObject::markRecovering](#) (const char *)
- void [RTRPublicObject::markWaiting](#) (const char *)
- void [RTRPublicObject::markDead](#) (const char *)
- void [RTRPublicObject::indicateInfo](#) (const char *)

Function Documentation

`void RTRPublicObject::markNormal (const char *)` [*inherited*]

Mark this object normal and notify clients;

`void RTRPublicObject::markRecovering (const char *)` [*inherited*]

Mark this object recovering and notify clients;

void RTRPublicObject::markWaiting (const char *) [inherited]
Mark this object waiting and notify clients;

void RTRPublicObject::markDead (const char *) [inherited]
Mark this object dead and notify clients;

void RTRPublicObject::indicateInfo (const char *) [inherited]
Notify clients of change in text.

RTRProxyManagedObjectClassDirectory - Attributes

Functions

- const [RTRObjectId](#) & **classFilter** () const
- const [RTRProxyManagedObjectServerPool](#) & **serverPool** () const

RTRProxyManagedObjectClassDirectory - Query

Functions

- RTRBOOL [RTRProxyManagedObjectClassDirectory::hasHandle](#) (const [RTRObjectId](#) &) const
- RTRBOOL [RTRProxyManagedObjectClassDirectory::hasHandle](#) (const [RTRProxyManagedObjectHandle](#) &) const

Function Documentation

RTRBOOL RTRProxyManagedObjectClassDirectory::hasHandle (const [RTRObjectId](#) &) const [inherited]
Does this directory contain a handle which has the given instance identifiers?

RTRBOOL RTRProxyManagedObjectClassDirectory::hasHandle (const [RTRProxyManagedObjectHandle](#) &) const [inherited]
Does this directory contain the given handle?

RTRProxyManagedObjectClassDirectory - Access Randomly

Functions

- const [RTRProxyManagedObjectHandle](#) * [RTRProxyManagedObjectClassDirectory::handle](#) (const [RTRObjectId](#) &iid) const

Function Documentation

const [RTRProxyManagedObjectHandle](#) * RTRProxyManagedObjectClassDirectory::handle (const [RTRObjectId](#) & iid) const [inherited]
The handle, if any, which has the given instance identifier.
ENSURE: (result == null) == hasHandle(iid)

RTRProxyManagedObjectClassDirectory - Access Sequentially

Functions

- [RTRProxyManagedObjectHandleIterator](#) [RTRProxyManagedObjectClassDirectory::handles](#) () const

Function Documentation

[RTRProxyManagedObjectHandleIterator](#) `RTRProxyManagedObjectClassDirectory::handles () const` [inherited]

An iterator providing sequential access to the handles in this directory.

RTRProxyManagedObjectClassDirectory - Event processing from RTRProxyManagedObjectServerPoolClient

Functions

- virtual void `RTRProxyManagedObjectClassDirectory::processProxyManagedObjectServerAdded` (`RTRProxyManagedObjectServerPool &`, `RTRProxyManagedObjectServer &`)

Function Documentation

virtual void `RTRProxyManagedObjectClassDirectory::processProxyManagedObjectServerAdded` ([RTRProxyManagedObjectServerPool](#) &, [RTRProxyManagedObjectServer](#) &) [virtual, inherited]

The given server has been added to the pool.

Reimplemented from [RTRProxyManagedObjectServerPoolClient](#).

RTRProxyManagedObjectClassDirectory - Event client management

Functions

- RTRBOOL [RTRProxyManagedObjectClassDirectory::hasClient](#) (const [RTRProxyManagedObjectClassDirectoryClient](#) &) const
- void [RTRProxyManagedObjectClassDirectory::addClient](#) (const [RTRProxyManagedObjectClassDirectoryClient](#) &client)
- void `RTRProxyManagedObjectClassDirectory::dropClient` (const `RTRProxyManagedObjectClassDirectoryClient` &client)

Function Documentation

RTRBOOL `RTRProxyManagedObjectClassDirectory::hasClient` (const [RTRProxyManagedObjectClassDirectoryClient](#) &) const [inherited]

Is the given client registered to receive update events from this directory?

void `RTRProxyManagedObjectClassDirectory::addClient` (const [RTRProxyManagedObjectClassDirectoryClient](#) &client) [inherited]

Register the given client to receive update events from this directory.

REQUIRE: !hasClient(client)

REQUIRE: hasClient(client)

void `RTRProxyManagedObjectClassDirectory::dropClient` (const [RTRProxyManagedObjectClassDirectoryClient](#) &client) [inherited]

Un-register the given client to receive update events from this directory.

ENSURE: !hasClient(client)

RTRProxyManagedObjectClassDirectoryClient - Event processing

Functions

- virtual void `RTRProxyManagedObjectClassDirectoryClient::processDirectoryHandleAdded` (`RTRProxyManagedObjectClassDirectory &`, `RTRProxyManagedObjectServer &`, const `RTRProxyManagedObjectHandle` &)=0

- virtual void RTRProxyManagedObjectClassDirectoryClient::processDirectoryHandleRemoved (RTRProxyManagedObjectClassDirectory &, RTRProxyManagedObjectServer &, const RTRProxyManagedObjectHandle &)=0

Function Documentation

virtual void RTRProxyManagedObjectClassDirectoryClient::processDirectoryHandleAdded ([RTRProxyManagedObjectClassDirectory](#) &, [RTRProxyManagedObjectServer](#) &, const [RTRProxyManagedObjectHandle](#) &) [pure virtual, inherited]

The handle has been added to the given directory.

Implemented in [RTRProxyManagedObjectPool](#). virtual void RTRProxyManagedObjectClassDirectoryClient::processDirectoryHandleRemoved ([RTRProxyManagedObjectClassDirectory](#) &, [RTRProxyManagedObjectServer](#) &, const [RTRProxyManagedObjectHandle](#) &) [pure virtual, inherited]

The handle has been removed from the given directory.

Implemented in [RTRProxyManagedObjectPool](#).

RTRProxyManagedObjectPool - Query

Functions

- RTRBOOL [RTRProxyManagedObjectPool::hasObject](#) (const [RTRObjectId](#) &) const

Function Documentation

RTRBOOL RTRProxyManagedObjectPool::hasObject (const [RTRObjectId](#) &) const [inherited]

Does this pool contain an object with the given identifier?

RTRProxyManagedObjectPool - Attributes

Functions

- [RTRProxyManagedObjectClassDirectory](#) & directory () const

RTRProxyManagedObjectPool - Access Randomly

Functions

- RTRProxyManagedObjectPtr [RTRProxyManagedObjectPool::object](#) (const [RTRObjectId](#) &) const

Function Documentation

RTRProxyManagedObjectPtr RTRProxyManagedObjectPool::object (const [RTRObjectId](#) &) const [inherited]

The object, if any, with the given identifier;

RTRProxyManagedObjectPool - Access Sequentially

Functions

- RTRLinkedListCursor< RTRProxyManagedObjectPtr > [RTRProxyManagedObjectPool::objects](#) () const

Function Documentation

RTRLinkedListCursor<RTRProxyManagedObjectPtr> RTRProxyManagedObjectPool::objects () const [inherited]

An iterator providing sequential access to the objects in this pool.

RTRProxyManagedObjectPool - Event processing

Functions

- virtual void [RTRProxyManagedObjectPool::processDirectoryHandleAdded](#) ([RTRProxyManagedObjectClassDirectory](#) &, [RTRProxyManagedObjectServer](#) &, const [RTRProxyManagedObjectHandle](#) &)
- virtual void [RTRProxyManagedObjectPool::processDirectoryHandleRemoved](#) ([RTRProxyManagedObjectClassDirectory](#) &, [RTRProxyManagedObjectServer](#) &, const [RTRProxyManagedObjectHandle](#) &)

Function Documentation

virtual void [RTRProxyManagedObjectPool::processDirectoryHandleAdded](#) ([RTRProxyManagedObjectClassDirectory](#) &, [RTRProxyManagedObjectServer](#) &, const [RTRProxyManagedObjectHandle](#) &) [virtual, inherited]

The handle has been added to the given directory.

Implements [RTRProxyManagedObjectClassDirectoryClient](#).virtual void [RTRProxyManagedObjectPool::processDirectoryHandleRemoved](#) ([RTRProxyManagedObjectClassDirectory](#) &, [RTRProxyManagedObjectServer](#) &, const [RTRProxyManagedObjectHandle](#) &) [virtual, inherited]

The handle has been removed from the given directory.

Implements [RTRProxyManagedObjectClassDirectoryClient](#).

RTRProxyManagedObjectPool - Event client management

Functions

- RTRBOOL [RTRProxyManagedObjectPool::hasClient](#) (const [RTRProxyManagedObjectPoolClient](#) &) const
- void [RTRProxyManagedObjectPool::addClient](#) (const [RTRProxyManagedObjectPoolClient](#) &client)
- void [RTRProxyManagedObjectPool::dropClient](#) (const [RTRProxyManagedObjectPoolClient](#) &client)

Function Documentation

RTRBOOL [RTRProxyManagedObjectPool::hasClient](#) (const [RTRProxyManagedObjectPoolClient](#) &) const [inherited]

Is the given client registered to receive update events from this pool?

void [RTRProxyManagedObjectPool::addClient](#) (const [RTRProxyManagedObjectPoolClient](#) & client) [inherited]

Register the given client to receive update events from this pool.

REQUIRE: !hasClient(client)

REQUIRE: hasClient(client)

void [RTRProxyManagedObjectPool::dropClient](#) (const [RTRProxyManagedObjectPoolClient](#) & client) [inherited]

Un-register the given client to receive update events from this pool.

ENSURE: !hasClient(client)

RTRProxyManagedObjectPoolClient - Event processing

Functions

- virtual void [RTRProxyManagedObjectPoolClient::processProxyManagedObjectAdded](#) ([RTRProxyManagedObjectPool](#) &, [RTRProxyManagedObject](#) &)=0
- virtual void [RTRProxyManagedObjectPoolClient::processProxyManagedObjectRemoved](#) ([RTRProxyManagedObjectPool](#) &, [RTRProxyManagedObject](#) &)=0

Function Documentation

virtual void RTRProxyManagedObjectPoolClient::processProxyManagedObjectAdded ([RTRProxyManagedObjectPool](#) &, [RTRProxyManagedObject](#) &) [pure virtual, inherited]

A new object has been added to the given pool.

virtual void RTRProxyManagedObjectPoolClient::processProxyManagedObjectRemoved ([RTRProxyManagedObjectPool](#) &, [RTRProxyManagedObject](#) &) [pure virtual, inherited]

An object has been removed from the given pool.

RTRProxyManagedObjectServerClient - Event processing

Functions

- **virtual void RTRProxyManagedObjectServerClient::processObjectServerError** ([RTRProxyManagedObjectServer](#) &)
- **virtual void RTRProxyManagedObjectServerClient::processObjectServerSync** ([RTRProxyManagedObjectServer](#) &)
- **virtual void RTRProxyManagedObjectServerClient::processObjectServerRootAdded** ([RTRProxyManagedObjectServer](#) &, const [RTRProxyManagedObjectHandle](#) &)
- **virtual void RTRProxyManagedObjectServerClient::processObjectServerRootRemoved** ([RTRProxyManagedObjectServer](#) &, const [RTRProxyManagedObjectHandle](#) &)

Function Documentation

virtual void RTRProxyManagedObjectServerClient::processObjectServerError ([RTRProxyManagedObjectServer](#) &) [virtual, inherited]

The given server has transitioned into an unrecoverable error state.

virtual void RTRProxyManagedObjectServerClient::processObjectServerSync ([RTRProxyManagedObjectServer](#) &) [virtual, inherited]

The given server has transitioned into the Sync state.

virtual void RTRProxyManagedObjectServerClient::processObjectServerRootAdded ([RTRProxyManagedObjectServer](#) &, const [RTRProxyManagedObjectHandle](#) &) [virtual, inherited]

The given root proxy managed object (handle) has been added to the server.

virtual void RTRProxyManagedObjectServerClient::processObjectServerRootRemoved ([RTRProxyManagedObjectServer](#) &, const [RTRProxyManagedObjectHandle](#) &) [virtual, inherited]

The given root proxy managed object (handle) has been removed from the server.

RTRProxyManagedObjectServerPool - Access Sequentially

Functions

- **RTRLLinkedListCursor<** [RTRProxyManagedObjectServer](#) **>** [RTRProxyManagedObjectServerPool::servers](#) () const

Function Documentation

RTRLLinkedListCursor< [RTRProxyManagedObjectServer](#) **>** [RTRProxyManagedObjectServerPool::servers](#) () const [inherited]

A cursor providing sequential access to the servers in this pool.

RTRProxyManagedObjectServerPool - Event client management

Functions

- RTRBOOL [RTRProxyManagedObjectServerPool::hasClient](#) (const [RTRProxyManagedObjectServerPoolClient](#) &) const
- void [RTRProxyManagedObjectServerPool::addClient](#) (const [RTRProxyManagedObjectServerPoolClient](#) &client)
- void [RTRProxyManagedObjectServerPool::dropClient](#) (const [RTRProxyManagedObjectServerPoolClient](#) &client)

Function Documentation

RTRBOOL [RTRProxyManagedObjectServerPool::hasClient](#) (const [RTRProxyManagedObjectServerPoolClient](#) &) const
[inherited]

Is the given client registered to receive update events from this pool?

void [RTRProxyManagedObjectServerPool::addClient](#) (const [RTRProxyManagedObjectServerPoolClient](#) & client)
[inherited]

Register the given client to receive update events from this pool.

REQUIRE: !hasClient(client)

REQUIRE: hasClient(client)

void [RTRProxyManagedObjectServerPool::dropClient](#) (const [RTRProxyManagedObjectServerPoolClient](#) & client)
[inherited]

Un-register the given client to receive update events from this pool.

ENSURE: !hasClient(client)

RTRProxyManagedObjectServerPoolClient - Event processing

Functions

- virtual void [RTRProxyManagedObjectServerPoolClient::processProxyManagedObjectServerAdded](#) (RTRProxyManagedObjectServerPool &, RTRProxyManagedObjectServer &)
- virtual void [RTRProxyManagedObjectServerPoolClient::processProxyManagedObjectServerRemoved](#) (RTRProxyManagedObjectServerPool &, RTRProxyManagedObjectServer &)

Function Documentation

virtual void [RTRProxyManagedObjectServerPoolClient::processProxyManagedObjectServerAdded](#) ([RTRProxyManagedObjectServerPool](#) &, [RTRProxyManagedObjectServer](#) &) [virtual, inherited]

The given server has been added to the pool.

Reimplemented in [RTRProxyManagedObjectClassDirectory](#). virtual void [RTRProxyManagedObjectServerPoolClient::processProxyManagedObjectServerRemoved](#) ([RTRProxyManagedObjectServerPool](#) &, [RTRProxyManagedObjectServer](#) &) [virtual, inherited]

The given server is being removed from the pool.

RTRPublicStringConfig - Assignment

Functions

- [RTRPublicStringConfig](#) & [RTRPublicStringConfig::operator=](#) (const char *rhs)

Function Documentation

[RTRPublicStringConfig](#) & **[RTRPublicStringConfig::operator=](#)** (const char * *rhs*) [inline, inherited]

Set the active value.

Note: The [modifyEnabled\(\)](#) REQUIRE from the base class has been removed.

Synchronized

Reimplemented from [RTRManagedStringConfig](#).

RTRPublicStringConfig - Operations

Functions

- virtual void [RTRPublicStringConfig::set](#) (const char *newValue)
- virtual void [RTRPublicStringConfig::internalSet](#) (const char *newValue)
- void [RTRPublicStringConfig::setStore](#) (const char *newStore)

Function Documentation

virtual void [RTRPublicStringConfig::set](#) (const char * *newValue*) [virtual, inherited]

Sets the current value to newValue. Notify the context managed object.

Synchronized

Reimplemented from [RTRManagedString](#). **virtual void [RTRPublicStringConfig::internalSet](#)** (const char * *newValue*) [virtual, inherited]

Set the active value. Does not notify the context managed object.

Synchronized

void [RTRPublicStringConfig::setStore](#) (const char * *newStore*) [inherited]

Set the store value.

Synchronized

RTRPublicString - Assignment

Functions

- [RTRPublicString](#) & [RTRPublicString::operator=](#) (const char *rhs)

Function Documentation

[RTRPublicString](#) & **[RTRPublicString::operator=](#)** (const char * *rhs*) [inline, inherited]

Set the current value to rhs.

Note: The [modifyEnabled\(\)](#) REQUIRE from the base class has been removed.

Synchronized

Reimplemented from [RTRManagedString](#).

RTRPublicString - Operations

Functions

- void [RTRPublicString::set](#) (const char *newValue)
- void [RTRPublicString::internalSet](#) (const char *newValue)

Function Documentation

void **RTRPublicString::set** (const char * *newValue*) [virtual, inherited]

Sets the current value to newValue. Notify the context (managed object)

Synchronized

Reimplemented from [RTRManagedString](#).void **RTRPublicString::internalSet** (const char * *newValue*) [inherited]

Sets the current value to newValue.

Synchronized

RTRProxyManagedObjectHandle - Attributes

Functions

- const [RTRObjectId](#) & [RTRProxyManagedObjectHandle::classId](#) () const
- const [RTRObjectId](#) & [RTRProxyManagedObjectHandle::instanceId](#) () const
- const [RTRString](#) & [RTRProxyManagedObjectHandle::name](#) () const

Function Documentation

const [RTRObjectId](#) & **RTRProxyManagedObjectHandle::classId** () const [inline, inherited]

The class identifier of a proxy managed object.

const [RTRObjectId](#) & **RTRProxyManagedObjectHandle::instanceId** () const [inline, inherited]

The instance identifier of a proxy managed object.

const [RTRString](#) & **RTRProxyManagedObjectHandle::name** () const [inline, inherited]

The name of a proxy managed object.

RTRProxyManagedVariableHandle - Attributes

Functions

- const [RTRString](#) & [RTRProxyManagedVariableHandle::name](#) () const
- MVType [RTRProxyManagedVariableHandle::type](#) () const
- [RTRString](#) [RTRProxyManagedVariableHandle::typeString](#) () const

Function Documentation

const [RTRString](#) & **RTRProxyManagedVariableHandle::name** () const [inline, inherited]

The name of the variable.

[RTRProxyManagedVariableHandle::MVType](#) **RTRProxyManagedVariableHandle::type** () const [inline, inherited]

The variable type.

[RTRString](#) **RTRProxyManagedVariableHandle::typeString** () const [inline, inherited]

The variable type represented as a string.

RTRProxyManagedObjectHandleIterator - Attributes

Functions

- int [RTRProxyManagedObjectHandleIterator::count](#) () const

Function Documentation

int RTRProxyManagedObjectHandleIterator::count () const [inherited]

The number of children available via this iterator.

RTRProxyManagedObjectHandleIterator - State

Functions

- RTRBOOL [RTRProxyManagedObjectHandleIterator::off](#) () const
- RTRBOOL [RTRProxyManagedObjectHandleIterator::empty](#) () const

Function Documentation

RTRBOOL RTRProxyManagedObjectHandleIterator::off () const [inherited]

Is this iteration complete?

RTRBOOL RTRProxyManagedObjectHandleIterator::empty () const [inherited]

Are there no children available via this iterator?

ENSURE: result implies [count\(\)](#) == 0

RTRProxyManagedObjectHandleIterator - Access

Functions

- [RTRProxyManagedObjectHandle](#) & [RTRProxyManagedObjectHandleIterator::item](#) () const

Function Documentation

[RTRProxyManagedObjectHandle](#) & RTRProxyManagedObjectHandleIterator::item () const [inherited]

The current item in the current iteration.

RTRProxyManagedObjectHandleIterator - Operations

Functions

- void RTRProxyManagedObjectHandleIterator::start ()
- void RTRProxyManagedObjectHandleIterator::finish ()
- void RTRProxyManagedObjectHandleIterator::forth ()
- void [RTRProxyManagedObjectHandleIterator::back](#) ()

Function Documentation

void RTRProxyManagedObjectHandleIterator::start () [inherited]

Start a new iteration

ENSURE: [off\(\)](#) implies [empty\(\)](#)

void RTRProxyManagedObjectHandleIterator::finish () [inherited]

Start an iteration from the last available child.

ENSURE: [off\(\)](#) implies [empty\(\)](#)

void RTRProxyManagedObjectHandleIterator::forth () [inherited]

Continue the current iteration from [start\(\)](#) to [finish\(\)](#).

REQUIRE: !off()

void RTRProxyManagedObjectHandleIterator::back () [inherited]

Continue the current iteration from [finish\(\)](#) to [start\(\)](#).

REQUIRE: !off()

RTRProxyManagedVarHandleIterator - Attributes

Functions

- int [RTRProxyManagedVarHandleIterator::count \(\)](#) const

Function Documentation

int RTRProxyManagedVarHandleIterator::count () const [inherited]

The number of variables available via this iterator.

RTRProxyManagedVarHandleIterator - State

Functions

- RTRBOOL [RTRProxyManagedVarHandleIterator::off \(\)](#) const
- RTRBOOL [RTRProxyManagedVarHandleIterator::empty \(\)](#) const

Function Documentation

RTRBOOL RTRProxyManagedVarHandleIterator::off () const [inherited]

Is this iteration complete?

RTRBOOL RTRProxyManagedVarHandleIterator::empty () const [inherited]

Are there no variable available via this iterator?

ENSURE: result implies [count\(\)](#) == 0

RTRProxyManagedVarHandleIterator - Access

Functions

- [RTRProxyManagedVariableHandle](#) & [RTRProxyManagedVarHandleIterator::item \(\)](#) const

Function Documentation

[RTRProxyManagedVariableHandle](#) & RTRProxyManagedVarHandleIterator::item () const [inherited]

The current item in the current iteration.

RTRProxyManagedVarHandleIterator - Operations

Functions

- void [RTRProxyManagedVarHandleIterator::start](#) ()
- void [RTRProxyManagedVarHandleIterator::finish](#) ()
- void [RTRProxyManagedVarHandleIterator::forth](#) ()
- void [RTRProxyManagedVarHandleIterator::back](#) ()

Function Documentation

void RTRProxyManagedVarHandleIterator::start () [inherited]

Start a new iteration

ENSURE: [off\(\)](#) implies [empty\(\)](#)

void RTRProxyManagedVarHandleIterator::finish () [inherited]

Start an iteration from the last available variable.

ENSURE: [off\(\)](#) implies [empty\(\)](#)

void RTRProxyManagedVarHandleIterator::forth () [inherited]

Continue the current iteration from [start\(\)](#) to end.

REQUIRE: ![off\(\)](#)

void RTRProxyManagedVarHandleIterator::back () [inherited]

Continue the current iteration from end to [start\(\)](#).

REQUIRE: ![off\(\)](#)

RTRProxyManagedBoolean - Comparison

Functions

- RTRBOOL [RTRProxyManagedBoolean::operator==](#) (RTRBOOL rhs) const

Function Documentation

RTRBOOL RTRProxyManagedBoolean::operator== (RTRBOOL rhs) const [inline, inherited]

REQUIRE: [inSync\(\)](#) && ![error\(\)](#)

RTRProxyManagedBoolean - Access

Functions

- RTRBOOL [RTRProxyManagedBoolean::value](#) () const

Function Documentation

RTRBOOL RTRProxyManagedBoolean::value () const [inline, inherited]

REQUIRE: [inSync\(\)](#) && ![error\(\)](#)

RTRProxyManagedBoolean - Transformation

Functions

- virtual [RTRString RTRProxyManagedBoolean::toString \(\)](#) const
- RTRProxyManagedBoolean::operator RTRProxyManagedBooleanConfig & ()
- RTRProxyManagedBoolean::operator const RTRProxyManagedBooleanConfig & () const

Function Documentation

virtual [RTRString RTRProxyManagedBoolean::toString \(\)](#) const [virtual, inherited]

The value of this variable represented as a string.

REQUIRE: [inSync\(\)](#) && !error()

Implements [RTRProxyManagedVariable.RTRProxyManagedBoolean::operator RTRProxyManagedBooleanConfig & \(\)](#) [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::BooleanConfig

Reimplemented from [RTRProxyManagedVariable.RTRProxyManagedBoolean::operator const RTRProxyManagedBooleanConfig & \(\)](#) const [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::BooleanConfig

Reimplemented from [RTRProxyManagedVariable](#).

RTRProxyManagedBoolean - Attributes

Functions

- RTRBOOL [RTRProxyManagedBoolean::modifyEnabled \(\)](#) const

Function Documentation

RTRBOOL [RTRProxyManagedBoolean::modifyEnabled \(\)](#) const [inline, inherited]

Is the consumer permitted to modify this variable?

REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedBoolean - Assignment

Functions

- [RTRProxyManagedBoolean & RTRProxyManagedBoolean::operator=](#) (RTRBOOL rhs)

Function Documentation

[RTRProxyManagedBoolean & RTRProxyManagedBoolean::operator=](#) (RTRBOOL *rhs*) [inline, inherited]

REQUIRE: [inSync\(\)](#) && !error()

REQUIRE: [modifyEnabled\(\)](#)

Reimplemented in [RTRProxyManagedBooleanConfig](#).

RTRProxyManagedBoolean - Operations

Functions

- virtual void [RTRProxyManagedBoolean::set](#) ()=0
- virtual void [RTRProxyManagedBoolean::clear](#) ()=0

Function Documentation

virtual void [RTRProxyManagedBoolean::set](#) () [pure virtual, inherited]

REQUIRE: [inSync\(\)](#) && !error()

REQUIRE: [modifyEnabled\(\)](#)

virtual void [RTRProxyManagedBoolean::clear](#) () [pure virtual, inherited]

REQUIRE: [inSync\(\)](#) && !error()

REQUIRE: [modifyEnabled\(\)](#)

RTRProxyManagedBooleanConfig - Assignment

Functions

- [RTRProxyManagedBooleanConfig](#) & [RTRProxyManagedBooleanConfig::operator=](#) (RTRBOOL rhs)

Function Documentation

[RTRProxyManagedBooleanConfig](#) & [RTRProxyManagedBooleanConfig::operator=](#) (RTRBOOL *rhs*) [inline, inherited]

Set the current value to rhs.

REQUIRE: [inSync\(\)](#) && !error()

REQUIRE: [modifyEnabled\(\)](#)

Reimplemented from [RTRProxyManagedBoolean](#).

RTRProxyManagedBooleanConfig - Attributes

Functions

- RTRBOOL [RTRProxyManagedBooleanConfig::activeValue](#) () const
- RTRBOOL [RTRProxyManagedBooleanConfig::storeValue](#) () const
- RTRBOOL [RTRProxyManagedBooleanConfig::factoryDefault](#) () const

Function Documentation

RTRBOOL [RTRProxyManagedBooleanConfig::activeValue](#) () const [inline, inherited]

A synonym for [value\(\)](#).

REQUIRE: [inSync\(\)](#) && !error()

RTRBOOL [RTRProxyManagedBooleanConfig::storeValue](#) () const [inline, inherited]

The store value.

REQUIRE: [inSync\(\)](#) && !error()

RTRBOOL [RTRProxyManagedBooleanConfig::factoryDefault](#) () const [inline, inherited]

The factory default value.

REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedCounter - Comparison

Functions

- RTRBOOL [RTRProxyManagedCounter::operator==](#) (unsigned long rhs) const

Function Documentation

RTRBOOL RTRProxyManagedCounter::operator== (unsigned long *rhs*) const [inline, inherited]

REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedCounter - Access

Functions

- unsigned long [RTRProxyManagedCounter::value](#) () const

Function Documentation

unsigned long RTRProxyManagedCounter::value () const [inline, inherited]

The current value of this variable.

REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedCounter - Transformation

Functions

- virtual [RTRString RTRProxyManagedCounter::toString](#) () const
- RTRProxyManagedCounter::operator unsigned long () const

Function Documentation

virtual [RTRString](#) RTRProxyManagedCounter::toString () const [virtual, inherited]

The value of this variable represented as a string.

REQUIRE: [inSync\(\)](#) && !error()

Implements [RTRProxyManagedVariable](#).RTRProxyManagedCounter::operator unsigned long () const [inline, inherited]

This variable as an unsigned long.

REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedCounter - Operations

Functions

- virtual void [RTRProxyManagedCounter::reset](#) ()=0

Function Documentation

virtual void RTRProxyManagedCounter::reset () [pure virtual, inherited]

REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedGauge - Transformation

Functions

- `RTRProxyManagedGauge::operator RTRProxyManagedGaugeConfig & ()`
- `RTRProxyManagedGauge::operator const RTRProxyManagedGaugeConfig & () const`

Function Documentation

RTRProxyManagedGauge::operator [RTRProxyManagedGaugeConfig](#) & () [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::GaugeConfig

Reimplemented from [RTRProxyManagedNumeric](#). **RTRProxyManagedGauge::operator const [RTRProxyManagedGaugeConfig](#) & () const** [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::GaugeConfig

Reimplemented from [RTRProxyManagedNumeric](#).

RTRProxyManagedGauge - Attributes

Functions

- `long RTRProxyManagedGauge::minValue () const`
- `long RTRProxyManagedGauge::maxValue () const`
- `long RTRProxyManagedGauge::lowWaterMark () const`
- `long RTRProxyManagedGauge::highWaterMark () const`
- `RTRBOOL RTRProxyManagedGauge::modifyEnabled () const`

Function Documentation

long [RTRProxyManagedGauge::minValue](#) () const [inline, inherited]

The minimum value which may be assigned to this variable.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: [minValue\(\)](#) <= [maxValue\(\)](#)

ENSURE: ([minValue\(\)](#) <= [value\(\)](#)) || [modifyEnabled\(\)](#)

ENSURE: ([minValue\(\)](#) <= [highWaterMark\(\)](#)) || [modifyEnabled\(\)](#)

long [RTRProxyManagedGauge::maxValue](#) () const [inline, inherited]

The maximum value which may be assigned to this variable.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: [maxValue\(\)](#) >= [minValue\(\)](#)

ENSURE: ([maxValue\(\)](#) >= [value\(\)](#)) || [modifyEnabled\(\)](#)

ENSURE: ([maxValue\(\)](#) >= [lowWaterMark\(\)](#)) || [modifyEnabled\(\)](#)

long [RTRProxyManagedGauge::lowWaterMark](#) () const [inline, inherited]

The lowest value assumed by this gauge since its creation.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: [lowWaterMark\(\)](#) <= [highWaterMark\(\)](#)

ENSURE: [lowWaterMark\(\)](#) <= [value\(\)](#)

ENSURE: ([lowWaterMark\(\)](#) <= [maxValue\(\)](#)) || [modifyEnabled\(\)](#)

long RTRProxyManagedGauge::highWaterMark () const [inline, inherited]

The highest value assumed by this gauge since its creation.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: [highWaterMark\(\)](#) >= [lowWaterMark\(\)](#)

ENSURE: [highWaterMark\(\)](#) >= [value\(\)](#)

ENSURE: ([highWaterMark\(\)](#) >= [minValue\(\)](#)) || [modifyEnabled\(\)](#)

RTRBOOL RTRProxyManagedGauge::modifyEnabled () const [inline, inherited]

Is the consumer permitted to modify this variable?

REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedGauge - Operations

Functions

- virtual void [RTRProxyManagedGauge::setRange](#) (long newMin, long newMax)=0

Function Documentation

virtual void RTRProxyManagedGauge::setRange (long newMin, long newMax) [pure virtual, inherited]

Sets the min and max values.

REQUIRE: [inSync\(\)](#) && !error()

REQUIRE: [modifyEnabled\(\)](#)

REQUIRE: newMin <= newMax

RTRProxyManagedGaugeConfig - Attributes

Functions

- long [RTRProxyManagedGaugeConfig::minStoreValue](#) () const
- long [RTRProxyManagedGaugeConfig::minFactoryDefault](#) () const
- long [RTRProxyManagedGaugeConfig::maxStoreValue](#) () const
- long [RTRProxyManagedGaugeConfig::maxFactoryDefault](#) () const

Function Documentation

long RTRProxyManagedGaugeConfig::minStoreValue () const [inline, inherited]

The minimum store value.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: [minStoreValue\(\)](#) <= [maxStoreValue\(\)](#)

long RTRProxyManagedGaugeConfig::minFactoryDefault () const [inline, inherited]

The minimum factory default value.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: [minFactoryDefault\(\)](#) <= [maxFactoryDefault\(\)](#)

long RTRProxyManagedGaugeConfig::maxStoreValue () const [inline, inherited]

The maximum store value.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: [maxStoreValue\(\)](#) >= [minStoreValue\(\)](#)

long RTRProxyManagedGaugeConfig::maxFactoryDefault () const [inline, inherited]

The maximum factory default value.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: [maxFactoryDefault\(\)](#) >= [minFactoryDefault\(\)](#)

RTRProxyManagedLargeNumeric - Comparison

Functions

- RTR_BOOL [RTRProxyManagedLargeNumeric::operator==](#) (RTR_I64 rhs) const

Function Documentation

RTR_BOOL RTRProxyManagedLargeNumeric::operator== (RTR_I64 rhs) const [inline, inherited]

REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedLargeNumeric - Access

Functions

- RTR_I64 [RTRProxyManagedLargeNumeric::value](#) () const

Function Documentation

RTR_I64 RTRProxyManagedLargeNumeric::value () const [inline, inherited]

The current value of this variable.

REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedLargeNumeric - Transformation

Functions

- virtual [RTRString RTRProxyManagedLargeNumeric::toString](#) () const
- [RTRProxyManagedLargeNumeric::operator](#) RTR_I64 () const

Function Documentation

virtual [RTRString](#) RTRProxyManagedLargeNumeric::toString () const [virtual, inherited]

The value of this variable represented as a string.

REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedLargeNumeric::operator RTR_I64 () const [inline, inherited]

This variable as an RTR_I64 (long long for Unix or __int64 for Windows).

REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedNumeric - Comparison

Functions

- RTRBOOL [RTRProxyManagedNumeric::operator==](#) (long rhs) const

Function Documentation

RTRBOOL [RTRProxyManagedNumeric::operator==](#) (long *rhs*) const [inline, inherited]
 REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedNumeric - Access

Functions

- long [RTRProxyManagedNumeric::value](#) () const

Function Documentation

long [RTRProxyManagedNumeric::value](#) () const [inline, inherited]
 The current value of this variable.
 REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedNumeric - Transformation

Functions

- virtual [RTRString RTRProxyManagedNumeric::toString](#) () const
- [RTRProxyManagedNumeric::operator long](#) () const
- RTRProxyManagedNumeric::operator RTRProxyManagedGauge & ()
- RTRProxyManagedNumeric::operator const RTRProxyManagedGauge & () const
- RTRProxyManagedNumeric::operator RTRProxyManagedGaugeConfig & ()
- RTRProxyManagedNumeric::operator const RTRProxyManagedGaugeConfig & () const
- RTRProxyManagedNumeric::operator RTRProxyManagedNumericConfig & ()
- RTRProxyManagedNumeric::operator const RTRProxyManagedNumericConfig & () const
- RTRProxyManagedNumeric::operator RTRProxyManagedNumericRange & ()
- RTRProxyManagedNumeric::operator const RTRProxyManagedNumericRange & () const

Function Documentation

virtual [RTRString RTRProxyManagedNumeric::toString](#) () const [virtual, inherited]
 The value of this variable represented as a string.
 REQUIRE: [inSync\(\)](#) && !error()

Implements [RTRProxyManagedVariable.RTRProxyManagedNumeric::operator long](#) () const [inline, inherited]
 This variable as a long.
 REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedNumeric::operator [RTRProxyManagedGauge](#) & () [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::Gauge || [type\(\)](#) == RTRProxyManagedVariableHandle::GaugeConfig

Reimplemented from [RTRProxyManagedVariable](#).RTRProxyManagedNumeric::operator const [RTRProxyManagedGauge](#) & () const [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::Gauge || [type\(\)](#) == RTRProxyManagedVariableHandle::GaugeConfig

Reimplemented from [RTRProxyManagedVariable](#).RTRProxyManagedNumeric::operator [RTRProxyManagedGaugeConfig](#) & () [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::GaugeConfig

Reimplemented from [RTRProxyManagedVariable](#).Reimplemented in [RTRProxyManagedGauge](#).RTRProxyManagedNumeric::operator const [RTRProxyManagedGaugeConfig](#) & () const [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::GaugeConfig

Reimplemented from [RTRProxyManagedVariable](#).Reimplemented in [RTRProxyManagedGauge](#).RTRProxyManagedNumeric::operator [RTRProxyManagedNumericConfig](#) & () [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::NumericConfig

Reimplemented from [RTRProxyManagedVariable](#).RTRProxyManagedNumeric::operator const [RTRProxyManagedNumericConfig](#) & () const [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::NumericConfig

Reimplemented from [RTRProxyManagedVariable](#).RTRProxyManagedNumeric::operator [RTRProxyManagedNumericRange](#) & () [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::NumericRange

Reimplemented from [RTRProxyManagedVariable](#).RTRProxyManagedNumeric::operator const [RTRProxyManagedNumericRange](#) & () const [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::NumericRange

Reimplemented from [RTRProxyManagedVariable](#).

RTRProxyManagedNumericConfig - Access

Functions

- long [RTRProxyManagedNumericConfig::activeValue](#) () const

Function Documentation

long RTRProxyManagedNumericConfig::activeValue () const [inline, inherited]

A synonym for [value\(\)](#).

REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedNumericConfig - Attributes

Functions

- long [RTRProxyManagedNumericConfig::minValue](#) () const
- long [RTRProxyManagedNumericConfig::maxValue](#) () const
- long [RTRProxyManagedNumericConfig::storeValue](#) () const
- long [RTRProxyManagedNumericConfig::factoryDefault](#) () const

- RTRBOOL [RTRProxyManagedNumericConfig::modifyEnabled](#) () const
- RTRBOOL [RTRProxyManagedNumericConfig::hasStore](#) () const
- RTRBOOL [RTRProxyManagedNumericConfig::isStoreActive](#) () const
- RTRBOOL [RTRProxyManagedNumericConfig::isStoreClassConfig](#) () const
- RTRBOOL [RTRProxyManagedNumericConfig::isStoreInstanceConfig](#) () const

Function Documentation

long RTRProxyManagedNumericConfig::minValue () const [inline, inherited]

The minimum value which may be assigned to this variable.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: [minValue\(\)](#) <= [maxValue\(\)](#)

ENSURE: [minValue\(\)](#) <= [value\(\)](#)

long RTRProxyManagedNumericConfig::maxValue () const [inline, inherited]

The maximum value which may be assigned to this variable.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: [maxValue\(\)](#) >= [minValue\(\)](#)

ENSURE: [maxValue\(\)](#) >= [value\(\)](#)

long RTRProxyManagedNumericConfig::storeValue () const [inline, inherited]

The store value

REQUIRE: [inSync\(\)](#) && !error()

long RTRProxyManagedNumericConfig::factoryDefault () const [inline, inherited]

The factory default value for this variable.

REQUIRE: [inSync\(\)](#) && !error()

RTRBOOL RTRProxyManagedNumericConfig::modifyEnabled () const [inline, inherited]

Is the consumer permitted to modify this variable?

REQUIRE: [inSync\(\)](#) && !error()

RTRBOOL RTRProxyManagedNumericConfig::hasStore () const [inherited]

Is the variable a client of a RTRVariableConfig?

REQUIRE: [inSync\(\)](#) && !error()

RTRBOOL RTRProxyManagedNumericConfig::isStoreActive () const [inherited]

Is the RTRVariableConfig in an active state?

REQUIRE: [inSync\(\)](#) && !error()

REQUIRE: [hasStore\(\)](#)

RTRBOOL RTRProxyManagedNumericConfig::isStoreClassConfig () const [inherited]

Is the RTRVariableConfig's context a class config?

REQUIRE: [inSync\(\)](#) && !error()

REQUIRE: [hasStore\(\)](#)

RTRBOOL RTRProxyManagedNumericConfig::isStoreInstanceConfig () const [inherited]

Is the RTRVariableConfig's context a instance config?

REQUIRE: [inSync\(\)](#) && !error()

REQUIRE: [hasStore\(\)](#)

RTRProxyManagedNumericConfig - Assignment

Functions

- [RTRProxyManagedNumericConfig](#) & [RTRProxyManagedNumericConfig::operator=](#) (long rhs)

Function Documentation

[RTRProxyManagedNumericConfig](#) & [RTRProxyManagedNumericConfig::operator=](#) (long rhs) [inline, inherited]

REQUIRE: [inSync\(\)](#) && !error()

REQUIRE: rhs >= [minValue\(\)](#)

REQUIRE: rhs <= [maxValue\(\)](#)

RTRProxyManagedNumericConfig - Operations

Functions

- virtual void [RTRProxyManagedNumericConfig::set](#) (long newValue)=0

Function Documentation

virtual void [RTRProxyManagedNumericConfig::set](#) (long *newValue*) [pure virtual, inherited]

A synonym for [operator=\(\)](#)

REQUIRE: [inSync\(\)](#) && !error()

REQUIRE: newValue >= [minValue\(\)](#)

REQUIRE: newValue <= [maxValue\(\)](#)

RTRProxyManagedNumericRange - Attributes

Functions

- long [RTRProxyManagedNumericRange::minValue](#) () const
- long [RTRProxyManagedNumericRange::maxValue](#) () const

Function Documentation

long [RTRProxyManagedNumericRange::minValue](#) () const [inline, inherited]

The minimum value which may be assigned to this variable.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: [minValue\(\)](#) <= [maxValue\(\)](#)

ENSURE: [minValue\(\)](#) <= [value\(\)](#)

long [RTRProxyManagedNumericRange::maxValue](#) () const [inline, inherited]

The maximum value which may be assigned to this variable.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: [maxValue\(\)](#) >= [minValue\(\)](#)

ENSURE: [maxValue\(\)](#) >= [value\(\)](#)

RTRProxyManagedNumericRange - Assignment

Functions

- [RTRProxyManagedNumericRange](#) & [RTRProxyManagedNumericRange::operator=](#) (long rhs)

Function Documentation

[RTRProxyManagedNumericRange](#) & [RTRProxyManagedNumericRange::operator=](#) (long rhs) [inline, inherited]

REQUIRE: [inSync\(\)](#) && !error()

REQUIRE: rhs >= [minValue\(\)](#)

REQUIRE: rhs <= [maxValue\(\)](#)

RTRProxyManagedNumericRange - Operations

Functions

- virtual void [RTRProxyManagedNumericRange::set](#) (long newValue)=0

Function Documentation

virtual void [RTRProxyManagedNumericRange::set](#) (long newValue) [pure virtual, inherited]

A synonym for [operator=\(\)](#)

REQUIRE: [inSync\(\)](#) && !error()

REQUIRE: newValue >= [minValue\(\)](#)

REQUIRE: newValue <= [maxValue\(\)](#)

RTRProxyManagedObject - Identity

Functions

- const [RTRObjectId](#) & [RTRProxyManagedObject::instanceId](#) () const
- const [RTRString](#) & [RTRProxyManagedObject::name](#) () const
- const [RTRObjectId](#) & [RTRProxyManagedObject::classId](#) () const

Function Documentation

const [RTRObjectId](#) & [RTRProxyManagedObject::instanceId](#) () const [inline, inherited]

The instance identifier of this object.

const [RTRString](#) & [RTRProxyManagedObject::name](#) () const [inline, inherited]

The name of this object.

const [RTRObjectId](#) & [RTRProxyManagedObject::classId](#) () const [inline, inherited]

The class (type) identifier of this object.

RTRProxyManagedObject - Attributes

Functions

- const [RTRString](#) & [RTRProxyManagedObject::description](#) () const
- PMOState [RTRProxyManagedObject::state](#) () const
- PMOState [RTRProxyManagedObject::previousState](#) () const
- const [RTRString](#) & [RTRProxyManagedObject::text](#) () const

Function Documentation

const [RTRString](#) & [RTRProxyManagedObject::description](#) () const [inline, inherited]

The description of this variable.

REQUIRE: [inSync\(\)](#) && !error()

[RTRProxyManagedObject::PMOState](#) [RTRProxyManagedObject::state](#) () const [inline, inherited]

The state attribute of this proxy managed object.

REQUIRE: [inSync\(\)](#) && !error()

[RTRProxyManagedObject::PMOState](#) [RTRProxyManagedObject::previousState](#) () const [inline, inherited]

The previous state attribute of this proxy managed object.

REQUIRE: [inSync\(\)](#) && !error()

const [RTRString](#) & [RTRProxyManagedObject::text](#) () const [inline, inherited]

The textual explanation of any error state.

RTRProxyManagedObject - State

Functions

- RTRBOOL [RTRProxyManagedObject::error](#) () const
- RTRBOOL [RTRProxyManagedObject::inSync](#) () const

Function Documentation

RTRBOOL [RTRProxyManagedObject::error](#) () const [inline, inherited]

Is this proxy in an error state? If so, [text\(\)](#) provides an explanation.

RTRBOOL [RTRProxyManagedObject::inSync](#) () const [inline, inherited]

Is this proxy in sync with the server which provided it?

RTRProxyManagedObject - Query

Functions

- RTRBOOL [RTRProxyManagedObject::hasChild](#) (const [RTRString](#) &) const
- RTRBOOL [RTRProxyManagedObject::hasVariable](#) (const [RTRString](#) &) const

Function Documentation

RTRBOOL [RTRProxyManagedObject::hasChild](#) (const [RTRString](#) &) const [inherited]

Does the object represented by this proxy have a child with the given name?

REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedObject::hasVariable (const [RTRString](#) &) const [inherited]

Does the object represented by this proxy contain a variable with the given name?

REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedObject - Access Sequentially

Functions

- [RTRProxyManagedVarHandleIterator](#) [RTRProxyManagedObject::variableHandles](#) () const
- [RTRProxyManagedObjectHandleIterator](#) [RTRProxyManagedObject::childHandles](#) () const

Function Documentation

[RTRProxyManagedVarHandleIterator](#) [RTRProxyManagedObject::variableHandles](#) () const [inherited]

An iterator which provides sequential access to all variable handles contained by this ProxyManagedObject.

REQUIRE: [inSync\(\)](#) && !error()

[RTRProxyManagedObjectHandleIterator](#) [RTRProxyManagedObject::childHandles](#) () const [inherited]

An iterator which provides sequential access to all child ProxyManagedObjects contained by this ProxyManagedObject.

REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedObject - Access Randomly

Functions

- virtual RTRObjRef< [RTRProxyManagedObject](#) > [RTRProxyManagedObject::childByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedVariablePtr [RTRProxyManagedObject::variableByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedBooleanPtr [RTRProxyManagedObject::booleanByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedBooleanConfigPtr [RTRProxyManagedObject::booleanConfigByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedCounterPtr [RTRProxyManagedObject::counterByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedGaugePtr [RTRProxyManagedObject::gaugeByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedGaugeConfigPtr [RTRProxyManagedObject::gaugeConfigByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedNumericPtr [RTRProxyManagedObject::numericByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedNumericConfigPtr [RTRProxyManagedObject::numericConfigByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedNumericRangePtr [RTRProxyManagedObject::numericRangeByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedStringPtr [RTRProxyManagedObject::stringByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedStringConfigPtr [RTRProxyManagedObject::stringConfigByName](#) (const [RTRString](#) &name) const

Function Documentation

virtual RTRObjRef<[RTRProxyManagedObject](#)> [RTRProxyManagedObject::childByName](#) (const [RTRString](#) & name) const [virtual, inherited]

The child, if any, which has the given name. The return type is RTRProxyManagedObjectPtr

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: result == null implies !hasChild(name)

RTRProxyManagedVariablePtr RTRProxyManagedObject::variableByName (const [RTRString](#) & name) const [inherited]
The variable, if any, which has the given name.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: result == null implies !hasVariable(name)

RTRProxyManagedBooleanPtr RTRProxyManagedObject::booleanByName (const [RTRString](#) & name) const [inherited]
The boolean variable, if any, which has the given name.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: result == null implies !hasVariable(name) || (variable(name)->type != Boolean && variable(name)->type != BooleanConfig)

RTRProxyManagedBooleanConfigPtr RTRProxyManagedObject::booleanConfigByName (const [RTRString](#) & name) const [inherited]
The boolean config variable, if any, which has the given name.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: result == null implies !hasVariable(name) || variable(name)->type != BooleanConfig

RTRProxyManagedCounterPtr RTRProxyManagedObject::counterByName (const [RTRString](#) & name) const [inherited]
The counter variable, if any, which has the given name.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: result == null implies !hasVariable(name) || variable(name)->type != Counter

RTRProxyManagedGaugePtr RTRProxyManagedObject::gaugeByName (const [RTRString](#) & name) const [inherited]
The gauge variable, if any, which has the given name.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: result == null implies !hasVariable(name) || (variable(name)->type != Gauge && variable(name)->type != GaugeConfig)

RTRProxyManagedGaugeConfigPtr RTRProxyManagedObject::gaugeConfigByName (const [RTRString](#) & name) const [inherited]
The gauge config variable, if any, which has the given name.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: result == null implies !hasVariable(name) || variable(name)->type != GaugeConfig

RTRProxyManagedNumericPtr RTRProxyManagedObject::numericByName (const [RTRString](#) & name) const [inherited]
The numeric variable, if any, which has the given name.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE:

result == null implies !hasVariable(name) || (
variable(name)->type != Numeric &&
variable(name)->type != NumericConfig &&
variable(name)->type != NumericRange &&
variable(name)->type != Gauge &&

variable(name)->type != GaugeConfig)

RTRProxyManagedNumericConfigPtr RTRProxyManagedObject::numericConfigByName (const [RTRString](#) & name) const [inherited]

The numeric config variable, if any, which has the given name.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: result == null implies !hasVariable(name) || variable(name)->type != NumericConfig

RTRProxyManagedNumericRangePtr RTRProxyManagedObject::numericRangeByName (const [RTRString](#) & name) const [inherited]

The numeric parameter variable, if any, which has the given name.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: result == null implies !hasVariable(name) || variable(name)->type != NumericRange

RTRProxyManagedStringPtr RTRProxyManagedObject::stringByName (const [RTRString](#) & name) const [inherited]

The string variable, if any, which has the given name.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: result == null implies !hasVariable(name) || (variable(name)->type != String && variable(name)->type != StringConfig)

RTRProxyManagedStringConfigPtr RTRProxyManagedObject::stringConfigByName (const [RTRString](#) & name) const [inherited]

The string config variable, if any, which has the given name.

REQUIRE: [inSync\(\)](#) && !error()

ENSURE: result == null implies !hasVariable(name) || variable(name)->type != StringConfig

RTRProxyManagedObject - Event client management

Functions

- RTRBOOL [RTRProxyManagedObject::hasClient](#) ([RTRProxyManagedObjectClient](#) &) const
- void [RTRProxyManagedObject::addClient](#) ([RTRProxyManagedObjectClient](#) &)
- void [RTRProxyManagedObject::dropClient](#) ([RTRProxyManagedObjectClient](#) &)

Function Documentation

RTRBOOL RTRProxyManagedObject::hasClient ([RTRProxyManagedObjectClient](#) &) const [inherited]

Is the given client registered to receive update and state events from this ProxyManagedObject.

void RTRProxyManagedObject::addClient ([RTRProxyManagedObjectClient](#) &) [inherited]

Register the given client to receive update and state events from this ProxyManagedObject.

REQUIRE: !hasClient(client)

ENSURE: hasClient(client)

void RTRProxyManagedObject::dropClient ([RTRProxyManagedObjectClient](#) &) [inherited]

Un-register the given client to receive update events from this ProxyManagedObject.

REQUIRE: hasClient(client)

ENSURE: !hasClient(client)

RTRProxyManagedObject - Operations from RTRLockableObj

Functions

- virtual void [RTRProxyManagedObject::lock](#) ()
- virtual void **unlock** ()
- virtual RTRBOOL [RTRProxyManagedObject::locked](#) () const

Function Documentation

virtual void RTRProxyManagedObject::lock () [virtual, inherited]

Locking is implemented via the server.

Reimplemented from [RTRLockableObj](#). **virtual RTRBOOL RTRProxyManagedObject::locked () const** [virtual, inherited]

Is this locked by calling thread? It is used in PRECONDITION of application class to ensure that an instance of [RTRLockableObj](#) must be locked before access.

NOTE: This only serves as necessary-but-not-sufficient condition;
i.e., ([locked\(\)](#)== RTRTRUE) == (possibly right);
([locked\(\)](#)== RTRFALSE) == (definitely wrong);

By default this call will always return RTRTRUE unless static member alwaysLocked is set RTRFALSE.

Reimplemented from [RTRLockableObj](#).

RTRProxyManagedObjectClient - Event processing

Functions

- virtual void [RTRProxyManagedObjectClient::processProxyManagedObjectError](#) (const [RTRProxyManagedObject](#) &)=0
- virtual void [RTRProxyManagedObjectClient::processProxyManagedObjectSync](#) (const [RTRProxyManagedObject](#) &)=0
- virtual void [RTRProxyManagedObjectClient::processProxyManagedObjectDeleted](#) (const [RTRProxyManagedObject](#) &)=0
- virtual void [RTRProxyManagedObjectClient::processProxyManagedObjectInfo](#) (const [RTRProxyManagedObject](#) &)=0
- virtual void [RTRProxyManagedObjectClient::processProxyManagedObjectInService](#) (const [RTRProxyManagedObject](#) &)=0
- virtual void [RTRProxyManagedObjectClient::processProxyManagedObjectRecovering](#) (const [RTRProxyManagedObject](#) &)=0
- virtual void [RTRProxyManagedObjectClient::processProxyManagedObjectWaiting](#) (const [RTRProxyManagedObject](#) &)=0
- virtual void [RTRProxyManagedObjectClient::processProxyManagedObjectDead](#) (const [RTRProxyManagedObject](#) &)=0
- virtual void [RTRProxyManagedObjectClient::processProxyManagedObjectChildAdded](#) (const [RTRProxyManagedObject](#) &, const [RTRProxyManagedObjectHandle](#) &)=0
- virtual void [RTRProxyManagedObjectClient::processProxyManagedObjectChildRemoved](#) (const [RTRProxyManagedObject](#) &, const [RTRProxyManagedObjectHandle](#) &)=0
- virtual void [RTRProxyManagedObjectClient::processProxyManagedObjectVariableAdded](#) (const [RTRProxyManagedObject](#) &, const [RTRProxyManagedVariableHandle](#) &)=0
- virtual void [RTRProxyManagedObjectClient::processProxyManagedObjectVariableRemoved](#) (const [RTRProxyManagedObject](#) &, const [RTRProxyManagedVariableHandle](#) &)=0

Function Documentation

virtual void RTRProxyManagedObjectClient::processProxyManagedObjectError (const [RTRProxyManagedObject](#) &) [pure virtual, inherited]

The given proxy managed object has transitioned into an unrecoverable error state.

virtual void RTRProxyManagedObjectClient::processProxyManagedObjectSync (const [RTRProxyManagedObject](#) &) [pure virtual, inherited]

The given ProxyManagedObject is in Sync.

virtual void RTRProxyManagedObjectClient::processProxyManagedObjectDeleted (const [RTRProxyManagedObject](#) &) [pure virtual, inherited]

The given ProxyManagedObject has been removed.

virtual void RTRProxyManagedObjectClient::processProxyManagedObjectInfo (const [RTRProxyManagedObject](#) &) [pure virtual, inherited]

The given proxy managed object has additional information concerning its state attribute. The proxy managed object has not changed state. Refer to the text() attribute of the given proxy managed object.

virtual void RTRProxyManagedObjectClient::processProxyManagedObjectInService (const [RTRProxyManagedObject](#) &) [pure virtual, inherited]

The given proxy managed object is now in a normal service state.

virtual void RTRProxyManagedObjectClient::processProxyManagedObjectRecovering (const [RTRProxyManagedObject](#) &) [pure virtual, inherited]

The given proxy managed object is in a service interrupted state but is attempting to recover normal service automatically.

virtual void RTRProxyManagedObjectClient::processProxyManagedObjectWaiting (const [RTRProxyManagedObject](#) &) [pure virtual, inherited]

The given proxy managed object is in a service interrupted state and is waiting for manual intervention to restore normal service.

virtual void RTRProxyManagedObjectClient::processProxyManagedObjectDead (const [RTRProxyManagedObject](#) &) [pure virtual, inherited]

The given proxy managed object has entered an unrecoverable error state.

virtual void RTRProxyManagedObjectClient::processProxyManagedObjectChildAdded (const [RTRProxyManagedObject](#) &, const [RTRProxyManagedObjectHandle](#) &) [pure virtual, inherited]

The given proxy managed object has a new child.

virtual void RTRProxyManagedObjectClient::processProxyManagedObjectChildRemoved (const [RTRProxyManagedObject](#) &, const [RTRProxyManagedObjectHandle](#) &) [pure virtual, inherited]

The given proxy managed object has had a child removed.

virtual void RTRProxyManagedObjectClient::processProxyManagedObjectVariableAdded (const [RTRProxyManagedObject](#) &, const [RTRProxyManagedVariableHandle](#) &) [pure virtual, inherited]

The given proxy managed object has a new variable.

virtual void RTRProxyManagedObjectClient::processProxyManagedObjectVariableRemoved (const [RTRProxyManagedObject](#) &, const [RTRProxyManagedVariableHandle](#) &) [pure virtual, inherited]

The given proxy managed object has had a variable removed.

RTRProxyManagedObjectServer - Attributes

Functions

- const [RTRString](#) & [RTRProxyManagedObjectServer::text](#) () const

Function Documentation

const [RTRString](#) & RTRProxyManagedObjectServer::text () const [inline, inherited]

An explanation for the state of this proxy.

RTRProxyManagedObjectServer - State

Functions

- RTRBOOL [RTRProxyManagedObjectServer::error](#) () const
- RTRBOOL [RTRProxyManagedObjectServer::inSync](#) () const

Function Documentation

RTRBOOL RTRProxyManagedObjectServer::error () const [inline, inherited]

Is this proxy in an unrecoverable error state?

RTRBOOL RTRProxyManagedObjectServer::inSync () const [inline, inherited]

Is this proxy synchronized with its remote server?

RTRProxyManagedObjectServer - Access Sequentially

Functions

- [RTRProxyManagedObjectHandleIterator](#) [RTRProxyManagedObjectServer::roots](#) () const

Function Documentation

[RTRProxyManagedObjectHandleIterator](#) RTRProxyManagedObjectServer::roots () const [inherited]

An iterator which provides sequential access to handles of all root objects available from this proxy.

REQUIRE: [inSync\(\)](#)

RTRProxyManagedObjectServer - Access Randomly

Functions

- virtual RTRProxyManagedObjectPtr [RTRProxyManagedObjectServer::object](#) (const [RTRProxyManagedObjectHandle](#) &id) const

Function Documentation

virtual RTRProxyManagedObjectPtr RTRProxyManagedObjectServer::object (const [RTRProxyManagedObjectHandle](#) &id) const [virtual, inherited]

The object with the given id (handle).

RTRProxyManagedObjectServer - Event client management

Functions

- RTRBOOL [RTRProxyManagedObjectServer::hasClient](#) (const [RTRProxyManagedObjectServerClient](#) &) const
- void [RTRProxyManagedObjectServer::addClient](#) (const [RTRProxyManagedObjectServerClient](#) &client)
- void [RTRProxyManagedObjectServer::dropClient](#) (const [RTRProxyManagedObjectServerClient](#) &client)

Function Documentation

RTRBOOL RTRProxyManagedObjectServer::hasClient (const [RTRProxyManagedObjectServerClient](#) &) const [inherited]
Is the given client registered to receive update and state events from this ProxyManaged Object Server?

void RTRProxyManagedObjectServer::addClient (const [RTRProxyManagedObjectServerClient](#) & *client*) [inherited]
Register the given client to receive update and state events from this ProxyManaged Object Server.

REQUIRE: !hasClient(client)

REQUIRE: hasClient(client)

void RTRProxyManagedObjectServer::dropClient (const [RTRProxyManagedObjectServerClient](#) & *client*) [inherited]
Un-register the given client to receive update events from this ProxyManaged Object Server.

ENSURE: !hasClient(client)

RTRProxyManagedString - Access

Functions

- const [RTRString](#) & [RTRProxyManagedString::value](#) () const

Function Documentation

const [RTRString](#) & RTRProxyManagedString::value () const [inline, inherited]
The current value of this variable.

REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedString - Transformation

Functions

- virtual [RTRString](#) [RTRProxyManagedString::toString](#) () const
- RTRProxyManagedString::operator const char * () const
- RTRProxyManagedString::operator RTRProxyManagedStringConfig & ()
- RTRProxyManagedString::operator const RTRProxyManagedStringConfig & () const

Function Documentation

virtual [RTRString](#) RTRProxyManagedString::toString () const [virtual, inherited]
The value of this variable represented as a string.

REQUIRE: [inSync\(\)](#) && !error()

Implements [RTRProxyManagedVariable](#).RTRProxyManagedString::operator const char * () const [inline, inherited]
This variable as a C string (null terminated).

REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedString::operator [RTRProxyManagedStringConfig](#) & () [inherited]
REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::StringConfig

Reimplemented from [RTRProxyManagedVariable](#).RTRProxyManagedString::operator const [RTRProxyManagedStringConfig](#) & () const [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::StringConfig

Reimplemented from [RTRProxyManagedVariable](#).

RTRProxyManagedString - Comparison

Functions

- RTRBOOL [RTRProxyManagedString::operator==](#) (const char *) const

Function Documentation

RTRBOOL RTRProxyManagedString::operator== (const char *) const [inline, inherited]
 REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedString - Attributes

Functions

- RTRBOOL [RTRProxyManagedString::modifyEnabled](#) () const

Function Documentation

RTRBOOL RTRProxyManagedString::modifyEnabled () const [inline, inherited]
 Is the managing application permitted to modify this variable?
 REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedString - Assignment

Functions

- [RTRProxyManagedString](#) & [RTRProxyManagedString::operator=](#) (const [RTRString](#) &rhs)
- [RTRProxyManagedString](#) & [RTRProxyManagedString::operator=](#) (const char *rhs)

Function Documentation

[RTRProxyManagedString](#) & RTRProxyManagedString::operator= (const [RTRString](#) & rhs) [inline, inherited]
 Assigns the value of this variable to rhs.
 REQUIRE: [inSync\(\)](#) && !error()
 REQUIRE: [modifyEnabled\(\)](#)

Reimplemented in [RTRProxyManagedStringConfig](#). [RTRProxyManagedString](#) & RTRProxyManagedString::operator= (const char * rhs) [inline, inherited]
 Assigns the value of this variable to rhs.
 REQUIRE: [inSync\(\)](#) && !error()
 REQUIRE: [modifyEnabled\(\)](#)

Reimplemented in [RTRProxyManagedStringConfig](#).

RTRProxyManagedString - Operations

Functions

- virtual void [RTRProxyManagedString::set](#) (const [RTRString](#) &newValue)=0

- virtual void [RTRProxyManagedString::set](#) (const char *newValue)=0

Function Documentation

virtual void [RTRProxyManagedString::set](#) (const [RTRString](#) & *newValue*) [pure virtual, inherited]

A synonym for [operator=\(\)](#)

REQUIRE: [inSync\(\)](#) && !error()

REQUIRE: [modifyEnabled\(\)](#)

virtual void [RTRProxyManagedString::set](#) (const char * *newValue*) [pure virtual, inherited]

A synonym for [operator=\(\)](#)

REQUIRE: [inSync\(\)](#) && !error()

REQUIRE: [modifyEnabled\(\)](#)

RTRProxyManagedStringConfig - Access

Functions

- const [RTRString](#) & [RTRProxyManagedStringConfig::activeValue](#) () const

Function Documentation

const [RTRString](#) & [RTRProxyManagedStringConfig::activeValue](#) () const [inline, inherited]

A synonym for [value\(\)](#).

REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedStringConfig - Attributes

Functions

- const [RTRString](#) & [RTRProxyManagedStringConfig::storeValue](#) () const
- const [RTRString](#) & [RTRProxyManagedStringConfig::factoryDefault](#) () const
- RTRBOOL [RTRProxyManagedStringConfig::hasStore](#) () const
- RTRBOOL [RTRProxyManagedStringConfig::isStoreActive](#) () const
- RTRBOOL [RTRProxyManagedStringConfig::isStoreClassConfig](#) () const
- RTRBOOL [RTRProxyManagedStringConfig::isStoreInstanceConfig](#) () const

Function Documentation

const [RTRString](#) & [RTRProxyManagedStringConfig::storeValue](#) () const [inline, inherited]

The store value.

REQUIRE: [inSync\(\)](#) && !error()

const [RTRString](#) & [RTRProxyManagedStringConfig::factoryDefault](#) () const [inline, inherited]

The factory default value.

REQUIRE: [inSync\(\)](#) && !error()

RTRBOOL [RTRProxyManagedStringConfig::hasStore](#) () const [inherited]

Is the variable a client of a RTRVariableConfig?

REQUIRE: [inSync\(\)](#) && !error()

RTRBOOL RTRProxyManagedStringConfig::isStoreActive () const [inherited]

Is the RTRVariableConfig in an active state?

REQUIRE: [inSync\(\)](#) && !error()

REQUIRE: [hasStore\(\)](#)

RTRBOOL RTRProxyManagedStringConfig::isStoreClassConfig () const [inherited]

Is the RTRVariableConfig's context a class config?

REQUIRE: [inSync\(\)](#) && !error()

REQUIRE: [hasStore\(\)](#)

RTRBOOL RTRProxyManagedStringConfig::isStoreInstanceConfig () const [inherited]

Is the RTRVariableConfig's context a instance config?

REQUIRE: [inSync\(\)](#) && !error()

REQUIRE: [hasStore\(\)](#)

RTRProxyManagedStringConfig - Assignment

Functions

- [RTRProxyManagedStringConfig](#) & [RTRProxyManagedStringConfig::operator=](#) (const [RTRString](#) &rhs)
- [RTRProxyManagedStringConfig](#) & [RTRProxyManagedStringConfig::operator=](#) (const char *rhs)

Function Documentation

[RTRProxyManagedStringConfig](#) & [RTRProxyManagedStringConfig::operator=](#) (const [RTRString](#) & rhs) [inline, inherited]

Assigns the value of this variable to rhs.

REQUIRE: [inSync\(\)](#) && !error()

REQUIRE: [modifyEnabled\(\)](#)

Reimplemented from [RTRProxyManagedString.RTRProxyManagedStringConfig](#) & [RTRProxyManagedStringConfig::operator=](#) (const char * rhs) [inline, inherited]

Assigns the value of this variable to rhs.

REQUIRE: [inSync\(\)](#) && !error()

REQUIRE: [modifyEnabled\(\)](#)

Reimplemented from [RTRProxyManagedString](#).

RTRProxyManagedVariable - Identify

Functions

- const [RTRString](#) & [RTRProxyManagedVariable::name](#) () const

Function Documentation

const [RTRString](#) & [RTRProxyManagedVariable::name](#) () const [inline, inherited]

The name of this variable.

RTRProxyManagedVariable - Attributes

Functions

- [RTRProxyManagedObject](#) & [RTRProxyManagedVariable::context](#) () const
- const [RTRString](#) & [RTRProxyManagedVariable::text](#) () const
- [RTRProxyManagedVariableHandle::MVType](#) [RTRProxyManagedVariable::type](#) () const
- const [RTRString](#) & [RTRProxyManagedVariable::description](#) () const

Function Documentation

[RTRProxyManagedObject](#) & [RTRProxyManagedVariable::context](#) () const [inline, inherited]

The object which contains this variable.

REQUIRE: !error()

const [RTRString](#) & [RTRProxyManagedVariable::text](#) () const [inline, inherited]

The textual explanation of any error state.

[RTRProxyManagedVariableHandle::MVType](#) [RTRProxyManagedVariable::type](#) () const [inline, inherited]

The type of this variable. Either Boolean, Numeric, Gauge, String, Counter, NumericRange, NumericConfig, StringConfig, BooleanConfig or GaugeConfig

const [RTRString](#) & [RTRProxyManagedVariable::description](#) () const [inline, inherited]

A textual description of the variable.

REQUIRE: [inSync\(\)](#) && !error()

RTRProxyManagedVariable - State

Functions

- RTRBOOL [RTRProxyManagedVariable::error](#) () const
- RTRBOOL [RTRProxyManagedVariable::inSync](#) () const

Function Documentation

RTRBOOL [RTRProxyManagedVariable::error](#) () const [inline, inherited]

Is this proxy in an error state? If so, [text\(\)](#) provides an explanation.

RTRBOOL [RTRProxyManagedVariable::inSync](#) () const [inline, inherited]

Is this proxy in sync with the server that provided it?

RTRProxyManagedVariable - Transformation

Functions

- virtual [RTRString](#) [RTRProxyManagedVariable::toString](#) () const =0
- [RTRString](#) [RTRProxyManagedVariable::typeString](#) () const
- [RTRProxyManagedVariable::operator](#) [RTRProxyManagedBoolean](#) & ()
- [RTRProxyManagedVariable::operator](#) const [RTRProxyManagedBoolean](#) & () const
- [RTRProxyManagedVariable::operator](#) [RTRProxyManagedBooleanConfig](#) & ()

- `RTRProxyManagedVariable::operator const RTRProxyManagedBooleanConfig & () const`
- `RTRProxyManagedVariable::operator RTRProxyManagedCounter & ()`
- `RTRProxyManagedVariable::operator const RTRProxyManagedCounter & () const`
- `RTRProxyManagedVariable::operator RTRProxyManagedGauge & ()`
- `RTRProxyManagedVariable::operator const RTRProxyManagedGauge & () const`
- `RTRProxyManagedVariable::operator RTRProxyManagedGaugeConfig & ()`
- `RTRProxyManagedVariable::operator const RTRProxyManagedGaugeConfig & () const`
- `RTRProxyManagedVariable::operator RTRProxyManagedNumeric & ()`
- `RTRProxyManagedVariable::operator const RTRProxyManagedNumeric & () const`
- `RTRProxyManagedVariable::operator RTRProxyManagedLargeNumeric & ()`
- `RTRProxyManagedVariable::operator const RTRProxyManagedLargeNumeric & () const`
- `RTRProxyManagedVariable::operator RTRProxyManagedNumericConfig & ()`
- `RTRProxyManagedVariable::operator const RTRProxyManagedNumericConfig & () const`
- `RTRProxyManagedVariable::operator RTRProxyManagedNumericRange & ()`
- `RTRProxyManagedVariable::operator const RTRProxyManagedNumericRange & () const`
- `RTRProxyManagedVariable::operator RTRProxyManagedString & ()`
- `RTRProxyManagedVariable::operator const RTRProxyManagedString & () const`
- `RTRProxyManagedVariable::operator RTRProxyManagedStringConfig & ()`
- `RTRProxyManagedVariable::operator const RTRProxyManagedStringConfig & () const`

Function Documentation

virtual [RTRString](#) `RTRProxyManagedVariable::toString () const` [pure virtual, inherited]

The value of this variable represented as a string.

REQUIRE: [inSync\(\)](#) && !error()

Implemented in [RTRProxyManagedBoolean](#), [RTRProxyManagedCounter](#), [RTRProxyManagedNumeric](#), and [RTRProxyManagedString](#). `RTRProxyManagedVariable::typeString () const` [inline, inherited]

The variable type represented as a string.

`RTRProxyManagedVariable::operator RTRProxyManagedBoolean & ()` [inherited]

REQUIRE: [type\(\)](#) == `RTRProxyManagedVariableHandle::Boolean` || [type\(\)](#) == `RTRProxyManagedVariableHandle::BooleanConfig`

`RTRProxyManagedVariable::operator const RTRProxyManagedBoolean & () const` [inherited]

REQUIRE: [type\(\)](#) == `RTRProxyManagedVariableHandle::Boolean` || [type\(\)](#) == `RTRProxyManagedVariableHandle::BooleanConfig`

`RTRProxyManagedVariable::operator RTRProxyManagedBooleanConfig & ()` [inherited]

REQUIRE: [type\(\)](#) == `RTRProxyManagedVariableHandle::BooleanConfig`

Reimplemented in [RTRProxyManagedBoolean](#). `RTRProxyManagedVariable::operator const RTRProxyManagedBooleanConfig & () const` [inherited]

REQUIRE: [type\(\)](#) == `RTRProxyManagedVariableHandle::BooleanConfig`

Reimplemented in [RTRProxyManagedBoolean](#).RTRProxyManagedVariable::operator [RTRProxyManagedCounter](#) & () [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::Counter

RTRProxyManagedVariable::operator const [RTRProxyManagedCounter](#) & () const [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::Counter

RTRProxyManagedVariable::operator const [RTRProxyManagedLargeNumeric](#) & () const [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::LargeNumeric

RTRProxyManagedVariable::operator [RTRProxyManagedGauge](#) & () [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::Gauge || [type\(\)](#) == RTRProxyManagedVariableHandle::GaugeConfig

Reimplemented in [RTRProxyManagedNumeric](#).RTRProxyManagedVariable::operator const [RTRProxyManagedGauge](#) & () const [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::Gauge || [type\(\)](#) == RTRProxyManagedVariableHandle::GaugeConfig

Reimplemented in [RTRProxyManagedNumeric](#).RTRProxyManagedVariable::operator [RTRProxyManagedGaugeConfig](#) & () [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::GaugeConfig

Reimplemented in [RTRProxyManagedGauge](#), and [RTRProxyManagedNumeric](#).RTRProxyManagedVariable::operator const [RTRProxyManagedGaugeConfig](#) & () const [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::GaugeConfig

Reimplemented in [RTRProxyManagedGauge](#), and [RTRProxyManagedNumeric](#).RTRProxyManagedVariable::operator [RTRProxyManagedNumeric](#) & () [inherited]

REQUIRE:

[type\(\)](#) == RTRProxyManagedVariableHandle::Numeric ||

[type\(\)](#) == RTRProxyManagedVariableHandle::NumericConfig ||

[type\(\)](#) == RTRProxyManagedVariableHandle::NumericRange ||

[type\(\)](#) == RTRProxyManagedVariableHandle::Gauge ||

[type\(\)](#) == RTRProxyManagedVariableHandle::GaugeConfig

RTRProxyManagedVariable::operator const [RTRProxyManagedNumeric](#) & () const [inherited]

REQUIRE:

[type\(\)](#) == RTRProxyManagedVariableHandle::Numeric ||

[type\(\)](#) == RTRProxyManagedVariableHandle::NumericConfig ||

[type\(\)](#) == RTRProxyManagedVariableHandle::NumericRange ||

[type\(\)](#) == RTRProxyManagedVariableHandle::Gauge ||

[type\(\)](#) == RTRProxyManagedVariableHandle::GaugeConfig

RTRProxyManagedVariable::operator [RTRProxyManagedNumericConfig](#) & () [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::NumericConfig

Reimplemented in [RTRProxyManagedNumeric](#).RTRProxyManagedVariable::operator const [RTRProxyManagedNumericConfig](#) & () const [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::NumericConfig

Reimplemented in [RTRProxyManagedNumeric](#).RTRProxyManagedVariable::operator [RTRProxyManagedNumericRange](#) & () [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::NumericRange

Reimplemented in [RTRProxyManagedNumeric](#).RTRProxyManagedVariable::operator const [RTRProxyManagedNumericRange](#) & () const [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::NumericRange

Reimplemented in [RTRProxyManagedNumeric](#).RTRProxyManagedVariable::operator [RTRProxyManagedString](#) & () [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::String || [type\(\)](#) == RTRProxyManagedVariableHandle::StringConfig

RTRProxyManagedVariable::operator const [RTRProxyManagedString](#) & () const [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::String || [type\(\)](#) == RTRProxyManagedVariableHandle::StringConfig

RTRProxyManagedVariable::operator [RTRProxyManagedStringConfig](#) & () [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::StringConfig

Reimplemented in [RTRProxyManagedString](#).RTRProxyManagedVariable::operator const [RTRProxyManagedStringConfig](#) & () const [inherited]

REQUIRE: [type\(\)](#) == RTRProxyManagedVariableHandle::StringConfig

Reimplemented in [RTRProxyManagedString](#).

RTRProxyManagedVariable - Event client management

Functions

- RTRBOOL [RTRProxyManagedVariable::hasClient](#) ([RTRProxyManagedVariableClient](#) &) const
- void [RTRProxyManagedVariable::addClient](#) ([RTRProxyManagedVariableClient](#) &client)
- void [RTRProxyManagedVariable::dropClient](#) ([RTRProxyManagedVariableClient](#) &client)

Function Documentation

RTRBOOL [RTRProxyManagedVariable::hasClient](#) ([RTRProxyManagedVariableClient](#) &) const [inherited]

Is the given client registered to receive update and state events from this ProxyManagedwork Variable?

void [RTRProxyManagedVariable::addClient](#) ([RTRProxyManagedVariableClient](#) & *client*) [inherited]

Register the given client to receive update and state events from this ProxyManagedwork Variable.

REQUIRE: !hasClient(client)

ENSURE: hasClient(client)

void [RTRProxyManagedVariable::dropClient](#) ([RTRProxyManagedVariableClient](#) & *client*) [inherited]

Un-register the given client to receive update events from this ProxyManagedwork Variable.

REQUIRE: hasClient(client)

ENSURE: !hasClient(client)

RTRProxyManagedVariable - Operations from RTRLockableObj

Functions

- virtual void [RTRProxyManagedVariable::lock](#) ()
- virtual void [unlock](#) ()

- virtual RTRBOOL [RTRProxyManagedVariable::locked](#) () const

Function Documentation

virtual void RTRProxyManagedVariable::lock () [virtual, inherited]

Locking is implemented via its context object.

Reimplemented from [RTRLockableObj](#). **virtual RTRBOOL RTRProxyManagedVariable::locked () const** [virtual, inherited]

Is this locked by calling thread? It is used in PRECONDITION of application class to ensure that an instance of [RTRLockableObj](#) must be locked before access.

NOTE: This only serves as necessary-but-not-sufficient condition;
i.e., ([locked\(\)](#)== RTRTRUE) == (possibly right);
([locked\(\)](#)== RTRFALSE) == (definitely wrong);

By default this call will always return RTRTRUE unless static member alwaysLocked is set RTRFALSE.

Reimplemented from [RTRLockableObj](#).

RTRProxyManagedVariableClient - Event processing

Functions

- virtual void RTRProxyManagedVariableClient::processProxyManagedVariableError (RTRProxyManagedVariable &)=0
- virtual void [RTRProxyManagedVariableClient::processProxyManagedVariableSync](#) ([RTRProxyManagedVariable](#) &)=0
- virtual void RTRProxyManagedVariableClient::processProxyManagedVariableUpdate (RTRProxyManagedVariable &)=0
- virtual void RTRProxyManagedVariableClient::processProxyManagedVariableDeleted (RTRProxyManagedVariable &)=0

Function Documentation

virtual void RTRProxyManagedVariableClient::processProxyManagedVariableError ([RTRProxyManagedVariable](#) &) [pure virtual, inherited]

The given variable has transitioned to an unrecoverable error state.

virtual void RTRProxyManagedVariableClient::processProxyManagedVariableSync ([RTRProxyManagedVariable](#) &) [pure virtual, inherited]

The given variable has transitioned to the Sync state.

virtual void RTRProxyManagedVariableClient::processProxyManagedVariableUpdate ([RTRProxyManagedVariable](#) &) [pure virtual, inherited]

The given variable has changed.

virtual void RTRProxyManagedVariableClient::processProxyManagedVariableDeleted ([RTRProxyManagedVariable](#) &) [pure virtual, inherited]

The given variable has been deleted.

RTRExternalValue - Assignment

Functions

- [RTRExternalValue](#) & operator= (const [RTRExternalValue](#) &n)

RTRExternalValue - Conversion Functions

Functions

- `int RTRExternalValue::integer_from_hexadecimal ()`
- `int RTRExternalValue::integer_from_octal ()`
- `RTRBOOL RTRExternalValue::isTrue ()`
- `RTRBOOL RTRExternalValue::isFalse ()`
- `RTRLListOfExternalValue RTRExternalValue::list (char delimiter)`
- `RTRExternalValue * RTRExternalValue::duplicate ()`

Function Documentation

`int RTRExternalValue::integer_from_hexadecimal ()` [inherited]

Int value of hex ASCII number in this

`int RTRExternalValue::integer_from_octal ()` [inherited]

Int value of octal ASCII number in this

`RTRBOOL RTRExternalValue::isTrue ()` [inherited]

Boolean interpretation, false if value "FALSE".

`RTRBOOL RTRExternalValue::isFalse ()` [inherited]

Boolean interpretation, true if value "TRUE"

`RTRLListOfExternalValue RTRExternalValue::list (char delimiter)` [inherited]

Current value as a list of external based on the given delimiter character

`RTRExternalValue* RTRExternalValue::duplicate ()` [inherited]

Return a copy of this instance. The caller is responsible for memory managing the returned object.

RTRLListOfExternalValue - Assignment operator

Functions

- `RTRLListOfExternalValue & operator= (const RTRLListOfExternalValue &lev)`

RTRLListOfExternalValue - Iteration

Functions

- `void RTRLListOfExternalValue::start ()`
- `void RTRLListOfExternalValue::forth ()`
- `RTRBOOL RTRLListOfExternalValue::off ()`

Function Documentation

`void RTRLListOfExternalValue::start ()` [inherited]

Move to first external value.

`void RTRLListOfExternalValue::forth ()` [inherited]

Move to next external value.

REQUIRE: not_off: !off()

RTRBOOL RTRListOfExternalValue::off () [inherited]

Is there no item at the current position?

RTRListOfExternalValue - Extraction

Functions

- [RTRExternalValue RTRListOfExternalValue::item \(\)](#)
- [RTRExternalValue RTRListOfExternalValue::iTh \(int index\)](#)

Function Documentation

[RTRExternalValue RTRListOfExternalValue::item \(\)](#) [inherited]

Return item at current position.

REQUIRE: not_off: !off()

[RTRExternalValue RTRListOfExternalValue::iTh \(int index\)](#) [inherited]

Return item at position "index".

REQUIRE: big_enough: index >= 1

REQUIRE: small_enough: index <= [count\(\)](#)

RTRListOfExternalValue - Limits

Functions

- int [RTRListOfExternalValue::count \(\)](#) const

Function Documentation

int RTRListOfExternalValue::count () const [inherited]

Number of items in list.

ENSURE: positive_result: Result > 0

RTRString - Attributes

Functions

- const unsigned int [RTRString::capacity \(\)](#) const
- unsigned int [RTRString::count \(\)](#) const
- RTRBOOL [RTRString::isEmpty \(\)](#) const
- unsigned long [RTRString::hash \(\)](#) const
- int [RTRString::lower \(\)](#) const
- int [RTRString::upper \(\)](#) const

Function Documentation

const unsigned int RTRString::capacity () const [inline, inherited]

The current capacity of this string.

unsigned int RTRString::count () const [inline, inherited]
The number of characters in this string.

RTRBOOL RTRString::isEmpty () const [inline, inherited]
Is this string empty? (result == RTRTRUE implies [count\(\)](#) == 0)

unsigned long RTRString::hash () const [inherited]
A hash value for this string.

int RTRString::lower () const [inline, inherited]
Minimum valid index for accessing this string.

int RTRString::upper () const [inline, inherited]
Maximum valid index for accessing this string.

RTRString - Modify in entirety

Functions

- [RTRString](#) & [RTRString::set](#) (const char *str, unsigned int p1, unsigned int p2)
- [RTRString](#) & [RTRString::set](#) (const char *str, unsigned int n)
- [RTRString](#) & [RTRString::readLine](#) (std::istream &, RTRBOOL skipWhite=1)
- [RTRString](#) & [RTRString::clear](#) ()
- [RTRString](#) & [fromNumeric](#) (int i)
- [RTRString](#) & [fromNumeric](#) (unsigned int i)
- [RTRString](#) & [fromNumeric](#) (long i)
- [RTRString](#) & [fromNumeric](#) (unsigned long i)
- [RTRString](#) & [RTRString::fromNumeric](#) (double i)

Function Documentation

[RTRString](#) & [RTRString::set](#) (const char * *str*, unsigned int *p1*, unsigned int *p2*) [inherited]
Initialize this string to the contents of str, starting a position p1 (0 based) and ending at position p2
REQUIRE: p1 <= p2

[RTRString](#) & [RTRString::set](#) (const char * *str*, unsigned int *n*) [inherited]
Initialize this string to the first n bytes of str.

[RTRString](#) & [RTRString::readLine](#) (std::istream &, RTRBOOL *skipWhite* = 1) [inherited]
Set this string to a line extracted from the given stream.

[RTRString](#) & [RTRString::clear](#) () [inline, inherited]
Empty this string.
ENSURE: [isEmpty\(\)](#)

[RTRString](#) & [RTRString::fromNumeric](#) (double *i*) [inherited]
Set this string to the ASCII representation of i.

RTRString - Modify in part

Functions

- char & [RTRString::operator\[\]](#) (int i)
- [RTRString](#) & **prepend** (const char *)
- [RTRString](#) & **prepend** (char)
- [RTRString](#) & **prepend** (long)
- [RTRString](#) & **prepend** (unsigned long)
- [RTRString](#) & [RTRString::prepend](#) (double)
- [RTRString](#) & **append** (const char *)
- [RTRString](#) & **append** (const char *, int)
- [RTRString](#) & **append** (const [RTRString](#) &)
- [RTRString](#) & **append** (const char)
- [RTRString](#) & **append** (const unsigned char)
- [RTRString](#) & **append** (const short n)
- [RTRString](#) & **append** (const unsigned short n)
- [RTRString](#) & **append** (const int n)
- [RTRString](#) & **append** (const unsigned int n)
- [RTRString](#) & **append** (const long n)
- [RTRString](#) & **append** (const unsigned long n)
- [RTRString](#) & **append** (const float n)
- [RTRString](#) & [RTRString::append](#) (const double n)
- [RTRString](#) & [RTRString::toLower](#) ()
- [RTRString](#) & [RTRString::toUpper](#) ()

Function Documentation

char& RTRString::operator[] (int i) [inherited]

Set the i'th character in this string.

REQUIRE: i >= [lower\(\)](#)

REQUIRE: i <= [upper\(\)](#)

[RTRString](#)& RTRString::prepend (double) [inherited]

Prepend the given value to this string.

[RTRString](#)& RTRString::append (const double n) [inherited]

Append the given value to this string.

[RTRString](#)& RTRString::toLower () [inherited]

Put this string in lower case.

[RTRString](#) & [RTRString::toUpper\(\)](#) [inherited]

Put this string in upper case.

RTRString - Truncate**Functions**

- void [RTRString::leftAdjust\(\)](#)
- void [RTRString::rightAdjust\(\)](#)
- [RTRString](#) & [RTRString::head](#) (unsigned int n)
- [RTRString](#) & [RTRString::tail](#) (unsigned int n)

Function Documentation**[void RTRString::leftAdjust\(\)](#) [inherited]**

Remove leading white-space from this string.

[void RTRString::rightAdjust\(\)](#) [inherited]

Remove trailing white-space from this string.

[RTRString](#) & [RTRString::head](#) (unsigned int n) [inherited]

Trim this string to the first n characters.

ENSURE: [count\(\)](#) = n

[head](#)(0) implies [isEmpty\(\)](#)

[RTRString](#) & [RTRString::tail](#) (unsigned int n) [inherited]

Trim the first [count\(\)](#) - n characters from this string.

ENSURE: [count\(\)](#) = n

[tail](#)(0) implies [isEmpty\(\)](#)

RTRString - Comparison**Functions**

- int [RTRString::compare](#) (const char *) const

Function Documentation**[int RTRString::compare](#) (const char *) const [inherited]**

Is this string greater than (result == 1), equal to (result == 0), or less than (result == -1) the given string?

RTRString - Access**Functions**

- char [RTRString::operator\[\]](#) (int i) const
- [RTRString::operator const char *](#) () const
- [RTRString](#) [RTRString::subString](#) (int p1, int p2)
- const char * [RTRString::to_c](#) () const

Function Documentation

char RTRString::operator[] (int *i*) const [inherited]

The *i*'th character in this string.

REQUIRE: *i* >= [lower\(\)](#) **REQUIRE :** *i* <= [upper\(\)](#)

RTRString::operator const char * () const [inherited]

A pointer to the storage for this string.

Result is null terminated (i.e. result[[count\(\)](#)] == "\").

Note: Nulls may be imbedded in data.

[RTRString](#) RTRString::subString (int *p1*, int *p2*) [inherited]

A new string which characters from positions *p1* through *p2*

REQUIRE: *p1* >= [lower\(\)](#)

REQUIRE: *p2* <= [upper\(\)](#)

REQUIRE: *p1* <= *p2*

const char * RTRString::to_c () const [inline, inherited]

A pointer to the internal storage.

Note: unlike use of the cast operator (const char *) storage is not null terminated by this call.

RTRString - Query

Functions

- RTRBOOL [RTRString::contains](#) (const char *) const
- RTRBOOL [RTRString::contains](#) (const char) const
- int [RTRString::indexOf](#) (char *c*, int *p1*)

Function Documentation

RTRBOOL RTRString::contains (const char *) const [inherited]

Does this string contain a sub-string equal to the given string?

RTRBOOL RTRString::contains (const *char*) const [inherited]

Does this string contain the given character?

int RTRString::indexOf (char *c*, int *p1*) [inherited]

The index of the first instance *c* found in this string after position *p1*

REQUIRE: *p1* >= [lower\(\)](#)

REQUIRE: *p1* <= [upper\(\)](#)

ENSURE: result >= [lower\(\)](#) implies operator[](result) == *c*

RTRString - Transform

Functions

- int [RTRString::toInteger](#) () const
- float [RTRString::toFloat](#) () const

- double [RTRString::toDouble](#) () const
- RTRBOOL [RTRString::toBoolean](#) () const

Function Documentation

int RTRString::toInteger () const [inherited]
This string as an integer.

float RTRString::toFloat () const [inherited]
This string as a float.

double RTRString::toDouble () const [inherited]
This string as a double.

RTRBOOL RTRString::toBoolean () const [inherited]
This string as a boolean.

RTRString - Operators

Functions

- [RTRString](#) & **operator=** (const char *)
- [RTRString](#) & [RTRString::operator=](#) (const [RTRString](#) &)
- RTRBOOL **operator==** (const char *) const
- RTRBOOL **operator==** (const [RTRString](#) &) const
- RTRBOOL **operator!=** (const char *) const
- RTRBOOL **operator!=** (const [RTRString](#) &) const
- RTRBOOL **operator>** (const char *) const
- RTRBOOL **operator>** (const [RTRString](#) &) const
- RTRBOOL **operator>=** (const char *) const
- RTRBOOL **operator>=** (const [RTRString](#) &) const
- RTRBOOL **operator<** (const char *) const
- RTRBOOL **operator<** (const [RTRString](#) &) const
- RTRBOOL **operator<=** (const char *) const
- RTRBOOL [RTRString::operator<=](#) (const [RTRString](#) &) const
- [RTRString](#) & **operator+=** (const char *)
- [RTRString](#) & **operator+=** (const [RTRString](#) &)
- [RTRString](#) & [RTRString::operator+=](#) (const char)

Function Documentation

[RTRString](#) & RTRString::operator= (const [RTRString](#) &) [inherited]
Assign this string to other string.

RTRBOOL RTRString::operator<= (const [RTRString](#) &) const [inherited]

Compare this string with other

[RTRString](#) & **[RTRString::operator+=](#)** (const *char*) [*inherited*]
Append other string or character to this string.

RTRString - Operations

Functions

- void [RTRString::grow](#) (unsigned int *n*)
- void [RTRString::trim](#) (unsigned int)
- void [RTRString::setCount](#) (unsigned int *i*)

Function Documentation

void [RTRString::grow](#) (unsigned int *n*) [*inherited*]
Increase the capacity of this string to accomodate *n* bytes.

ENSURE: [capacity\(\)](#) >= *n*

void [RTRString::trim](#) (unsigned int) [*inherited*]
Decrease the capacity of this string to accomodate *n* bytes.

ENSURE: [capacity\(\)](#) <= *n* [count\(\)](#) <= *n*

void [RTRString::setCount](#) (unsigned int *i*) [*inherited*]
Set count to *i*.
[Useful when using the string storage (via [to_c\(\)](#)) as a buffer]

REQUIRE: *i* <= [capacity\(\)](#)

RTRString - OBSOLETE

Functions

- RTRBOOL [RTRString::isEqual](#) (const char *) const
- [RTRString](#) & [RTRString::empty](#) ()
- int [RTRString::length](#) () const
- int [RTRString::index](#) (char *c*, int *start*)
- [RTRString](#) & **set** ([RTRString](#) &, unsigned int *p1*, unsigned int *p2*)
- [RTRString](#) & [RTRString::fromInteger](#) (int *i*)
- void **appendNumeric** (const char)
- void **appendNumeric** (const unsigned char)
- void **appendNumeric** (const short *n*)
- void **appendNumeric** (const unsigned short *n*)
- void **appendNumeric** (const int *n*)
- void **appendNumeric** (const unsigned int *n*)
- void **appendNumeric** (const long *n*)

- void **appendNumeric** (const unsigned long n)
- void **appendNumeric** (const float n)
- void [RTRString::appendNumeric](#) (const double n)

Function Documentation

RTRBOOL RTRString::isEqual (const char *) const [inherited]
Use [compare\(\)](#)

[RTRString](#) & RTRString::empty () [inline, inherited]
Use [clear\(\)](#)

int RTRString::length () const [inline, inherited]
Use [count\(\)](#)

int RTRString::index (char c, int start) [inherited]
Use [indexOf](#)

[RTRString](#) & RTRString::fromInteger (int i) [inherited]
Use [fromNumeric](#)

void RTRString::appendNumeric (const double n) [inline, inherited]
Use [append](#)

RTRSelectNotifier - From RTREventNotifier

Functions

- void [RTRSelectNotifier::enable](#) ()
- void [RTRSelectNotifier::disable](#) ()

Function Documentation

void RTRSelectNotifier::enable () [virtual, inherited]
Enable the notifier. Left here for compatibility. The "enable" of a notifier is implementation specific.

Implements [RTREventNotifier](#). **void RTRSelectNotifier::disable ()** [virtual, inherited]
Stop dispatching events.

NOTE: This causes control to return to the context which "started" the notifier. Exact behaviour of this is implementation specific. This feature provided as a convenience for simple programs and for debugging. In general components should not "stop" an application. Descendant implementations should, in general, cease operation when there are not IO clients and no timers pending.

Implements [RTREventNotifier](#).

RTRSelectNotifier - From RTREventNotifierImp

Functions

- void [RTRSelectNotifier::enableTimer](#) (long seconds, int milliseconds)
- void [RTRSelectNotifier::disableTimer](#) ()

- void enableReadNotification (int fd)
- void disableReadNotification (int fd)
- void enableWriteNotification (int fd)
- void disableWriteNotification (int fd)
- void enableExceptNotification (int fd)
- void disableExceptNotification (int fd)

Function Documentation

void RTRSelectNotifier::enableTimer (long *seconds*, int *milliseconds*) [inherited]

Establish a timer for the given time, canceling any previous timer.

void RTRSelectNotifier::disableTimer () [inherited]

Cancel an installed timer.

RTRShmMOServerMemPool - State

Functions

- RTRBOOL **error** () const

RTRShmMOServerMemPool - Attributes

Functions

- const [RTRString](#) & [RTRShmMOServerMemPool::text](#) () const
- RTRServerPartition & **partition** ()

Function Documentation

const [RTRString](#) & RTRShmMOServerMemPool::text () const [inline, inherited]

REQUIRE: error()

RTRShmMOServerMemPool - Util

Functions

- void **useStats** ([RTRSharedMemoryStats](#) *)

RTRShmMOServerMemPool - Event processing

Functions

- void [RTRShmMOServerMemPool::processTimerEvent](#) ()
- void pollForMessages ()

Function Documentation

void RTRShmMOServerMemPool::processTimerEvent () [virtual, inherited]

Redefined by descendants to provide specific behaviour for this timer.

Implements [RTRTimerCmd](#).

RTRShmMOServer - State

Functions

- RTRBOOL [RTRShmMOServer::enabled](#) () const
- RTRBOOL [RTRShmMOServer::error](#) () const

Function Documentation

RTRBOOL RTRShmMOServer::enabled () const [inline, inherited]

Has the [RTRShmMOServerMemPool](#) been instantiated?

RTRBOOL RTRShmMOServer::error () const [inherited]

Is this class or the [RTRShmMOServerMemPool](#) in an error state?

RTRShmMOServer - Identity

Functions

- const [RTRObjectId](#) & [instanceId](#) () const

RTRShmMOServer - Attributes

Functions

- unsigned long [RTRShmMOServer::sharedMemorySize](#) () const
- int [RTRShmMOServer::maxClients](#) () const
- const [RTRString](#) & [text](#) () const
- [RTRShmMOServerMemPool](#) * [RTRShmMOServer::managedObjectServer](#) () const
- [RTRSharedMemoryStats](#) * [memoryStats](#) () const

Function Documentation

unsigned long RTRShmMOServer::sharedMemorySize () const [inline, inherited]

Amount of shared memory allocated for storing managed objects (bytes).

int RTRShmMOServer::maxClients () const [inline, inherited]

The maximum number of clients permitted to access the shared memory segment.

[RTRShmMOServerMemPool](#) * RTRShmMOServer::managedObjectServer () const [inline, inherited]

ENSURE: 0 implies !enabled()

RTRShmMOServer - Operations

Functions

- void [RTRShmMOServer::enable](#) ()
- void [RTRShmMOServer::disable](#) ()

Function Documentation

void RTRShmMOServer::enable () [inherited]
Create an instance of [RTRShmMOServerMemPool](#).

REQUIRE: !enabled()

ENSURE: [enabled\(\)](#)

void RTRShmMOServer::disable () [inherited]
Destroys an instance of [RTRShmMOServerMemPool](#)

REQUIRE: [enabled\(\)](#)

ENSURE: !enabled()

RTRShmProxyManagedObjectClassDirFactory - Operations from RTRProxyManagedObjectDirFactory

Functions

- RTRProxyManagedObjectClassDirectoryPtr **newClassDirectory** (const [RTRObjectId](#) &classFilter) const

RTRShmProxyManagedObjectServerPool - Operations

Functions

- RTRShmProxyManagedObjectServer * [RTRShmProxyManagedObjectServerPool::addServer](#) (const char *key, int pollInterval=1, int handshakeInterval=2) [inherited]
- void [RTRShmProxyManagedObjectServerPool::dropServer](#) (const char *key)

Function Documentation

RTRShmProxyManagedObjectServer* RTRShmProxyManagedObjectServerPool::addServer (const char * *key*, int *pollInterval* = 1, int *handshakeInterval* = 2) [inherited]

Add a server with the shared memory given key to this pool.

void RTRShmProxyManagedObjectServerPool::dropServer (const char * *key*) [inherited]

Drop the server with the given shared memory key from this pool.

RTRShmServer - State

Functions

RTRBOOL [RTRShmServer::enabled](#) () const

RTRBOOL [RTRShmServer::error](#) () const

Function Documentation

RTRBOOL RTRShmServer::enabled () const [inline, inherited]

Has the segment of shared memory been created?

RTRBOOL RTRShmServer::error () const [inherited]

Is this class or the [RTRServerSharedMemoryRoot](#) in an error state?

RTRShmServer - Identity

Functions

- const [RTRObjectId](#) & `instanceId` () const

RTRShmServer - Attributes

Functions

- const [RTRString](#) & [RTRShmServer::sharedMemoryKey](#) () const
- const [RTRString](#) & [RTRShmServer::semaphoreKey](#) () const
- unsigned long [RTRShmServer::sharedMemorySize](#) () const
- int [RTRShmServer::maxClients](#) () const
- int [RTRShmServer::numberOfSemaphores](#) () const
- const [RTRString](#) & `text` () const
- [RTRServerSharedMemoryRoot](#) * [RTRShmServer::sharedMemory](#) () const

Function Documentation

const [RTRString](#) & [RTRShmServer::sharedMemoryKey](#) () const [inline, inherited]
The shared memory key.

const [RTRString](#) & [RTRShmServer::semaphoreKey](#) () const [inline, inherited]
The semaphore key.

unsigned long [RTRShmServer::sharedMemorySize](#) () const [inline, inherited]
The desired size of the shared memory segment (in bytes).

int [RTRShmServer::maxClients](#) () const [inline, inherited]
The desired max number of allowed clients.

int [RTRShmServer::numberOfSemaphores](#) () const [inline, inherited]
The number of semaphores.

[RTRServerSharedMemoryRoot](#) * [RTRShmServer::sharedMemory](#) () const [inline, inherited]
A reference to the [RTRServerSharedMemoryRoot](#).
ENSURE: 0 implies !enabled()

RTRShmServer - Operations

Functions

- void [RTRShmServer::enable](#) ()
- void [RTRShmServer::disable](#) ()

Function Documentation

void [RTRShmServer::enable](#) () [inherited]
REQUIRE: !enabled()

Creates an [RTRServerSharedMemoryRoot](#).

ENSURE: [enabled\(\)](#)

void **RTRShmServer::disable** () [inherited]

REQUIRE: [enabled\(\)](#)

Destroys the [RTRServerSharedMemoryRoot](#).

ENSURE: !enabled()

RTRServerSharedMemoryRoot - Identity

Functions

- const [RTRObjectId](#) & **instanceId** () const

RTRServerSharedMemoryRoot - Attributes

Functions

- const [RTRString](#) & **RTRServerSharedMemoryRoot::text** () const
- [RTRString](#) & **RTRServerSharedMemoryRoot::key** ()
- HANDLE **RTRServerSharedMemoryRoot::id** () const
- RTRSharedMemoryHdr * **RTRServerSharedMemoryRoot::header** () const
- RTRServerSemaphoreSet * **RTRServerSharedMemoryRoot::semaphoreSet** () const

Function Documentation

const [RTRString](#) & **RTRServerSharedMemoryRoot::text** () const [inline, inherited]

Text explaining the state of this shared memory root.

[RTRString](#) & **RTRServerSharedMemoryRoot::key** () [inherited]

Key of this memory segment.

HANDLE **RTRServerSharedMemoryRoot::id** () const [inline, inherited]

The id assigned by the system to this memory allocation.

REQUIRE: !error()

RTRSharedMemoryHdr * **RTRServerSharedMemoryRoot::header** () const [inline, inherited]

The header overlaid on the memory segment.

RTRServerSemaphoreSet * **RTRServerSharedMemoryRoot::semaphoreSet** () const [inline, inherited]

The set of semaphores associated with this segment.

RTRServerSharedMemoryRoot - State

Functions

- virtual RTRBOOL **RTRServerSharedMemoryRoot::error** () const

Function Documentation

virtual RTRBOOL **RTRServerSharedMemoryRoot::error** () const [virtual, inherited]

Is this segment in an error state?

RTRServerSharedMemoryRoot - Access

Functions

- RTRSharedMemoryPartitionIterator **partitionIterator** () const

RTRServerSharedMemoryRoot - Event processing

Functions

- void RTRServerSharedMemoryRoot::processTimerEvent ()

Function Documentation

void RTRServerSharedMemoryRoot::processTimerEvent () [virtual, inherited]

Redefined by descendants to provide specific behaviour for this timer.

Implements [RTRTimerCmd](#).

RTRTimerCmd - Attributes

Functions

- const RTRTimeInterval & [RTRTimerCmd::timeOfEvent](#) () const
- long [RTRTimerCmd::offsetSeconds](#) () const
- short [RTRTimerCmd::offsetMilliseconds](#) () const

Function Documentation

const RTRTimeInterval & RTRTimerCmd::timeOfEvent () const [inline, inherited]

System time when event will expire. Set by [activate\(\)](#).

long RTRTimerCmd::offsetSeconds () const [inline, inherited]

Relative time of this event in seconds.

short RTRTimerCmd::offsetMilliseconds () const [inline, inherited]

Relative offset in milliseconds.

RTRTimerCmd - Comparison

Functions

- RTRBOOL **operator==** ([RTRTimerCmd](#) &) const
- RTRBOOL **operator<** ([RTRTimerCmd](#) &) const
- RTRBOOL **operator<=** ([RTRTimerCmd](#) &) const
- RTRBOOL **operator>** ([RTRTimerCmd](#) &) const
- RTRBOOL **operator>=** ([RTRTimerCmd](#) &) const

RTRTimerCmd - State

Functions

- RTRBOOL [RTRTimerCmd::active](#) () const

Function Documentation

RTRBOOL RTRTimerCmd::active () const [inline, inherited]
Is cmd() current installed in the notifier?

RTRTimerCmd - Operations

Functions

- void [RTRTimerCmd::setTimerOffset](#) (long s, short m)
- void [RTRTimerCmd::activate](#) ()
- void [RTRTimerCmd::deactivate](#) ()

Function Documentation

void RTRTimerCmd::setTimerOffset (long s, short m) [inline, inherited]
Set timer offset to s seconds and m milliseconds.

void RTRTimerCmd::activate () [inherited]
REQUIRE: ! [active\(\)](#)
ENSURE: [active\(\)](#)

void RTRTimerCmd::deactivate () [inherited]
REQUIRE: [active\(\)](#)
ENSURE: ![active\(\)](#)

RTRTimerCmd - Event processing

Functions

- virtual void [RTRTimerCmd::processTimerEvent](#) ()=0

Function Documentation

virtual void RTRTimerCmd::processTimerEvent () [pure virtual, inherited]
Redefined by descendants to provide specific behaviour for this timer.

Implemented in [RTRShmMOServerMemPool](#), [RTRSharedMemoryStats](#), and [RTRServerSharedMemoryRoot](#).

RTRTimerCmd - Implementation

Functions

- RTRTimeInterval & [RTRTimerCmd::eventTime](#) ()

Function Documentation

RTRTimeInterval & RTRTimerCmd::eventTime () [inline, inherited]

Reset the time the event will expire.

RTRWindowsNotifier - Control from RTREventNotifier

Functions

- void [RTRWindowsNotifier::enable](#) ()
- void [RTRWindowsNotifier::disable](#) ()

Function Documentation

void RTRWindowsNotifier::enable () [virtual, inherited]

Enable the notifier. Note that the notifier is enabled automatically when created, so this is not needed.

OBSOLETE - Left for compatibility with older versions.

Implements [RTREventNotifier](#). **void RTRWindowsNotifier::disable ()** [virtual, inherited]

Stop dispatching events. In general, components should not use this. Instead, disable I/O notification on any registered file descriptors and cancel timers. Then when the application exits, the notifier destructor will be called and clean up. Since everything registered with the notifier has already been unregistered, there will be no callbacks while the process is cleaning up objects. Use of disable merely avoids the requirement of unregistering before exiting.

Implements [RTREventNotifier](#).

RTRWindowsNotifier - Control from RTREventNotifierImp

Functions

- void [RTRWindowsNotifier::enableTimer](#) (long seconds, int milliseconds)
- virtual void [RTRWindowsNotifier::disableTimer](#) ()
- void enableReadNotification (int fd)
- void disableReadNotification (int fd)
- void enableWriteNotification (int fd)
- void disableWriteNotification (int fd)
- void enableExceptNotification (int fd)
- void disableExceptNotification (int fd)

Function Documentation

void RTRWindowsNotifier::enableTimer (long seconds, int milliseconds) [inherited]

Establish a timer for the given time, canceling any previous timer.

virtual void RTRWindowsNotifier::disableTimer () [virtual, inherited]

Cancel any previous timer.

RTRXFileDb - State

Functions

- RTRBOOL [RTRXFileDb::error](#) () const
- const char * [RTRXFileDb::errorText](#) () const

Function Documentation

RTRBOOL RTRXFileDb::error () const [virtual, inherited]

Is the config db in an error state?

Implements [RTRConfigDb](#).const char* RTRXFileDb::errorText () const [virtual, inherited]

Explanation for the error.

Implements [RTRConfigDb](#).

RTRXFileDb - Query

Functions

- RTRBOOL [RTRXFileDb::has](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &varName) const

Function Documentation

RTRBOOL RTRXFileDb::has (const [RTRObjectId](#) & classId, const [RTRObjectId](#) & instanceId, const [RTRString](#) & varName) const [virtual, inherited]

Does db contain a variable corresponding to the class identifier and instance identifier with the given variable name?

Synchronized

REQUIRE: !error()

Implements [RTRConfigDb](#).

RTRXFileDb - Access

Functions

- [RTRConfigVariable RTRXFileDb::variable](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &varName, const [RTRString](#) &dflt) const
- [RTRConfigVariable RTRXFileDb::variable](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &varName) const

Function Documentation

[RTRConfigVariable RTRXFileDb::variable](#) (const [RTRObjectId](#) & classId, const [RTRObjectId](#) & instanceId, const [RTRString](#) & varName, const [RTRString](#) & dflt) const [virtual, inherited]

The variable corresponding to the given class identifier and instance identifier with the given variable name. If no value is found, default will be used.

Synchronized

REQUIRE: not_in_error: !error()

ENSURE: has(classId, instanceId, varName)? !Result.[error\(\)](#) : 1

ENSURE: !has(classId, instanceId, varName)? !Result.[error\(\)](#): 1

Implements [RTRConfigDb.RTRConfigVariable RTRXFileDb::variable](#) (const [RTRObjectId](#) & classId, const [RTRObjectId](#) & instanceId, const [RTRString](#) & varName) const [virtual, inherited]

The variable corresponding to the given class identifier and instance identifier with the given variable name. If no value is found for these id's, the returned config var will has [error\(\)](#) set.

Synchronized

REQUIRE: not_in_error: !error()

ENSURE: has(classId, instanceId, varName)? !Result.[error\(\)](#) : 1

ENSURE: !has(classId, instanceId) ? Result.error(): 1

Implements [RTRConfigDb](#).

RTRXFileDb - OBSOLETE

Functions

- [RTRConfigVariable RTRXFileDb::value](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &varName, const [RTRString](#) &dflt) const
- [RTRConfigVariable RTRXFileDb::value](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &varName) const

Function Documentation

[RTRConfigVariable RTRXFileDb::value](#) (const [RTRObjectId](#) & classId, const [RTRObjectId](#) & instanceId, const [RTRString](#) & varName, const [RTRString](#) & dflt) const [virtual, inherited]
Synchronized

Implements [RTRConfigDb.RTRConfigVariable RTRXFileDb::value](#) (const [RTRObjectId](#) & classId, const [RTRObjectId](#) & instanceId, const [RTRString](#) & varName) const [virtual, inherited]
Synchronized

Implements [RTRConfigDb](#).

RTRXFileDb - File features

Functions

- void [RTRXFileDb::load](#) (const char *fileName)

Function Documentation

void [RTRXFileDb::load](#) (const char * *fileName*) [inherited]
Synchronized

Chapter 5 Refinitiv Management Classes Class Documentation

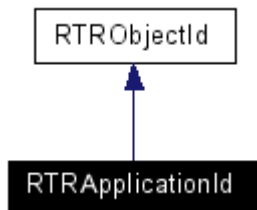
RTRApplicationId Class Reference

The class RTRApplicationId is a descendant of [RTRObjectId](#) that set's itself up as <hostname>.<instance>.<appname> where appName is the name of this application (typically the name of the executable) and instance is a numeric identifier which uniquely identifies this executable from other similar executables on the same host.

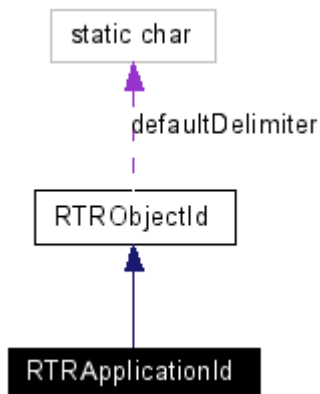
```
#include <appid.h>
```

Inherits [RTRObjectId](#).

Inheritance diagram for RTRApplicationId:



Collaboration diagram for RTRApplicationId:



Public Member Functions

- [RTRApplicationId](#) (int argc, char **argv)
- [RTRApplicationId](#) (int argc, char **argv, const char *appName)
- [RTRApplicationId](#) (const char *instance, const char *appName)
- virtual [~RTRApplicationId](#) ()

Detailed Description

The class RTRApplicationId is a descendant of [RTRObjectId](#) that set's itself up as <hostname>.<instance>.<appname> where appName is the name of this application (typically the name of the executable) and instance is a numeric identifier which uniquely identifies this executable from other similar executables on the same host.

An instance of RTRApplicationId is used as the root context for all object identifiers in application.

```
RTRApplicationId appId1(argc, argv);
```

```
RTRApplicationId appId2(argc, argv, "applicationName");
RTRApplicationId appId3("instance1", "applicationName");
```

See Also:

[RTRObjctId](#)

Constructor & Destructor Documentation

RTRApplicationId::RTRApplicationId (int *argc*, char ** *argv*)

Construct an id of the form <hostname>.<instance>.<appname>, where hostname is retrieved from the system, appName is the name of the executable (from the command line) and instance is extracted from the argv argument vector (using -instance, or -Instance, or -INSTANCE).

RTRApplicationId::RTRApplicationId (int *argc*, char ** *argv*, const char * *appName*)

Construct an id of the form <hostname>.<instance>.<appname>, where hostname is retrieved from the system and instance is extracted from the argv argument vector. (using -instance, or -Instance, or -INSTANCE).

RTRApplicationId::RTRApplicationId (const char * *instance*, const char * *appName*)

Construct an id of the form <hostname>.<instance>.<appname>, using the given instance and application name and retrieving hostname from the system.

virtual RTRApplicationId::~~RTRApplicationId () [virtual]

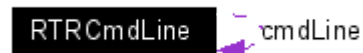
Destructor

RTRCmdLine Class Reference

It is assumed that there is only one instance of RTRCmdLine in an application. That instance should be accessed from RTRCmdLine::cmdLine. RTRCmdLine provides built in help flag (tag is "?"). Unused elements of argv are available from leftOvers() (a list of RTRCmdLineData).

```
#include <cmdline.h>
```

Collaboration diagram for RTRCmdLine:



Public Member Functions

- [RTRCmdLine](#) ()
- [~RTRCmdLine](#) ()
- const RTRDLinkedList< RTRCmdLineData, RTRDLink0 > & [leftOvers](#) () const
- RTRBOOL [error](#) () const
- RTRBOOL [resolved](#) () const
- void [resolve](#) (int argc, char **argv)
- void [printUsage](#) (std::ostream &, const char *argv0) const

Static Public Attributes

- static [RTRCmdLine cmdLine](#)

Friends

- class [RTRCmdLineArg](#)

Detailed Description

It is assumed that there is only one instance of RTRCmdLine in an application. That instance should be accessed from [RTRCmdLine::cmdLine](#). RTRCmdLine provides built in help flag (tag is "?"). Unused elements of argv are available from [leftOvers\(\)](#) (a list of RTRCmdLineData).

```
RTRCmdLine RTRCmdLine::cmdLine; // Create static first
//other includes

int main(int argc, char **argv)
{
    //Declare command line arguments
    //...

    //resolve the command line
    RTRCmdLine::cmdLine.resolve(argc, argv);

    //check for errors
    if ( RTRCmdLine::cmdLine.error() )
    {
        RTRCmdLine::cmdLine.printUsage(cerr, argv[0]);
        return -1;
    }

    //...
```

See Also:

RTRCmdLine, [RTRCmdLineArg](#), RTRCmdLineFlag, RTRCmdLineList, RTRCmdLineNumeric, [RTRCmdLineString](#), RTRCmdLineData

Constructor & Destructor Documentation

RTRCmdLine::RTRCmdLine ()
 Constructor

RTRCmdLine::~~RTRCmdLine ()
 Destructor

Member Function Documentation

const RTRDLinkList< RTRCmdLineData, RTRDLink0 > & RTRCmdLine::leftOvers () const [inline]
 Access

arguments found on the command line that did not map to a [RTRCmdLineArg](#). RTRCmdLineData is a descendant of [RTRString](#).
 arguments found on the command line that did not map to a [RTRCmdLineArg](#). RTRCmdLineData is a descendant of [RTRString](#).

RTRBOOL RTRCmdLine::error () const
 State
 Was there an error when parsing the command line?

RTRBOOL RTRCmdLine::resolved () const [inline]
 State
 Has this command line been resolved?

void RTRCmdLine::resolve (int argc, char ** argv)
 Operations

parse the command line parameters. Assign values to descendants of [RTRCmdLineArg](#) that have been created.

REQUIRE: `!resolved()`

ENSURE: `resolved()`

`void RTRCmdLine::printUsage (std::ostream &, const char * argv0) const`
Operations

called when '-' option is specified on the command line

Member Data Documentation

[RTRCmdLine](#) [RTRCmdLine::cmdLine](#) [static]
The command line

RTRCmdLineArg Class Reference

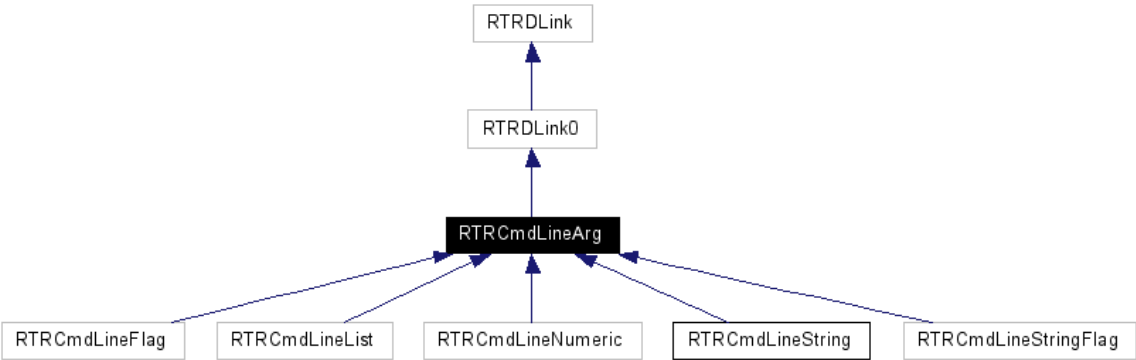
RTRCmdLineArg is the base class for command line arguments. This class includes the base constructor, accessor methods, state checking methods, and stringValue() for getting a RTRString for the argument's value.

```
#include <cmdline.h>
```

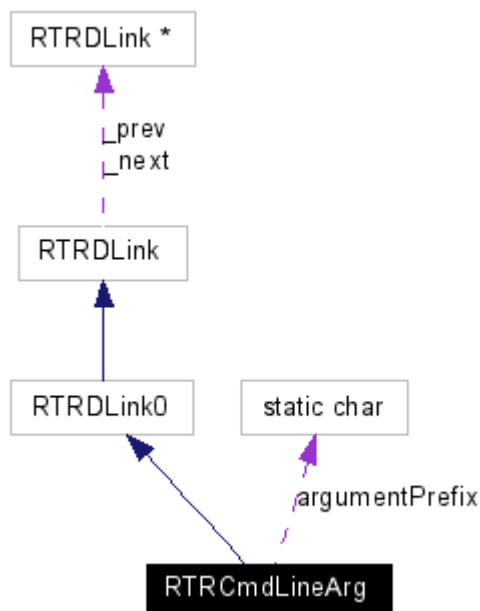
Inherits RTRDLink0.

Inherited by RTRCmdLineFlag, RTRCmdLineList, RTRCmdLineNumeric, [RTRCmdLineString](#), and RTRCmdLineStringFlag.

Inheritance diagram for RTRCmdLineArg:



Collaboration diagram for RTRCmdLineArg:



Public Types

- enum [Required](#) { **False** = 0, **True** = 1 }

Public Member Functions

- [RTRCmdLineArg](#) (const char *tag, const char *name, const char *purpose, const char *defaultValue, [Required](#) required=True)
- virtual [~RTRCmdLineArg](#) ()
- [Required](#) [required](#) () const
- RTRBOOL [hidden](#) () const
- const [RTRString](#) & [tag](#) () const
- const [RTRString](#) & [name](#) () const
- const [RTRString](#) & [defaultValue](#) () const
- const [RTRString](#) & [purpose](#) () const
- RTRBOOL **valid** () const
- RTRBOOL **error** () const
- const [RTRString](#) & [stringValue](#) () const
- RTRBOOL **hasDefault** () const
- virtual void **printShortUsage** (std::ostream &) const
- virtual void **printLongUsage** (std::ostream &) const
- virtual void **printLongUsage** (std::ostream &, int) const
- virtual void **resolve** (RTRDLinkList< RTRCmdLineData, RTRDLink0 > &)
- virtual void [hide](#) ()

Static Public Attributes

- static char argumentPrefix

Friends

- class [RTRCmdLine](#)

Detailed Description

RTRCmdLineArg is the base class for command line arguments. This class includes the base constructor, accessor methods, state checking methods, and [stringValue\(\)](#) for getting a [RTRString](#) for the argument's value.

See Also:

[RTRCmdLine](#), RTRCmdLineData, RTRCmdLineFlag, RTRCmdLineList, RTRCmdLineNumeric, [RTRCmdLineString](#), RTRDLink

Member Enumeration Documentation

enum [RTRCmdLineArg::Required](#)
Enumeration

Constructor & Destructor Documentation

RTRCmdLineArg::RTRCmdLineArg (const char * *tag*, const char * *name*, const char * *purpose*, const char * *defaultValue*, [Required](#) *required* = True)
Constructor

virtual RTRCmdLineArg::~~RTRCmdLineArg () [virtual]
Destructor

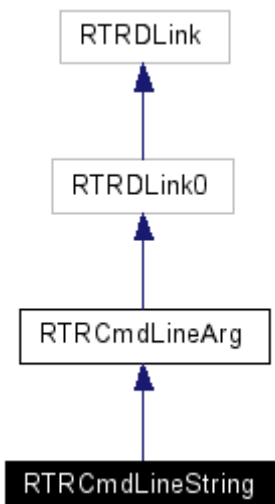
RTRCmdLineString Class Reference

Descendant of RTRCmdLineArg which provides type checking. Strings provide cast operator to RTRString.

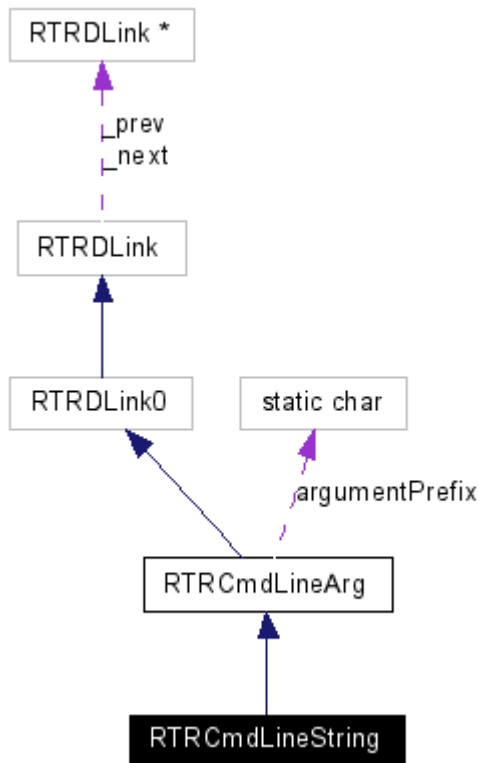
```
#include <cmdline.h>
```

Inherits [RTRCmdLineArg](#).

Inheritance diagram for RTRCmdLineString:



Collaboration diagram for RTRCmdLineString:



Public Member Functions

- [RTRCmdLineString](#) (const char *tag, const char *name, const char *purpose, [Required](#) req=True)
- [RTRCmdLineString](#) (const char *tag, const char *name, const char *purpose, const char *defaultValue, [Required](#) req=True)
- virtual [~RTRCmdLineString](#) ()
- [operator const RTRString &](#) () const

Detailed Description

Descendant of [RTRCmdLineArg](#) which provides type checking. Strings provide cast operator to [RTRString](#).

```

int main(int argc, char **argv)
{
    RTRCmdLineString string("config", "file_name", "load the given file", "default");

    //resolve and check command line

    RTRString &value = string;

    //...
  
```

See Also:

[RTRCmdLine](#)

Constructor & Destructor Documentation

RTRCmdLineString::RTRCmdLineString (const char * *tag*, const char * *name*, const char * *purpose*, [Required](#) req = True)
 Constructor

RTRCmdLineString::RTRCmdLineString (const char * *tag*, const char * *name*, const char * *purpose*, const char * *defaultValue*, [Required](#) req = True)
 Constructor

virtual RTRCmdLineString::~~RTRCmdLineString () [virtual]
 Destructor

Member Function Documentation

RTRCmdLineString::operator const [RTRString](#) & () const
 Transformation
 REQUIRE: valid();

RTRConfig Class Reference

Provides "global" access to a configuration database via the static function [configDb\(\)](#). By default, the available database will be an instance of RTRDefaultConfigDb. The application can override this by instantiating some other type of [RTRConfigDb](#) and "installing" it using the [setConfigDb\(\)](#) function.

```
#include <config.h>
```

Static Public Member Functions

- static void [setConfigDb](#) (const [RTRConfigDb](#) &db)
- static const [RTRConfigDb](#) & [configDb](#) ()

Detailed Description

Provides "global" access to a configuration database via the static function [configDb\(\)](#). By default, the available database will be an instance of RTRDefaultConfigDb. The application can override this by instantiating some other type of [RTRConfigDb](#) and "installing" it using the [setConfigDb\(\)](#) function.

See Also:

[RTRConfigDb](#), [RTRConfigVariable](#), RTRDebugConfig, RTRDefaultConfigDb

Member Function Documentation

static void RTRConfig::setConfigDb (const [RTRConfigDb](#) & *db*) [static]
 Operations
 Set the global config database.
 REQUIRE: !db.error()

static const [RTRConfigDb](#)& RTRConfig::configDb () [static]
 Access

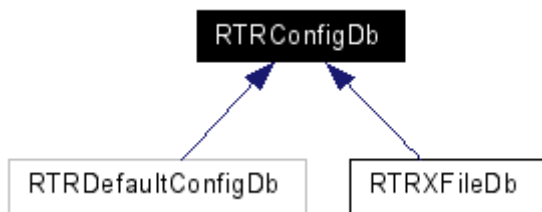
RTRConfigDb Class Reference

This class provides an abstract definition of a configuration database from which configuration variables may be obtained.


```
#include <cfgdb.h>
```

Inherited by `RTRDefaultConfigDb`, and [RTRXFileDb](#).

Inheritance diagram for `RTRConfigDb`:



Public Member Functions

- [RTRConfigDb](#) ()
- virtual [~RTRConfigDb](#) ()
- virtual `RTRBOOL` [error](#) () const =0
- virtual const char * [errorText](#) () const =0
- virtual `RTRBOOL` [has](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &varName) const =0
- virtual [RTRConfigVariable](#) [variable](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &varName, const [RTRString](#) &dflt) const =0
- virtual [RTRConfigVariable](#) [variable](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &varName) const =0
- virtual [RTRConfigVariable](#) [value](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &varName, const [RTRString](#) &dflt) const =0
- virtual [RTRConfigVariable](#) [value](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &varName) const =0

Detailed Description

This class provides an abstract definition of a configuration database from which configuration variables may be obtained.

Clients use [has\(\)](#) to determine if a config variable exists in the database and [variable\(\)](#) to obtain a config variable.

There are two ways a client can access a config variable. The client may ask for a config variable and provide no default value. If no value is found in the database, the config variable will be in an error state. Alternatively, the client can provide a default value that will be assigned as the value of the config variable if no value is found in the database.

Descendants of this class implement the features used to obtain configuration variables.

Application components which use configuration variables have associated with them both a class identifier and an instance identifier. This allows system components to be configured (by means of variables) on a class basis and on a per instance basis. The precedence of class identifiers relative to instance identifiers is an implementation issue determined by descendants of this class.

```

//RTRConfigDb& configDb;

RTRObjectId classId("Class");
RTRObjectId instanceId("instance");
RTRString name("name");
RTRString default("100");

if ( !configDb.error() )
{
    RTRConfigVariable var = configDb.variable(

```

```
int v = var.toInteger();
}

classId,
instanceId,
name,
default);
```

See Also:

[RTRConfig](#), [RTRConfigVariable](#), [RTRDebugConfig](#), [RTRDefaultConfigDb](#)

Constructor & Destructor Documentation

RTRConfigDb::RTRConfigDb () [inline]
Constructor

virtual RTRConfigDb::~~RTRConfigDb () [virtual]
Destructor

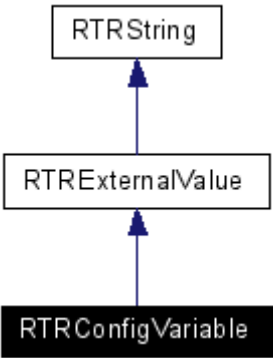
RTRConfigVariable Class Reference

Class RTRConfigVariable offers clients a convenient way to access configuration parameter values. If a value is not available for the parameter, the config variable will be in an error state.

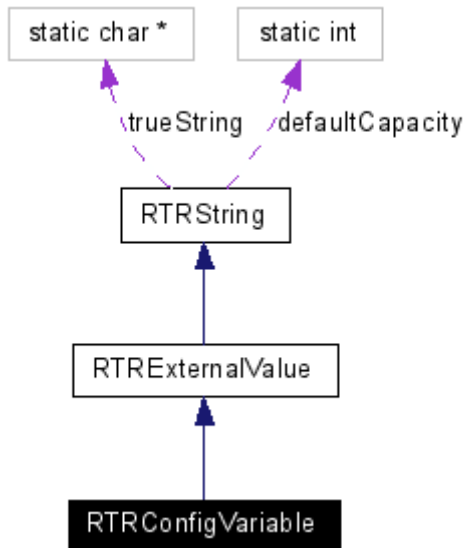
```
#include <cfgvar.h>
```

Inherits [RTRExternalValue](#).

Inheritance diagram for RTRConfigVariable:



Collaboration diagram for RTRConfigVariable:



Public Member Functions

- [RTRConfigVariable](#) ()
- [RTRConfigVariable](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &nm)
- [RTRConfigVariable](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &nm, const [RTRString](#) &val, const [RTRString](#) &dflt)
- [RTRConfigVariable](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &nm, const char *val, int length, const [RTRString](#) &dflt)
- [RTRConfigVariable](#) (const [RTRConfigVariable](#) &var)
- virtual [~RTRConfigVariable](#) ()
- const [RTRString](#) & [defaultValue](#) () const
- [RTRBOOL](#) [isDefaultValue](#) () const
- [RTRBOOL](#) [error](#) () const
- [RTRConfigVariable](#) & [operator=](#) (const [RTRConfigVariable](#) &other)
- [RTRExternalValue](#) [value](#) ()

Detailed Description

Class [RTRConfigVariable](#) offers clients a convenient way to access configuration parameter values. If a value is not available for the parameter, the config variable will be in an error state.

A config variable is typically obtained from a configuration database.

[error\(\)](#) returns TRUE if a value is available for the config variable. [value\(\)](#) returns the value of the config variable as an [RTRExternalValue](#).

```

RTRString value = var;
int i = var.toInteger();
float f = var.toFloat();
RTRBOOL b = var.toBoolean();
  
```

See Also:

[RTRObjectId](#), [RTRConfigDb](#), [RTRXFileDb](#), [RTRExternalValue](#)

Constructor & Destructor Documentation

RTRConfigVariable::RTRConfigVariable ()

Create a config variable that is in error.

For compatibility.

RTRConfigVariable::RTRConfigVariable (const [RTRObjectId](#) & *classId*, const [RTRObjectId](#) & *instanceId*, const [RTRString](#) & *nm*)

Create a config variable that is in error.

RTRConfigVariable::RTRConfigVariable (const [RTRObjectId](#) & *classId*, const [RTRObjectId](#) & *instanceId*, const [RTRString](#) & *nm*, const [RTRString](#) & *val*, const [RTRString](#) & *dflt*)

Create a config variable using the given string.

RTRConfigVariable::RTRConfigVariable (const [RTRObjectId](#) & *classId*, const [RTRObjectId](#) & *instanceId*, const [RTRString](#) & *nm*, const char * *val*, int *length*, const [RTRString](#) & *dflt*)

Create a config variable using the given buffer and length.

RTRConfigVariable::RTRConfigVariable (const [RTRConfigVariable](#) & *var*)

Create a config variable identical to the given config variable.

virtual RTRConfigVariable::~RTRConfigVariable () [virtual]

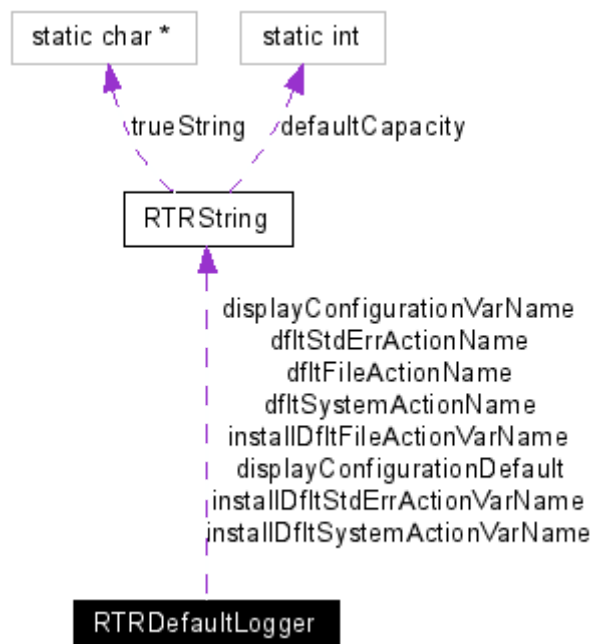
Destructor

RTRDefaultLogger Class Reference

This class provides three types of management event actions (file, std err and system). This logger consults the configuration database, passed on construction, to determine whether it is enabled and whether it is supposed to display its configuration on stdout.

```
#include <dfltlog.h>
```

Collaboration diagram for RTRDefaultLogger:



Public Member Functions

- [RTRDefaultLogger](#) (const [RTRObjectId](#) &appid, const [RTRString](#) &name, const [RTRConfigDb](#) &configDb=RTRConfig::configDb())
- [RTRDefaultLogger](#) (const [RTRObjectId](#) &appid, const [RTRString](#) &name, const [RTRString](#) &appName, const [RTRConfigDb](#) &configDb=RTRConfig::configDb())
- RTRDefaultFileAction * [defaultFileAction](#) ()
- RTRDefaultStdErrAction * [defaultStdErrorAction](#) ()
- RTRDefaultSystemAction * [defaultSystemAction](#) ()
- void [displayConfiguration](#) () const
- void [setSelector](#) ([RTRString](#) &name)

Static Public Attributes

- static [RTRString](#) displayConfigurationDefault
- static [RTRString](#) displayConfigurationVarName
- static [RTRString](#) installDfltFileActionVarName
- static [RTRString](#) installDfltStdErrActionVarName
- static [RTRString](#) installDfltSystemActionVarName
- static [RTRString](#) dfltFileActionName
- static [RTRString](#) dfltStdErrActionName
- static [RTRString](#) dfltSystemActionName

Detailed Description

This class provides three types of management event actions (file, std err and system). This logger consults the configuration database, passed on construction, to determine whether it is enabled and whether it is supposed to display its configuration on stdout.

The default installation results in a stderr logger which is instantiated but disabled, and a file logger which is instantiated and traps events of all severity levels (except debug) from all components.

The instance name of defaultFileAction is "defaultFileAction" and the instance name of the defaultStdErrorAction is "defaultStdErrorAction".

The instance id of each depends on the name given to the logger on its instantiation

e.g. if the name of the logger is "logger" the instance id of the defaultFileAction will be "logger.defaultFileAction".

Class id: SSLDispatcher		Instance Id: assigned from constructor	
Variable Name	Type	Default	Use
enable	Boolean	True	enables event processing
display_configuration	Boolean	False	print configuration to stdout on startup
install_file_action	Boolean	True	installs an instance of RTRDefaultFileAction on startup
install_stderr_action	Boolean	False	installs an instance of RTRDefaultStdErrAction on startup
install_system_action	Boolean	False	installs an instance of RTRDefaultSystemAction on startup

See Also:

RTRDefaultFileAction, RTRDefaultStdErrAction, [RTRMgmtEvent](#), [RTRMgmtAction](#), RTRMgmtEventRouter

Constructor & Destructor Documentation

RTRDefaultLogger::RTRDefaultLogger (const [RTRObjectId](#) & *appid*, const [RTRString](#) & *name*, const [RTRConfigDb](#) & *configDb* = RTRConfig::configDb())

Construct a default logger and associated default log actions as per configuration.

RTRDefaultLogger::RTRDefaultLogger (const [RTRObjectId](#) & *appid*, const [RTRString](#) & *name*, const [RTRString](#) & *appName*, const [RTRConfigDb](#) & *configDb* = RTRConfig::configDb())

Construct a default logger and associated default log actions as per configuration. The appName value will be used by all log actions which require an application name.

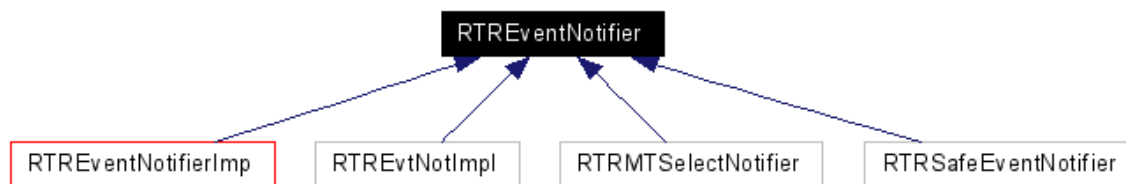
RTREventNotifier Class Reference

RTREventNotifier is the abstract base class for event managers which provide facilities whereby clients can register to receive I/O events. Timing events are also implemented by RTREventNotifier but are made available to clients by means of the [RTRTimerCmd](#) abstraction.

```
#include <evtnotif.h>
```

Inherited by [RTREventNotifierImp](#) [virtual], RTREvtNotImpl [virtual], RTRMTSelectNotifier, and RTRSafeEventNotifier [virtual].

Inheritance diagram for RTREventNotifier:



Public Member Functions

- virtual [~RTREventNotifier](#) ()
- RTRBOOL [isReadClient](#) (RTRIOClient &client, int fd) const
- RTRBOOL [isWriteClient](#) (RTRIOClient &client, int fd) const
- RTRBOOL [isExceptionClient](#) (RTRIOClient &client, int fd) const
- RTRBOOL [hasReadClient](#) (int fd) const
- RTRBOOL [hasWriteClient](#) (int fd) const
- RTRBOOL [hasExceptionClient](#) (int fd) const
- virtual RTRIOClient * [registeredReadClient](#) (int fd) const =0
- virtual RTRIOClient * [registeredWriteClient](#) (int fd) const =0
- virtual RTRIOClient * [registeredExceptionClient](#) (int fd) const =0
- virtual void [addReadClient](#) (RTRIOClient &client, int fd)=0
- virtual void [addWriteClient](#) (RTRIOClient &client, int fd)=0
- virtual void [addExceptionClient](#) (RTRIOClient &client, int fd)=0
- virtual void [dropReadClient](#) (int fd)=0
- virtual void [dropWriteClient](#) (int fd)=0
- virtual void [dropExceptionClient](#) (int fd)=0
- virtual void [disable](#) ()=0
- virtual void [enable](#) ()=0

Friends

- class [RTRTimerCmd](#)

Detailed Description

RTREventNotifier is the abstract base class for event managers which provides facilities whereby clients can register to receive I/O events. Timing events are also implemented by RTREventNotifier but are made available to clients by means of the [RTRTimerCmd](#) abstraction.

The purpose of this class is to allow components to share system resources (IO, timers) in a cooperative way and to do so without being dependent on any particular implementation of a "main loop".

There is only one instance of RTREventNotifier in an application. The type of this instance depends on the design of the application. The way in which a main loop is started is implementation specific, hence the "main()" of an application is typically tied to a particular type of notifier.

Clients of RTREventNotifier need access the "global" notifier. To do this they include the file "rtr/rtrnotif.h" giving access to a static class of type RTREventNotifierInit. This class implements a reference counting scheme which will automatically construct/destroy a notifier

as necessary. This is similar to the mechanism used for making cin and cout when accessing iostreams. Typically, the source file containing main() includes "rtr/rtrnotif.h", meaning that the notifier is always in existence (until program termination).

Descendants of RTREventNotifier are purpose built for a specific application environment. For example, an implementation intended for X applications uses calls to an underlying X Window library. Components which have no direct dependence on X (i.e. don't need access to X events) use the abstract notifier interface and are insulated from dependencies on X. Components which need X events (perhaps they're X specific) can interface with the X notifier directly. The same is true for a Windows based notifier. For non-windowing applications, it is probably appropriate to use an event notifier implementation which is a self contained main loop, perhaps using the select() system call to do its job.

```
//RTRIOClient& client;
//int fd;
RTREventNotifierInit::notifier->addReadClient(client, fd);
RTREventNotifierInit::notifier->dropReadClient(client, fd);
```

See Also:

RTRIOClient, [RTRTimerCmd](#), RTREventNotifierInit, [RTRSelectNotifier](#), RTRXtEventNotifier, RTRXViewEventNotifier, [RTRWindowsNotifier](#)

Constructor & Destructor Documentation

virtual RTREventNotifier::~~RTREventNotifier () [virtual]
Destructor

RTREventNotifierImp Class Reference

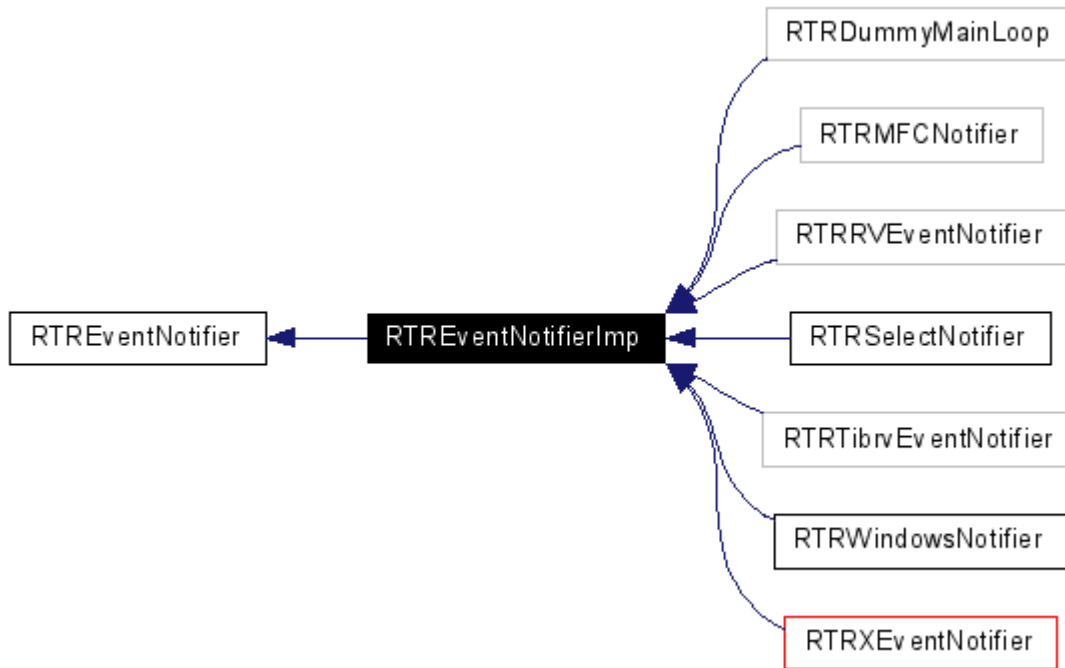
RTREventNotifierImp is an intermediate base class for implementations of the [RTREventNotifier](#) abstraction. It implements most of what is required and leaves some specific implementation details for descendants.

```
#include <enimp.h>
```

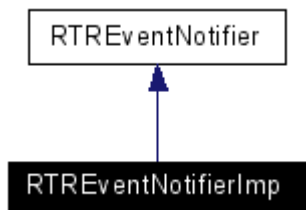
Inherits [RTREventNotifier](#).

Inherited by RTRDummyMainLoop, RTRMFCNotifier, RTRRVEventNotifier, [RTRSelectNotifier](#), RTRTibrvEventNotifier, [RTRWindowsNotifier](#), and [RTRXEventNotifier](#).

Inheritance diagram for RTREventNotifierImp:



Collaboration diagram for RTREventNotifierImp:



Public Member Functions

- [RTREventNotifierImp](#) (int size)
- [~RTREventNotifierImp](#) ()
- virtual `RTRIOClient *` [registeredReadClient](#) (int fd) const
- virtual `RTRIOClient *` [registeredWriteClient](#) (int fd) const
- virtual `RTRIOClient *` [registeredExceptionClient](#) (int fd) const
- virtual void [addReadClient](#) (RTRIOClient &client, int fd)
- virtual void [addWriteClient](#) (RTRIOClient &client, int fd)
- virtual void [addExceptionClient](#) (RTRIOClient &client, int fd)
- virtual void [dropReadClient](#) (int fd)
- virtual void [dropWriteClient](#) (int fd)
- virtual void [dropExceptionClient](#) (int fd)
- void [notifyReadPending](#) (int fd)

- void [notifyWritePending](#) (int fd)
- void [notifyExceptPending](#) (int fd)
- void [expireEvents](#) ()

Detailed Description

RTREventNotifierImp is an intermediate base class for implementations of the [RTREventNotifier](#) abstraction. It implements most of what is required and leaves some specific implementation details for descendants.

This class is only relevant to designers of new implementations of [RTREventNotifier](#).

Note for thread-safety: If an application has multiple threads that share same instance of notifier, then, that notifier must be made MT-safe for its public methods such as [addReadClient\(\)](#) and protected methods add/cancelEvent() which are called by friend class [RTRTimerCmd](#). The thread-safety for this base class here is only meant to synchronize the activities of register/unregister the IO and timer events. It is believed that such kind of multi-thread application hardly make any sense unless in its non-notifier-threads some meaningful things are being accomplished while the notifier-thread is processing its call-back functions, i.e. processIORead() or processTimerCmd(). In this scenario, the application should take the responsibilities to coordinate the calls that are happening in these different threads. One practice is that, a condition-variable is used in the call-back function to signal the non-notifier-thread and invoke actions there therefore the notifier-thread could be non-blocking for its events.

See Also:

RTRXtEventNotifier, RTRXViewEventNotifier, [RTRWindowsNotifier](#), [RTRSelectNotifier](#)

Constructor & Destructor Documentation

RTREventNotifierImp::RTREventNotifierImp (int size)
 Constructor

RTREventNotifierImp::~~RTREventNotifierImp ()
 Destructor

RTRExternalValue Class Reference

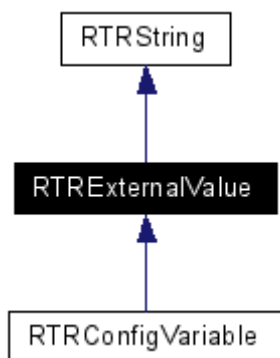
This class provides the same capabilities as the [RTRString](#) class plus the following:

```
#include <rtextval.h>
```

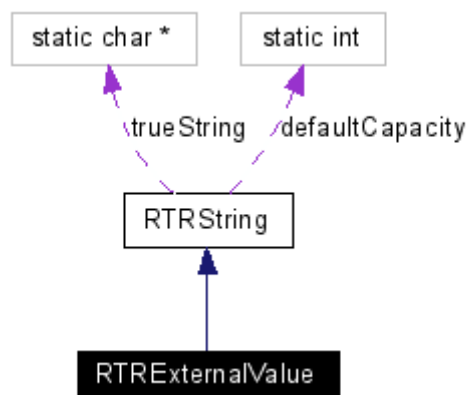
Inherits [RTRString](#).

Inherited by [RTRConfigVariable](#).

Inheritance diagram for RTRExternalValue:



Collaboration diagram for RTRExternalValue:



Public Member Functions

- [RTRExternalValue](#) ()
- [RTRExternalValue](#) (const char *n)
- [RTRExternalValue](#) (const char *n, int i)
- [RTRExternalValue](#) (unsigned int n)
- [RTRExternalValue](#) (const [RTRString](#) &n)
- [RTRExternalValue](#) (const [RTRExternalValue](#) &n)
- virtual [~RTRExternalValue](#) ()
- [RTRExternalValue](#) & **operator=** (const [RTRExternalValue](#) &n)
- int [integer_from_hexidecimal](#) ()
- int [integer_from_octal](#) ()
- RTRBOOL [isTrue](#) ()
- RTRBOOL [isFalse](#) ()
- [RTRLListofExternalValue](#) [list](#) (char delimiter)
- [RTRExternalValue](#) * [duplicate](#) ()

Detailed Description

This class provides the same capabilities as the [RTRString](#) class plus the following:.

- return whether the value is "true" or "false"
- if this value is delimited, return a list of the delimited values based on a given delimiter character
- conversion from integer to hexadecimal or octal value

See [RTRString](#) for a listing of other features.

See Also:

[RTRLListofExternalValue](#)

Constructor & Destructor Documentation

RTRExternalValue::RTRExternalValue ()

An empty value.

RTRExternalValue::RTRExternalValue (const char * *n*)

A copy of the given null terminated string.

RTRExternalValue::RTRExternalValue (const char * *n*, int *i*)

A copy of the first *n* characters of *s*.

RTRExternalValue::RTRExternalValue (unsigned int *n*)

A empty value with capacity *n*.

RTRExternalValue::RTRExternalValue (const [RTRString](#) & *n*)

A copy of the given string.

RTRExternalValue::RTRExternalValue (const [RTRExternalValue](#) & *n*)

A copy of the given value.

virtual RTRExternalValue::~~RTRExternalValue () [virtual]

Destructor

RTRLISTOfExternalValue Class Reference

A simple list of [RTRExternalValue](#) instances.

```
#include <rtextval.h>
```

Public Member Functions

- [RTRLISTOfExternalValue](#) ()
- [RTRLISTOfExternalValue](#) ([RTRExternalValue](#) &ev, char delim)
- [RTRLISTOfExternalValue](#) (const [RTRLISTOfExternalValue](#) &lev)
- virtual [~RTRLISTOfExternalValue](#) ()
- [RTRLISTOfExternalValue](#) & **operator=** (const [RTRLISTOfExternalValue](#) &lev)
- void [start](#) ()
- void [forth](#) ()
- RTRBOOL [off](#) ()
- [RTRExternalValue](#) [item](#) ()
- [RTRExternalValue](#) [iTh](#) (int index)
- int [count](#) () const

Detailed Description

A simple list of [RTRExternalValue](#) instances.

See Also:

[RTRExternalValue](#), [RTRString](#)

Constructor & Destructor Documentation

RTRListOfExternalValue::RTRListOfExternalValue ()

An empty list.

RTRListOfExternalValue::RTRListOfExternalValue ([RTRExternalValue](#) & *ev*, char *delim*)

Create a list of delimited values using the given external value and the given delimiter.

RTRListOfExternalValue::RTRListOfExternalValue (const [RTRListOfExternalValue](#) & *lev*)

Copy constructor

virtual RTRListOfExternalValue::~~RTRListOfExternalValue () [virtual]

Destructor

RTRLock Class Reference

An instance of RTRLock locks a [RTRLockableObj](#) object passed as an argument in constructor and unlocks it when the RTRLock instance is deleted.

```
#include <lock.h>
```

Public Member Functions

- [RTRLock](#) ([RTRLockableObj](#) &)
- [~RTRLock](#) ()

Detailed Description

An instance of RTRLock locks a [RTRLockableObj](#) object passed as an argument in constructor, and unlocks it when the RTRLock instance is deleted.

Dummy implementaions are provided for this class so that for platforms that thread programming is not supported, such as sunOS, or for applications that don't require the library classes to be MT-safe, the calls (contructor and destructor) embeded in the methods of MT-safed library classes will be no-op, thus minimal performance penalty is paid for application.

See Also:

[RTRLockableObj](#)

Constructor & Destructor Documentation

RTRLock::RTRLock ([RTRLockableObj](#) &)

On construction, lock the obj

RTRLock::~~RTRLock ()

On destruction, unlock the obj

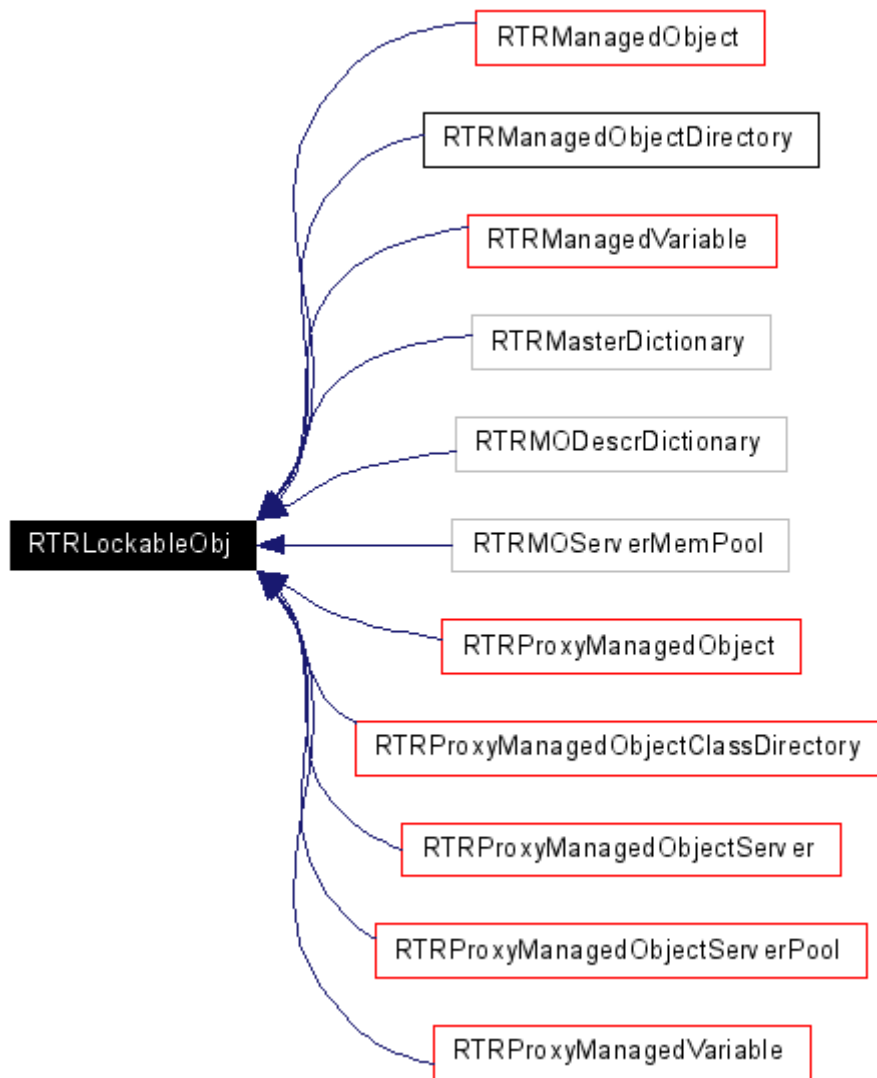
RTRLockableObj Class Reference

RTRLockableObj is a base class representing application component which provides lock/unlock operations on itself perceiving that its states/values could be accessed from multiple threads in applications thus need to be synchronized. Any component that want to be made MT-safe can be a decendent class of this.

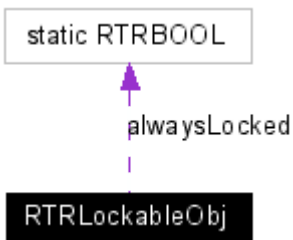
```
#include <lock.h>
```

Inherited by [RTRManagedObject](#), [RTRManagedObjectDirectory](#), [RTRManagedVariable](#), [RTRMasterDictionary](#), [RTRMODescrDictionary](#), [RTRMOServerMemPool](#), [RTRProxyManagedObject](#), [RTRProxyManagedObjectClassDirectory](#), [RTRProxyManagedObjectServer](#), [RTRProxyManagedObjectServerPool](#), and [RTRProxyManagedVariable](#).

Inheritance diagram for RTRLockableObj:



Collaboration diagram for RTRLockableObj:



Public Member Functions

- [RTRLockableObj](#) (RTRBOOL useMutex=RTRTRUE)
- virtual [~RTRLockableObj](#) ()
- virtual void **lock** ()

- virtual void **unlock** ()
- virtual RTRBOOL [locked](#) () const

Static Public Attributes

- static RTRBOOL [alwaysLocked](#)

Detailed Description

RTRLockableObj is a base class representing application component which provides lock/unlock operations on itself perceiving that its states/values could be accessed from multiple threads in applications thus need to be synchronized. Any component that want to be made MT-safe can be a decendent class of this.

The public methods lock()/unlock()/locked() are virtual, so, descendent classes can choose their locking implementation if they decide not to use RTRReentMutex which comes as default. For example, they can use RTRMutex, or share lock with other contained/referenced class.

The state query member function [locked\(\)](#) is used for RTPRECONDITION in application libraries to ensure internal integrity and correct external usage. It can be turn on by set static member alwaysLocked to be false

Dummy implementaions are provided for this class so that for platforms that thread programming is not supported, such as sunOS, or for applications that don't require the library classes to be MT-safe, the MT-safed libraries will have minimal performance penalty.

See Also:

RTRReentMutex, [RTRLock](#)

Constructor & Destructor Documentation

RTRLockableObj::RTRLockableObj (RTRBOOL *useMutex* = RTRTRUE)
Constructor

virtual RTRLockableObj::~~RTRLockableObj () [virtual]
Destructor

Member Data Documentation

RTRBOOL [RTRLockableObj::alwaysLocked](#) [static]
Used to turn on all internal PRECONDITIONs on "object being locked". This could be too strict for thread application where only necessary objects need to be locked in a specific context, thus the default is set to be RTRTRUE here

RTRManagedBoolean Class Reference

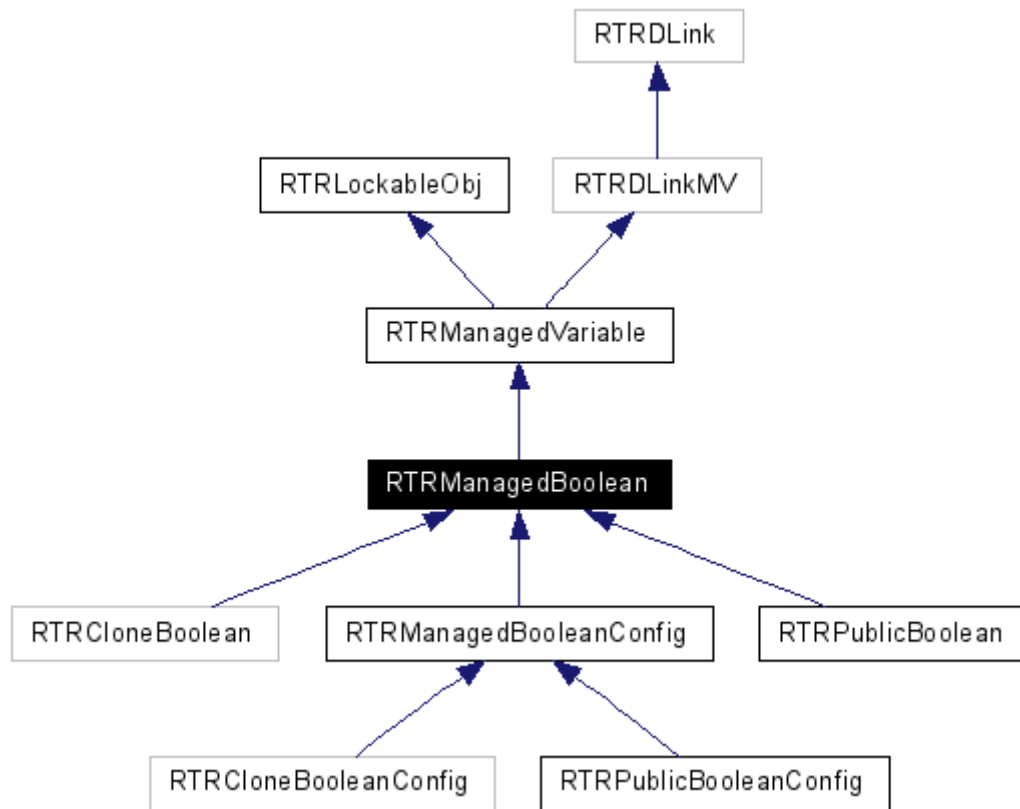
The base class for boolean managed variables. Inherits from [RTRManagedVariable](#) and provides services for accessing and modifying a variable of type boolean.

```
#include <mbvar.h>
```

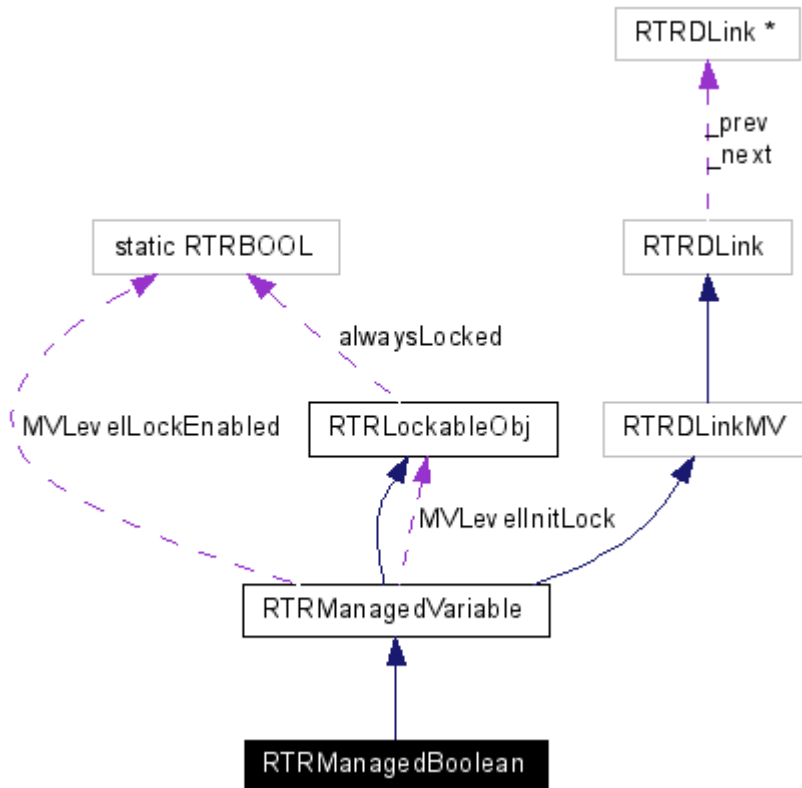
Inherits [RTRManagedVariable](#).

Inherited by RTRCloneBoolean, [RTRManagedBooleanConfig](#), and [RTRPublicBoolean](#).

Inheritance diagram for RTRManagedBoolean:



Collaboration diagram for RTRManagedBoolean:



Public Member Functions

- virtual [~RTRManagedBoolean](#) ()
- RTRBOOL [operator==](#) (RTRBOOL) const
- RTRBOOL [value](#) () const
- virtual [RTRString toString](#) () const
- RTRBOOL [modifyEnabled](#) ()
- [RTRManagedBoolean](#) & [operator=](#) (RTRBOOL rhs)
- virtual void [set](#) ()
- virtual void [clear](#) ()

Friends

- std::ostream & [operator<<](#) (std::ostream &, const [RTRManagedBoolean](#) &)

Detailed Description

The base class for boolean managed variables. Inherits from [RTRManagedVariable](#) and provides services for accessing and modifying a variable of type boolean.

Consumers can modify the variable if permitted by the publisher (`modifyEnabled == RTRTRUE`). Permission is granted/denied when the variable is created and cannot be changed during its life-cycle. The context (ManagedObject) will be notified of changes.

This class cannot be directly instantiated.

See Also:

[RTRManagedObject](#), [RTRManagedNumeric](#), [RTRManagedString](#), [RTRManagedCounter](#), [RTRManagedGauge](#), [RTRManagedNumericRange](#), [RTRManagedStringConfig](#), [RTRManagedNumericConfig](#), [RTRManagedBooleanConfig](#), [RTRManagedGaugeConfig](#)

Constructor & Destructor Documentation

virtual RTRManagedBoolean::~~RTRManagedBoolean () [virtual]
Destructor

RTRManagedBooleanConfig Class Reference

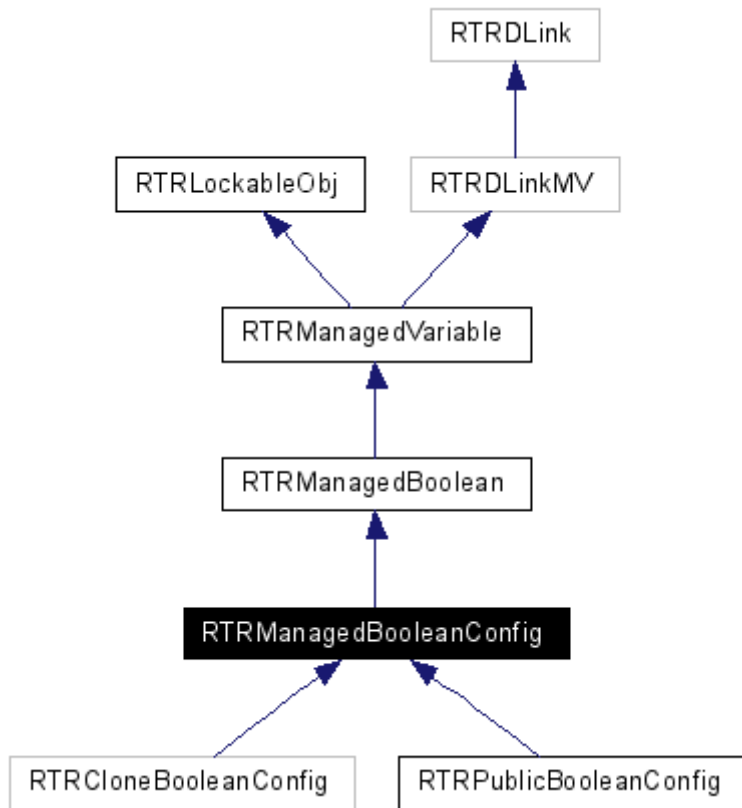
The base class for boolean config managed variables. Inherits from [RTRManagedBoolean](#) and provides services for specifying configuration and default values for the boolean variable.

```
#include <mbcvar.h>
```

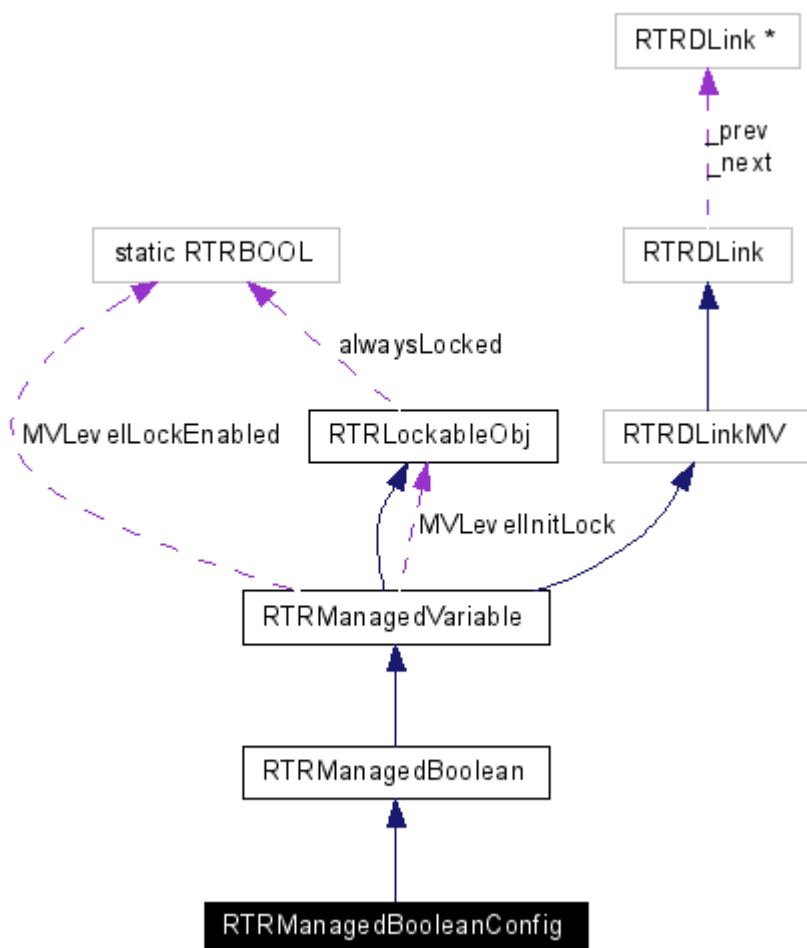
Inherits [RTRManagedBoolean](#).

Inherited by [RTRCloneBooleanConfig](#), and [RTRPublicBooleanConfig](#).

Inheritance diagram for RTRManagedBooleanConfig:



Collaboration diagram for RTRManagedBooleanConfig:



Public Member Functions

- virtual [~RTRManagedBooleanConfig](#) ()
- [RTRManagedBooleanConfig](#) & [operator=](#) (RTRBOOL rhs)
- RTRBOOL [activeValue](#) () const
- RTRBOOL [storeValue](#) () const
- RTRBOOL [factoryDefault](#) () const

Friends

- std::ostream & [operator<<](#) (std::ostream &, const [RTRManagedBooleanConfig](#) &)

Detailed Description

The base class for boolean config managed variables. Inherits from [RTRManagedBoolean](#) and provides services for specifying configuration and default values for the boolean variable.

Consumers can modify the variable if permitted by the publisher (`modifyEnabled == RTRTRUE`). Permission is granted/denied when the variable is created and cannot be changed during its life-cycle. The context (Managed Object) is notified of changes.

This class cannot be directly instantiated.

See Also:

[RTRManagedObject](#), [RTRManagedVariable](#), [RTRManagedNumeric](#), [RTRManagedString](#), [RTRManagedCounter](#), [RTRManagedGauge](#), [RTRManagedBoolean](#), [RTRManagedNumericRange](#), [RTRManagedStringConfig](#), [RTRManagedNumericConfig](#), [RTRManagedGaugeConfig](#)

Constructor & Destructor Documentation

virtual RTRManagedBooleanConfig::~~RTRManagedBooleanConfig () [virtual]
Destructor

RTRManagedCounter Class Reference

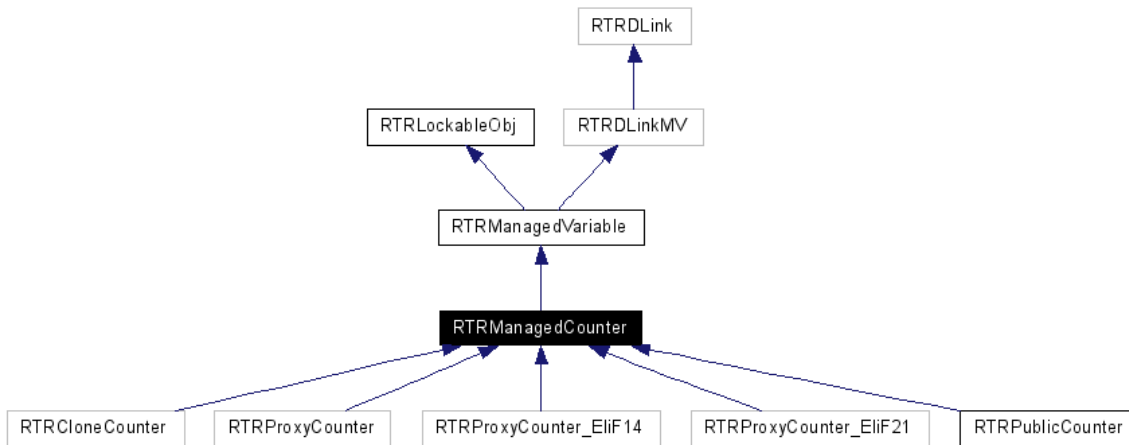
RTRManagedCounter is a descendant of [RTRManagedVariable](#). The RTRManagedCounter can be incremented or reset (to 0); it cannot be decremented.

```
#include <mcntrvar.h>
```

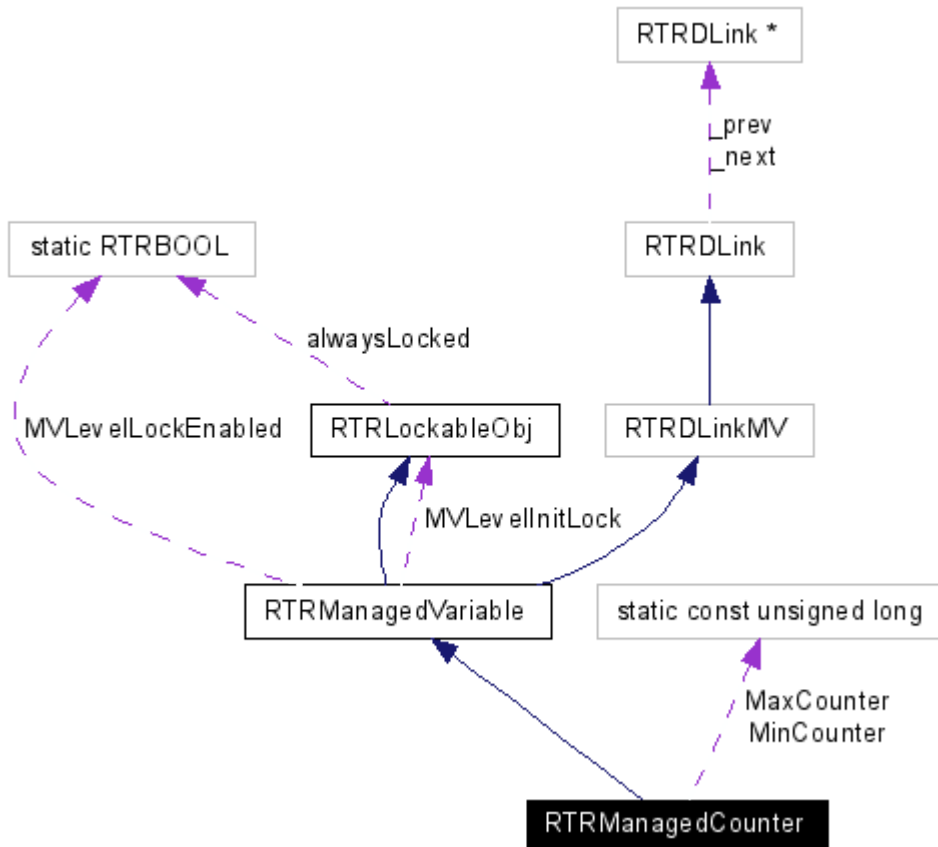
Inherits [RTRManagedVariable](#).

Inherited by RTRCloneCounter, RTRProxyCounter, RTRProxyCounter_EliF14, RTRProxyCounter_EliF21, and [RTRPublicCounter](#).

Inheritance diagram for RTRManagedCounter:



Collaboration diagram for RTRManagedCounter:



Public Member Functions

- virtual [~RTRManagedCounter](#) ()
- RTRBOOL [operator==](#) (unsigned long) const
- unsigned long [value](#) () const
- [operator unsigned long](#) () const
- virtual [RTRString toString](#) () const
- virtual void [reset](#) ()=0

Static Public Attributes

- static const unsigned long **MinCounter**
- static const unsigned long **MaxCounter**

Friends

- std::ostream & [operator<<](#) (std::ostream &, const [RTRManagedCounter](#) &)

Detailed Description

RTRManagedCounter is a descendant of [RTRManagedVariable](#). The RTRManagedCounter can be incremented or reset (to 0), it cannot be decremented.

Consumers are permitted to reset (to 0) the variable. The context (ManagedObject) is not notified of resets.

This class cannot be directly instantiated.

See Also:

[RTRManagedObject](#), [RTRManagedVariable](#), [RTRManagedString](#), [RTRManagedNumeric](#), [RTRManagedGauge](#), [RTRManagedBoolean](#), [RTRManagedNumericRange](#), [RTRManagedStringConfig](#), [RTRManagedNumericConfig](#), [RTRManagedGaugeConfig](#), [RTRManagedBooleanConfig](#).

Constructor & Destructor Documentation

virtual RTRManagedCounter::~~RTRManagedCounter () [virtual]
Destructor

RTRManagedGauge Class Reference

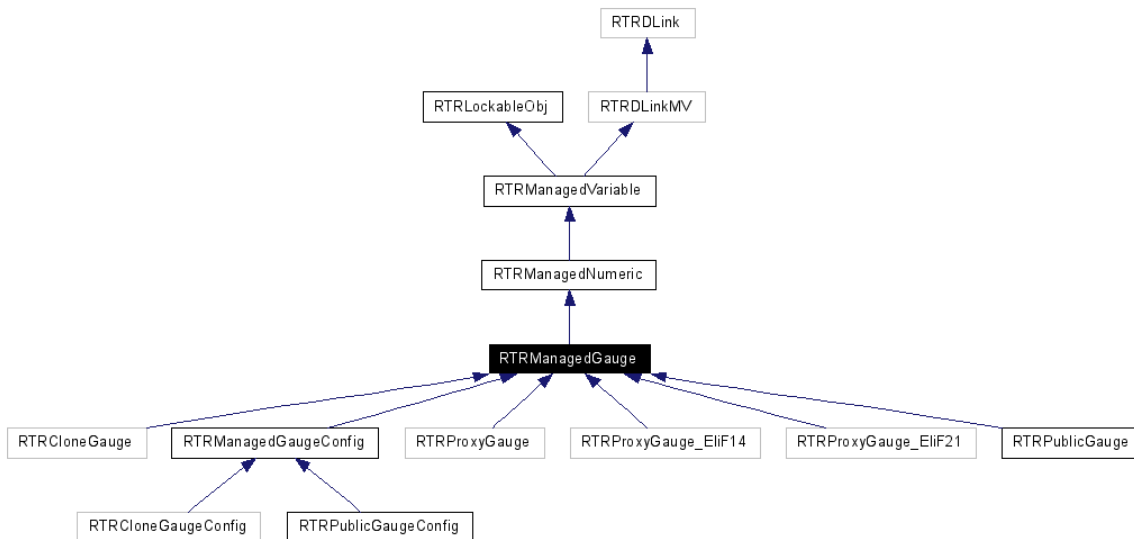
A RTRManagedGauge is a descendant of [RTRManagedNumeric](#) and provides services for min/max values and low/high water marks. The low/high water marks indicate the lowest/highest values assumed by a gauge since its creation.

```
#include <mgvar.h>
```

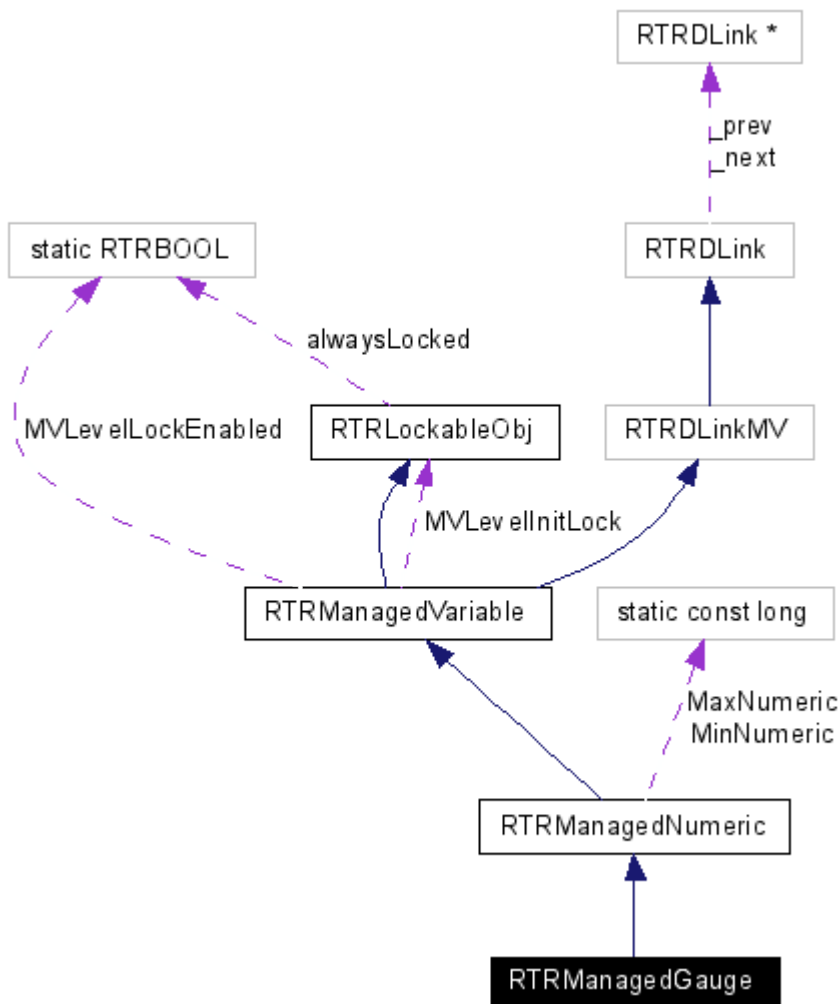
Inherits [RTRManagedNumeric](#).

Inherited by RTRCloneGauge, [RTRManagedGaugeConfig](#), RTRProxyGauge, RTRProxyGauge_EliF14, RTRProxyGauge_EliF21, and [RTRPublicGauge](#).

Inheritance diagram for RTRManagedGauge:



Collaboration diagram for RTRManagedGauge:



Public Member Functions

- virtual [~RTRManagedGauge](#) ()
- long [minValue](#) () const
- long [maxValue](#) () const
- long [lowWaterMark](#) () const
- long [highWaterMark](#) () const
- RTRBOOL [modifyEnabled](#) () const
- virtual void [setRange](#) (long newMin, long newMax)

Friends

- std::ostream & **operator<<** (std::ostream &, const [RTRManagedGauge](#) &)

Detailed Description

A [RTRManagedGauge](#) is a descendant of [RTRManagedNumeric](#) and provides services for min/max values and low/high water marks. The low/high water marks indicate the lowest/highest values assumed by a gauge since its creation.

Consumers can modify the min and max attributes of the variable if permitted by the publisher (`modifyEnabled == RTRTRUE`). Permission is granted/denied when the variable is created and cannot be changed during its life-cycle. The context (`ManagedObject`) is notified of changes.

if `modifyEnabled` is false, then the active value will be between [minValue\(\)](#) and [maxValue\(\)](#) (ie. `minValue <= activeValue <= maxValue`). This restriction does not exist if the consumer is permitted to modify the min/max values. The active value could be outside the min/max range.

This class cannot be directly instantiated.

See Also:

[RTRManagedObject](#), [RTRManagedVariable](#), [RTRManagedString](#), [RTRManagedNumeric](#), [RTRManagedCounter](#), [RTRManagedBoolean](#), [RTRManagedNumericRange](#), [RTRManagedStringConfig](#), [RTRManagedNumericConfig](#), [RTRManagedGaugeConfig](#), [RTRManagedBooleanConfig](#).

Constructor & Destructor Documentation

```
virtual RTRManagedGauge::~RTRManagedGauge () [virtual]
    Destructor
```

RTRManagedGaugeConfig Class Reference

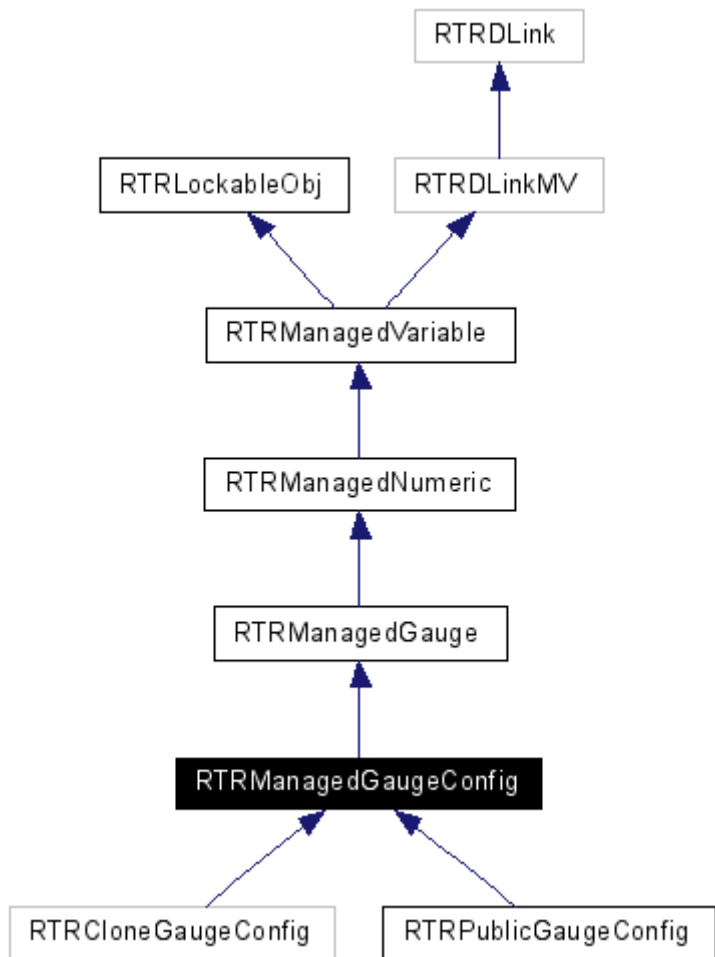
Inherits from [RTRManagedGauge](#) and provides services for providing a configuration and default min/max values for the gauge.

```
#include <mgcvar.h>
```

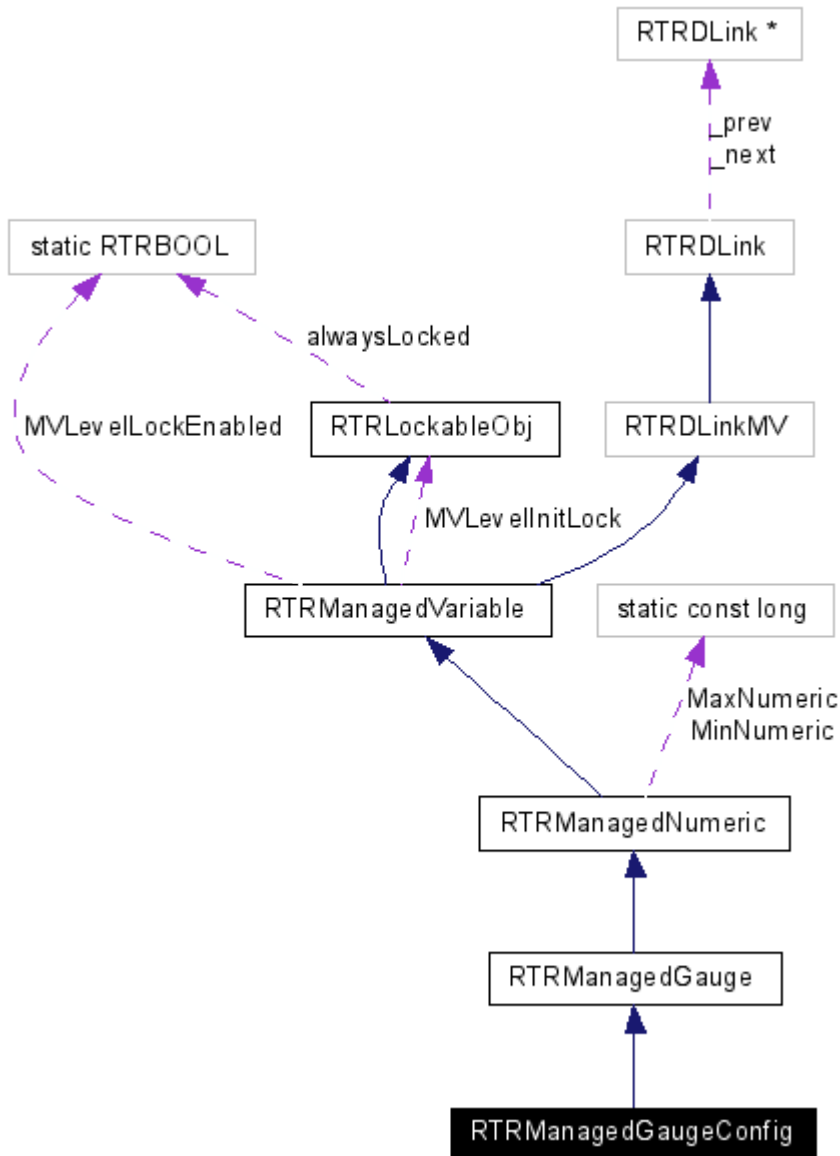
Inherits [RTRManagedGauge](#).

Inherited by [RTRCloneGaugeConfig](#), and [RTRPublicGaugeConfig](#).

Inheritance diagram for [RTRManagedGaugeConfig](#):



Collaboration diagram for RTRManagedGaugeConfig:



Public Member Functions

- virtual [~RTRManagedGaugeConfig](#) ()
- long [minStoreValue](#) () const
- long [minFactoryDefault](#) () const
- long [maxStoreValue](#) () const
- long [maxFactoryDefault](#) () const

Friends

- std::ostream & **operator**<< (std::ostream &, const [RTRManagedGaugeConfig](#) &)

Detailed Description

Inherits from [RTRManagedGauge](#) and provides services for providing a configuration and default min/max values for the gauge.

Consumers can modify the min and max attributes of the variable if permitted by the publisher (modifyEnabled == RTRTRUE). Permission is granted/denied when the variable is created and cannot be changed during its life-cycle. The context (ManagedObject) is notified of changes.

if modifyEnabled is false, then the active value will be between [minValue\(\)](#) and [maxValue\(\)](#) (ie. $\text{minValue} \leq \text{activeValue} \leq \text{maxValue}$). This restriction does not exist if the consumer is permitted to modify the min/max values. The active value could be outside the min/max range.

This class cannot be directly instantiated.

See Also:

[RTRManagedObject](#), [RTRManagedVariable](#), [RTRManagedString](#), [RTRManagedNumeric](#), [RTRManagedCounter](#), [RTRManagedGauge](#), [RTRManagedBoolean](#), [RTRManagedNumericRange](#), [RTRManagedStringConfig](#), [RTRManagedNumericConfig](#), [RTRManagedBooleanConfig](#)

Constructor & Destructor Documentation

`virtual RTRManagedGaugeConfig::~~RTRManagedGaugeConfig () [virtual]`

Destructor

RTRManagedLargeNumeric Class Reference

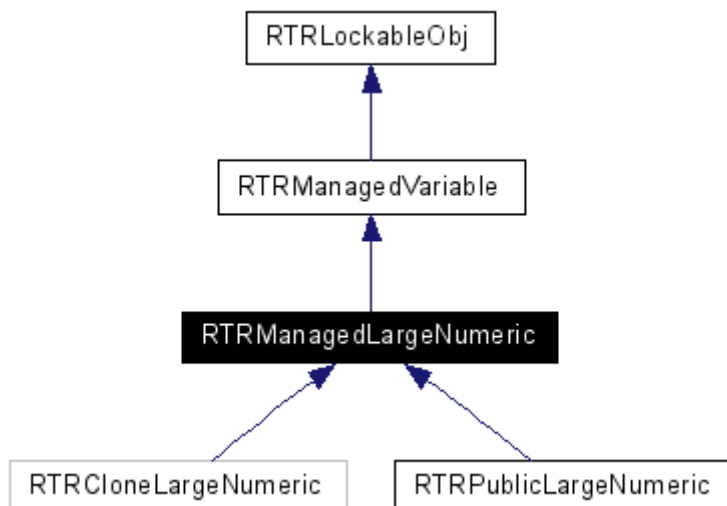
The base class for large numeric managed variables. This class provides a read-only interface to the large numeric value.

```
#include <m1numvar.h>
```

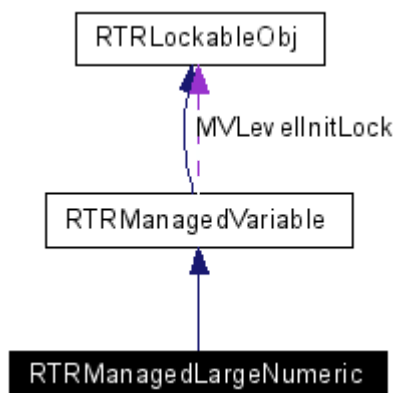
Inherits [RTRManagedVariable](#).

Inherited by [RTRCloneLargeNumeric](#), and [RTRPublicLargeNumeric](#).

Inheritance diagram for RTRManagedLargeNumeric:



Collaboration diagram for RTRManagedLargeNumeric:



Public Member Functions

- virtual [~RTRManagedLargeNumeric](#) ()
- RTRBOOL [operator==](#) (RTR_I64) const
- RTR_I64 [value](#) () const
- [operator RTR_I64](#) () const
- virtual [RTRString toString](#) () const

Static Public Attributes

- static const RTR_I64 **LongMinNumeric**
- static const RTR_I64 **LongMaxNumeric**

Friends

- std::ostream & [operator<<](#) (std::ostream &, const [RTRManagedLargeNumeric](#) &)

Detailed Description

The base class for large numeric managed variables. This class provides a read-only interface to the large numeric value. This class cannot be directly instantiated.

See Also:

[RTRManagedObject](#), [RTRManagedString](#), [RTRManagedBoolean](#), [RTRManagedCounter](#), [RTRManagedGauge](#), [RTRManagedNumeric](#), [RTRManagedNumericRange](#), [RTRManagedStringConfig](#), [RTRManagedNumericConfig](#), [RTRManagedBooleanConfig](#), [RTRManagedGaugeConfig](#)

Constructor & Destructor Documentation

virtual RTRManagedLargeNumeric::~~RTRManagedLargeNumeric () [virtual]
Destructor

RTRManagedNumeric Class Reference

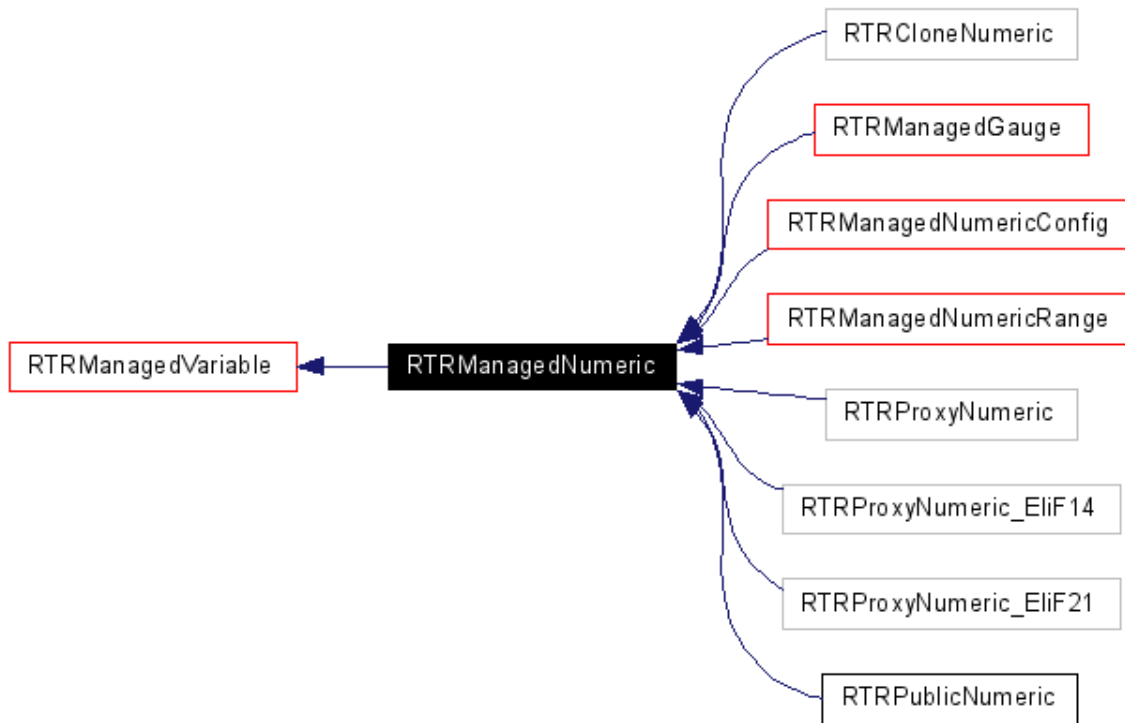
The base class for numeric managed variables. This class provides a read-only interface to the numeric value.

```
#include <mnumvar.h>
```

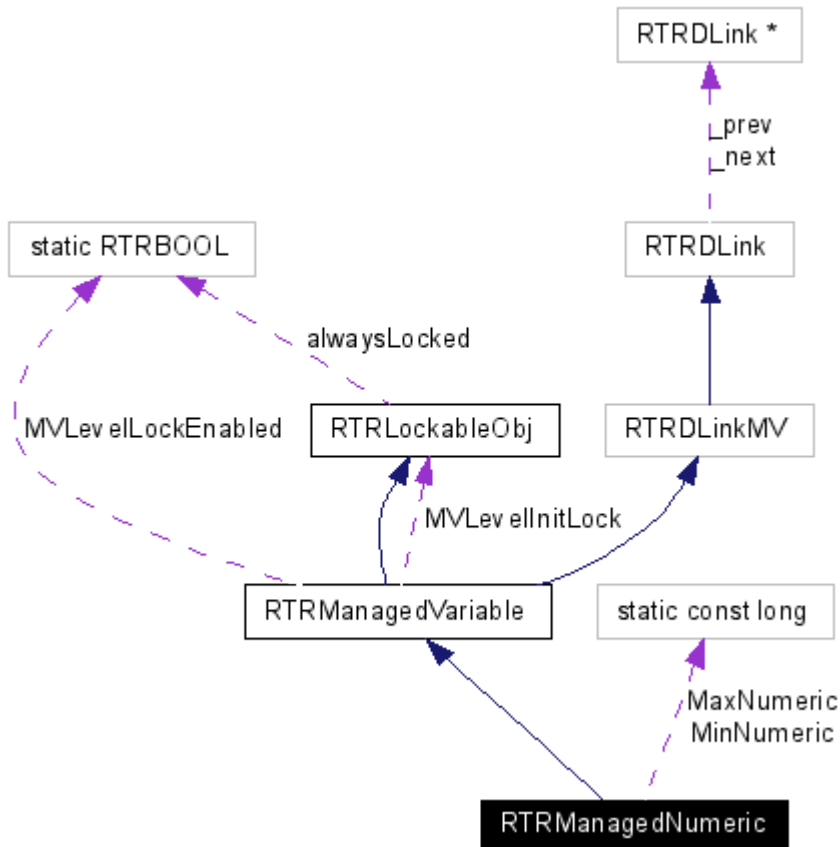
Inherits [RTRManagedVariable](#).

Inherited by [RTRCloneNumeric](#), [RTRManagedGauge](#), [RTRManagedNumericConfig](#), [RTRManagedNumericRange](#), [RTRProxyNumeric](#), [RTRProxyNumeric_EliF14](#), [RTRProxyNumeric_EliF21](#), and [RTRPublicNumeric](#).

Inheritance diagram for RTRManagedNumeric:



Collaboration diagram for RTRManagedNumeric:



Public Member Functions

- virtual [~RTRManagedNumeric](#) ()
- RTRBOOL [operator==](#) (long) const
- long [value](#) () const
- [operator long](#) () const
- virtual [RTRString toString](#) () const

Static Public Attributes

- static const long **MinNumeric**
- static const long **MaxNumeric**

Friends

- std::ostream & **operator<<** (std::ostream &, const [RTRManagedNumeric](#) &)

Detailed Description

The base class for numeric managed variables. This class provides a read-only interface to the numeric value.

This class cannot be directly instantiated.

See Also:

[RTRManagedObject](#), [RTRManagedString](#), [RTRManagedBoolean](#), [RTRManagedCounter](#), [RTRManagedGauge](#), [RTRManagedNumericRange](#), [RTRManagedStringConfig](#), [RTRManagedNumericConfig](#), [RTRManagedBooleanConfig](#), [RTRManagedGaugeConfig](#)

Constructor & Destructor Documentation

virtual RTRManagedNumeric::~~RTRManagedNumeric () [virtual]

Destructor

RTRManagedNumericConfig Class Reference

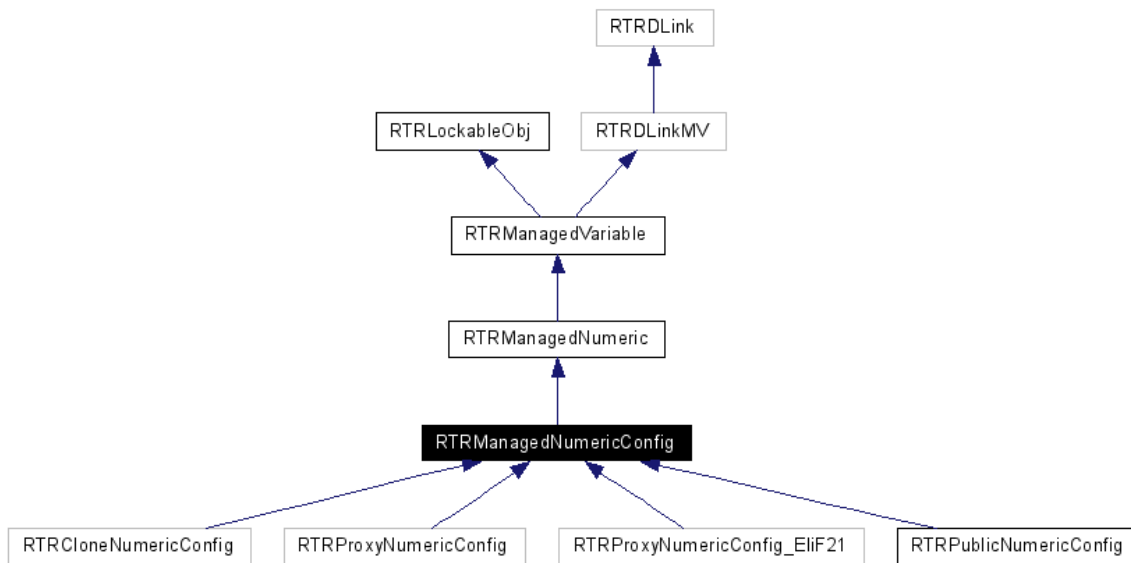
Numeric configs inherit from Numeric and provides services for specifying min/max values, configuration value and a default value.

```
#include <mnumcvar.h>
```

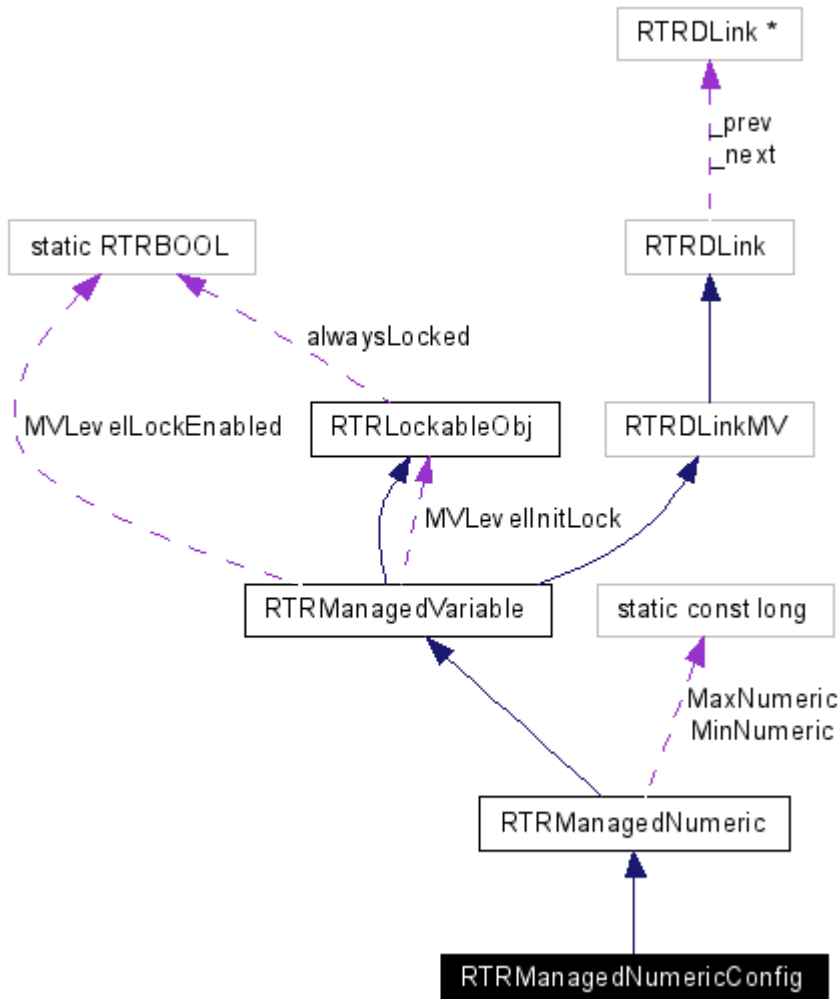
Inherits [RTRManagedNumeric](#).

Inherited by [RTRCloneNumericConfig](#), [RTRProxyNumericConfig](#), [RTRProxyNumericConfig_EliF21](#), and [RTRPublicNumericConfig](#).

Inheritance diagram for RTRManagedNumericConfig:



Collaboration diagram for RTRManagedNumericConfig:



Public Member Functions

- virtual [~RTRManagedNumericConfig](#) ()
- long [minValue](#) () const
- long [maxValue](#) () const
- long [activeValue](#) () const
- long [storeValue](#) () const
- int [storeState](#) () const
- long [factoryDefault](#) () const
- RTRBOOL [modifyEnabled](#) () const
- RTRBOOL [hasStore](#) () const
- RTRBOOL [isStoreActive](#) () const
- RTRBOOL [isStoreClassConfig](#) () const

- RTRBOOL [isStoreInstanceConfig](#) () const
- [RTRManagedNumericConfig](#) & [operator=](#) (long rhs)
- virtual void [set](#) (long newValue)

Friends

- std::ostream & [operator<<](#) (std::ostream &, const [RTRManagedNumericConfig](#) &)

Detailed Description

Numeric configs inherit from Numeric and provides services for specifying min/max values, configuration value and a default value.

Consumers can modify the variable if permitted by the publisher (modifyEnabled == RTRTRUE). Permission is granted/denied when the variable is created and cannot be changed during its life-cycle. The context (ManagedObject) will be notified of the change.

The value must always be within the min/max range. (ie. [minValue\(\)](#) <= [value\(\)](#) <= [maxValue\(\)](#))

This class cannot be instantiated directly.

See Also:

[RTRManagedObject](#), [RTRManagedVariable](#), [RTRManagedString](#), [RTRManagedNumeric](#), [RTRManagedCounter](#), [RTRManagedGauge](#), [RTRManagedBoolean](#), [RTRManagedNumericRange](#), [RTRManagedStringConfig](#), [RTRManagedBooleanConfig](#), [RTRManagedGaugeConfig](#)

Constructor & Destructor Documentation

virtual RTRManagedNumericConfig::~RTRManagedNumericConfig () [virtual]
Destructor

RTRManagedNumericRange Class Reference

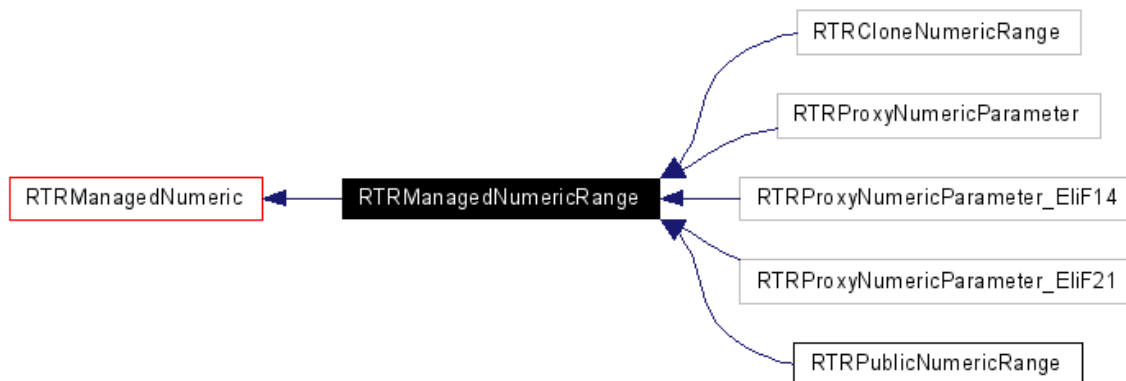
Numeric Ranges inherit from Numeric and provides services for specifying a min/max value.

```
#include <mnumrvar.h>
```

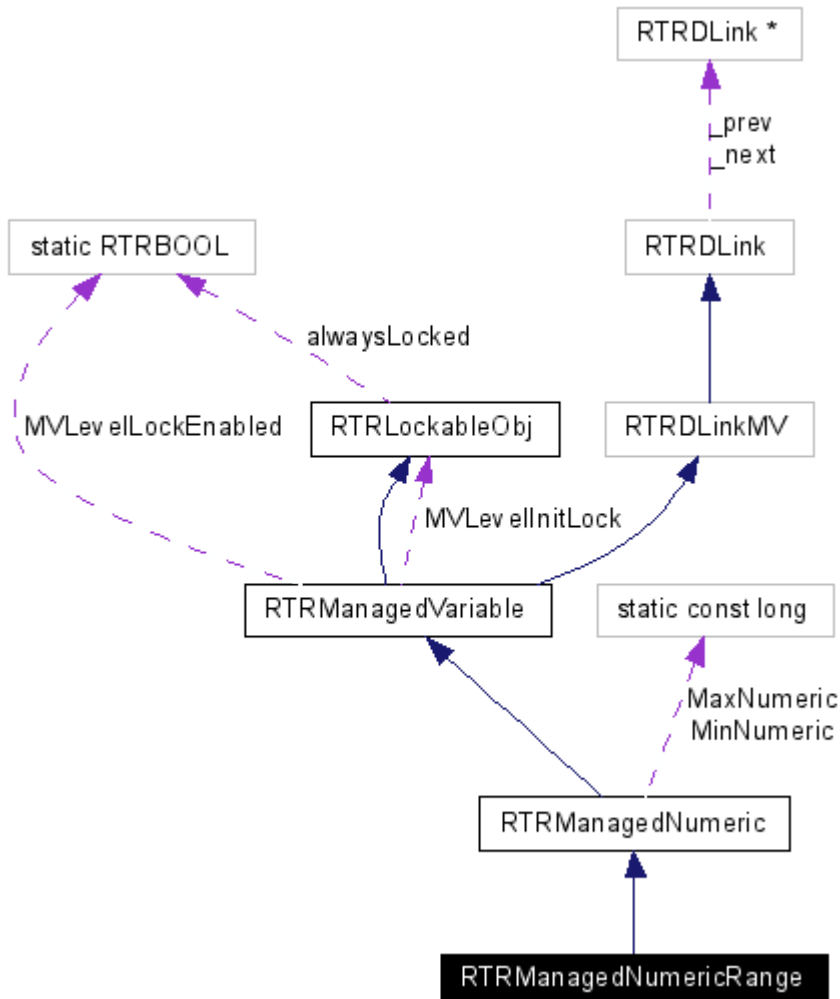
Inherits [RTRManagedNumeric](#).

Inherited by [RTRCloneNumericRange](#), [RTRProxyNumericParameter](#), [RTRProxyNumericParameter_EliF14](#), [RTRProxyNumericParameter_EliF21](#), and [RTRPublicNumericRange](#).

Inheritance diagram for RTRManagedNumericRange:



Collaboration diagram for RTRManagedNumericRange:



Public Member Functions

- virtual [~RTRManagedNumericRange](#) ()
- **Destructor.**
- long [minValue](#) () const
- long [maxValue](#) () const
- [RTRManagedNumericRange](#) & [operator=](#) (long rhs)
- virtual void [set](#) (long newValue)

Friends

- std::ostream & **operator<<** (std::ostream &, const [RTRManagedNumericRange](#) &)

Detailed Description

Numeric Ranges inherit from Numeric and provides services for specifying a min/max value.

Consumers are always permitted to modify the value. The value must always be within the min/max range. (ie. [minValue\(\)](#) <= [value\(\)](#) <= [maxValue\(\)](#)) The context (ManagedObject) will be notified of the change.

The range may be changed by the publisher but the value will always be within the min/max range. (i.e. [minValue\(\)](#) <= [value\(\)](#) <= [maxValue\(\)](#))

This class cannot be instantiated directly.

See Also:

[RTRManagedObject](#), [RTRManagedVariable](#), [RTRManagedString](#), [RTRManagedNumeric](#), [RTRManagedCounter](#), [RTRManagedGauge](#), [RTRManagedBoolean](#), [RTRManagedStringConfig](#), [RTRManagedNumericConfig](#), [RTRManagedGaugeConfig](#), [RTRManagedBooleanConfig](#).

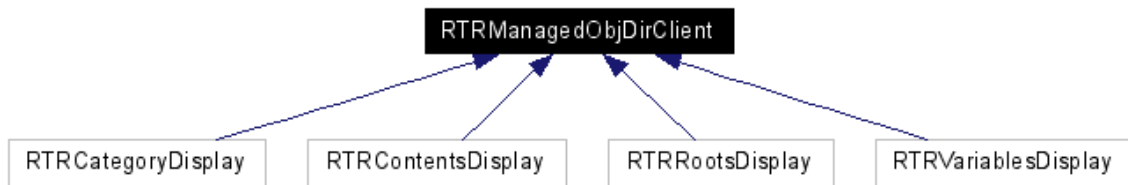
RTRManagedObjDirClient Class Reference

RTRManagedObjDirClient is the base class for application components which wish to register to receive events from an instance of [RTRManagedObjectDirectory](#).

```
#include <mod.h>
```

Inherited by RTRCategoryDisplay, RTRContentsDisplay, RTRRootsDisplay, and RTRVariablesDisplay.

Inheritance diagram for RTRManagedObjDirClient:



Public Member Functions

- virtual void [processCategoryAdded](#) ([RTRManagedObjectDirectory](#) &, RTRClassCategory< [RTRManagedObject](#) > &)
- virtual void [processManagedObjectAdded](#) ([RTRManagedObjectDirectory](#) &, [RTRManagedObject](#) &)
- virtual void [processManagedObjectRemoved](#) ([RTRManagedObjectDirectory](#) &, [RTRManagedObject](#) &)

Detailed Description

RTRManagedObjDirClient is the base class for application components which wish to register to receive events from an instance of [RTRManagedObjectDirectory](#).

See Also:

[RTRManagedObjectDirectory](#), RTRClassCategory<RTRManagedObject>, [RTRManagedObject](#)

Member Function Documentation

virtual void RTRManagedObjDirClient::processCategoryAdded ([RTRManagedObjectDirectory](#) &, RTRClassCategory< [RTRManagedObject](#) > &) [virtual]

Event processing

The given object has been added to the global directory.

virtual void RTRManagedObjDirClient::processManagedObjectAdded ([RTRManagedObjectDirectory](#) &, [RTRManagedObject](#) &) [virtual]

Event processing

The given object has been added to the global directory.

virtual void RTRManagedObjDirClient::processManagedObjectRemoved ([RTRManagedObjectDirectory](#) &, [RTRManagedObject](#) &) [virtual]

Event processing

The given object has been removed from the global directory.

RTRManagedObjDirRootIterator Class Reference

An iterator for directory root object.

```
#include <mod.h>
```

Public Member Functions

- [RTRManagedObjDirRootIterator](#) ([RTRManagedObjectDirectory](#) *)
- [~RTRManagedObjDirRootIterator](#) ()
- int [count](#) () const
- RTRBOOL [off](#) () const
- RTRBOOL [empty](#) () const
- [RTRManagedObject](#) & [item](#) () const
- void [start](#) ()
- void [finish](#) ()
- void [forth](#) ()
- void [back](#) ()

Detailed Description

An iterator for directory root object.

See Also:

[RTRManagedObjectDirectory](#), [RTRManagedObjDirClient](#)

Constructor & Destructor Documentation

RTRManagedObjDirRootIterator::RTRManagedObjDirRootIterator ([RTRManagedObjectDirectory](#) *)
Constructor

RTRManagedObjDirRootIterator::~~RTRManagedObjDirRootIterator ()
Destructor

Member Function Documentation

int RTRManagedObjDirRootIterator::count () const

Attributes

The number of roots available via this iterator.

RTRBOOL RTRManagedObjDirRootIterator::off () const

State

Is this iteration complete?

RTRBOOL RTRManagedObjDirRootIterator::empty () const

State

Are there no root available via this iterator?

ENSURE: result implies [count\(\)](#) == 0

[RTRManagedObject](#)& RTRManagedObjDirRootIterator::item () const

Access

The current item in the current iteration.

void RTRManagedObjDirRootIterator::start ()

Operation

Start a new iteration.

ENSURE:

[off\(\)](#) implies [empty\(\)](#)

void RTRManagedObjDirRootIterator::finish ()

Operation

Start an iteration from the last available root.

ENSURE: [off\(\)](#) implies [empty\(\)](#)

void RTRManagedObjDirRootIterator::forth ()

Operation

Continue the current iteration from [start\(\)](#) to [finish\(\)](#).

REQUIRE: ![off\(\)](#)

void RTRManagedObjDirRootIterator::back ()

Operation

Continue the current iteration from [finish\(\)](#) to [start\(\)](#).

REQUIRE: ![off\(\)](#)

RTRManagedObject Class Reference

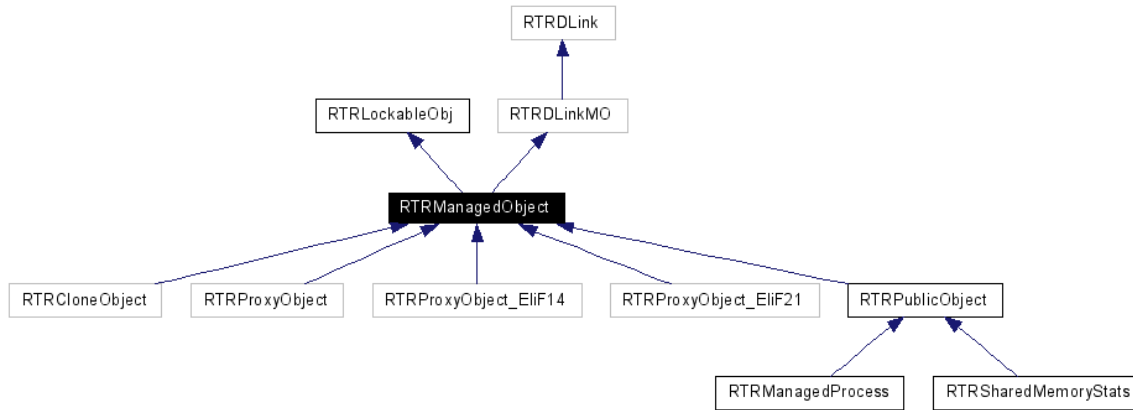
RTRManagedObject is an abstract base class representing application components which can be accessed and managed by external management entities. Management is effected by monitoring and possibly modifying variables made available by the application component to be managed.

```
#include <mo.h>
```

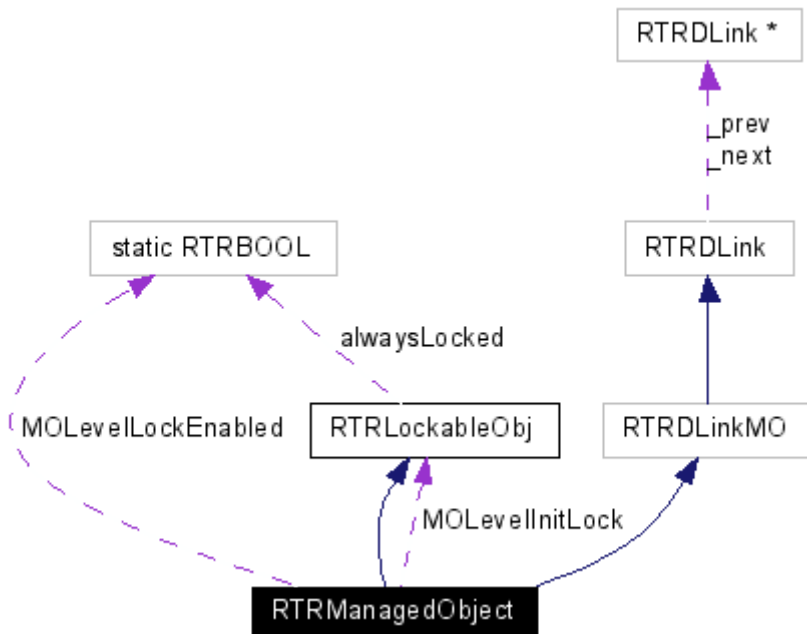
Inherits [RTRLockableObj](#), and RTRDLinkMO.

Inherited by RTRCloneObject, RTRProxyObject, RTRProxyObject_EliF14, RTRProxyObject_EliF21, and [RTRPublicObject](#).

Inheritance diagram for RTRManagedObject:



Collaboration diagram for RTRManagedObject:



Public Types

- enum [MOState](#) { Init, Normal, ManualRecovery, AutoRecovery, Dead }

Public Member Functions

- virtual [~RTRManagedObject](#) ()
- const [RTRObjectId](#) & [classId](#) () const
- const [RTRObjectId](#) & [instanceId](#) () const
- const [RTRString](#) & [name](#) () const
- [RTRManagedObject](#) * [parent](#) () const
- [MOState](#) [state](#) () const

- [MOState previousState](#) () const
- const char * [text](#) () const
- const [RTRString](#) & [description](#) () const
- RTRBOOL [isInitializing](#) () const
- RTRBOOL [isNormal](#) () const
- RTRBOOL [isRecovering](#) () const
- RTRBOOL [isWaiting](#) () const
- RTRBOOL [isInterrupted](#) () const
- RTRBOOL [isDead](#) () const
- RTRBOOL [hasChild](#) (const [RTRString](#) &) const
- RTRBOOL [hasVariable](#) (const [RTRString](#) &) const
- [RTRManagedObjectIterator childIterator](#) () const
- [RTRManagedVariableIterator variableIterator](#) () const
- [RTRManagedObject](#) * [childByName](#) (const char *) const
- [RTRManagedVariable](#) * [variableByName](#) (const char *) const
- [RTRManagedBoolean](#) * [booleanByName](#) (const char *) const
- [RTRManagedBooleanConfig](#) * [booleanConfigByName](#) (const char *) const
- [RTRManagedCounter](#) * [counterByName](#) (const char *) const
- [RTRManagedGauge](#) * [gaugeByName](#) (const char *) const
- [RTRManagedGaugeConfig](#) * [gaugeConfigByName](#) (const char *) const
- [RTRManagedNumeric](#) * [numericByName](#) (const char *) const
- [RTRManagedNumericConfig](#) * [numericConfigByName](#) (const char *) const
- [RTRManagedNumericRange](#) * [numericRangeByName](#) (const char *) const
- [RTRManagedString](#) * [stringByName](#) (const char *) const
- [RTRManagedStringConfig](#) * [stringConfigByName](#) (const char *) const
- void [addClient](#) ([RTRManagedObjectClient](#) &client)
- void [dropClient](#) ([RTRManagedObjectClient](#) &client)
- RTRBOOL [hasClient](#) ([RTRManagedObjectClient](#) &client) const
- virtual void [lock](#) ()
- virtual void [unlock](#) ()
- virtual RTRBOOL [locked](#) () const
- RTRMOImpl * [storeImpl](#) () const
- RTRMOImplPub * [storeImplPub](#) () const
- void [cleanUpImplPub](#) ()

- RTRBOOL [initImplPub](#) (RTRManagedMemAllocator &, RTRBOOL=RTRTRUE)
- virtual void [processParameterChange](#) ([RTRManagedVariable](#) &)
- virtual void [processConfigChange](#) ([RTRManagedVariable](#) &)

Static Public Attributes

- static RTRBOOL [MOLevelLockEnabled](#)
- static [RTRLockableObj](#) [MOLevelInitLock](#)

Friends

- class RTRMOImpIPub
- class RTRManagedMemAllocator
- class [RTRManagedVariable](#)
- class [RTRManagedVariableIterator](#)
- class [RTRManagedObjectIterator](#)
- std::ostream & **operator<<** (std::ostream &, const [RTRManagedObject](#) &)

Detailed Description

RTRManagedObject is an abstract base class representing application components which can be accessed and managed by external management entities. Management is effected by monitoring and possibly modifying variables made available by the application component to be managed.

Managed applications are perceived by management entities as a collection of managed objects. These objects provide some number of variables of interest. The exact variables provided by an instance of managed object depends on the type of that object. All variables conform to one of a limited number of types; all variable types are specializations of [RTRManagedVariable](#).

The managed objects within an application have relationships with other managed objects, i.e. objects may refer to other objects. These relationships form one or more trees (whose nodes are objects) and can be of interest to management entities. Managed objects contained by other objects are children. Managed objects not contained by other objects are the roots of object trees. Given the set of roots in an application, or system, all other objects can be reached by traversing the trees defined by those roots.

The instance tree represents the composition of an application as represented to external management entities, and provides the means for those entities to access and possibly modify variables within the application. The application components which comprise the instance tree are not concerned with the nature or implementation of the external components.

Managed objects have an instance identifier which uniquely identifies that object. Managed objects also have a class identifier which identifies the type (semantics) of that object. In principle, all managed objects with a given class identifier (type) will provide the same set of variables.

Application components which wish to be notified of changes in the composition of a particular managed object may register with that object in order to receive object level events. To do so, they must be descendants of [RTRManagedObjectClient](#).

RTRManagedObject is a base class which has implementation specific descendants. Application components should not inherit from this class directly. Managed objects are constructed with a name (which must be unique in the creating context), a class, and, optionally, a parent. The instance identifier of child objects is constructed by combining the instance identifier of the parent (if any) and the name. The object and variable relationships are maintained automatically in the appropriate constructors and destructors.

The current object tree can be accessed via the global managed object directory, an instance of [RTRManagedObjectDirectory](#) available via RTRGlobalManagedObjectDirectory.

See Also:

[RTRManagedObjectDirectory](#), [RTRGlobalManagedObjectDirectory](#), [RTRManagedVariable](#), [RTRManagedString](#), [RTRManagedBoolean](#), [RTRManagedCounter](#), [RTRManagedNumeric](#), [RTRManagedNumericRange](#), [RTRManagedGauge](#), [RTRManagedGaugeConfig](#), [RTRManagedBooleanConfig](#), [RTRPublicObject](#), [RTRManagedObjectClient](#)

Member Enumeration Documentation

enum [RTRManagedObject::MOState](#)

The type of managed variable

Constructor & Destructor Documentation

virtual [RTRManagedObject::~RTRManagedObject](#) () [**virtual**]

Destructor

Member Data Documentation

RTRBOOL [RTRManagedObject::MOLevelLockEnabled](#) [**static**]

Default is false. Enabled when same object could be accessed in multiple threads.

[RTRLockableObj](#) [RTRManagedObject::MOLevelInitLock](#) [**static**]

To synchronize the initialization when MOLevelLockEnabled is true.

RTRManagedObjectClient Class Reference

The abstract base class for components which wish to receive object level events.

```
#include <mo.h>
```

Inherited by [RTRChildrenDisplay](#), [RTRInstanceDisplay](#), and [RTRStateSelection](#).

Inheritance diagram for [RTRManagedObjectClient](#):



Public Member Functions

- virtual void [processObjectDeleted](#) ([RTRManagedObject](#) &)=0
- virtual void [processObjectInService](#) ([RTRManagedObject](#) &)
- virtual void [processObjectRecovering](#) ([RTRManagedObject](#) &)
- virtual void [processObjectWaiting](#) ([RTRManagedObject](#) &)
- virtual void [processObjectDead](#) ([RTRManagedObject](#) &)
- virtual void [processObjectInfo](#) ([RTRManagedObject](#) &)
- virtual void [processChildAdded](#) ([RTRManagedObject](#) &, [RTRManagedObject](#) &ch)
- virtual void [processChildRemoved](#) ([RTRManagedObject](#) &, [RTRManagedObject](#) &ch)
- virtual void [processVariableAdded](#) ([RTRManagedObject](#) &, [RTRManagedVariable](#) &)
- virtual void [processVariableRemoved](#) ([RTRManagedObject](#) &, [RTRManagedVariable](#) &)

Detailed Description

The abstract base class for components which wish to receive object level events.

See Also:

[RTRManagedObjectDirectory](#), [RTRGlobalManagedObjectDirectory](#), [RTRManagedVariable](#), [RTRManagedString](#), [RTRManagedBoolean](#), [RTRManagedCounter](#), [RTRManagedNumeric](#), [RTRManagedNumericRange](#), [RTRManagedGauge](#), [RTRManagedBooleanConfig](#), [RTRManagedGaugeConfig](#), [RTRPublicObject](#)

RTRManagedObjectDirectory Class Reference

A descendant of the template class RTRDirectory which is specific to managed objects, instances of this class provide access to a list of so-called root objects, i.e. managed objects with no parent.

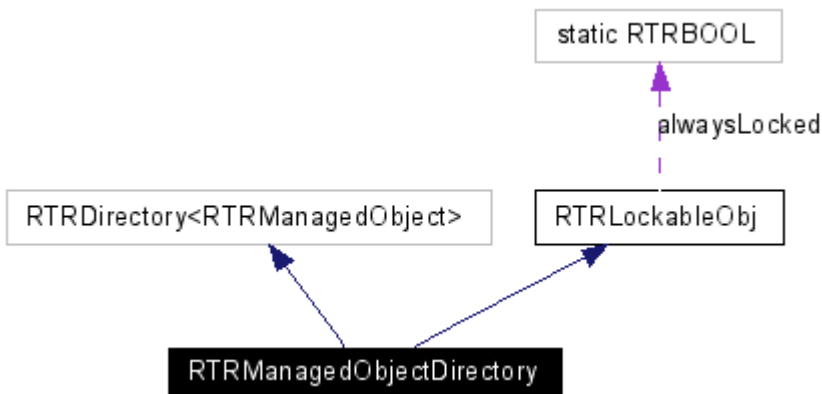
```
#include <mod.h>
```

Inherits RTRDirectory, and [RTRLockableObj](#).

Inheritance diagram for RTRManagedObjectDirectory:



Collaboration diagram for RTRManagedObjectDirectory:



Public Member Functions

- [RTRManagedObjectDirectory](#) ()
- [~RTRManagedObjectDirectory](#) ()
- [RTRManagedObjDirRootIterator](#) [rootIterator](#) ()
- const RTRClassCategory< [RTRManagedObject](#) > & [automaticCategory](#) (const [RTRObjId](#) &cid) const
- virtual void [put](#) ([RTRManagedObject](#) &mo)
- virtual void [remove](#) ([RTRManagedObject](#) &mo)

- void [addClient](#) ([RTRManagedObjDirClient](#) &newClient)
- void [dropClient](#) ([RTRManagedObjDirClient](#) &oldClient)
- RTRBOOL [hasClient](#) ([RTRManagedObjDirClient](#) &client) const

Friends

- class [RTRManagedObjDirRootIterator](#)

Detailed Description

A descendant of the template class RTRDirectory which is specific to managed objects, instances of this class provide access to a list of so-called root objects, i.e. managed objects with no parent.

Typically, an application (process) has only a single static instance of RTRManagedObjectDirectory. This instance is made available to application components via the RTRGlobalManagedObjectDirectory class.

A directory consists of a set of categories. Each category within the directory is a container for all objects ([RTRManagedObject](#)) in the directory whose class (type) conforms to the class represented by that category.

Categories are allocated by the directory as needed in response to the insertion of objects into the directory. As objects are added to the directory, the directory allocates the necessary categories. The number of categories affected depends on the class identifier of the inserted object.

An object may conform to more than one class (polymorphism) and hence may be contained by more than one category within the directory. For example: If the type BaseClass.SubClass is a descendant of BaseClass and object of type BaseClass.SubClass is inserted into the directory, then that object will be found in categories for both BaseClass and BaseClass.SubClass.

A directory provides both random and sequential access to its constituent categories. It also provides random access to objects (keyed by instance identifier). Lastly, it provides sequential access to those objects which do not have parents, otherwise known as "root" objects.

See Also:

[RTRManagedObject](#), [RTRClassCategory<RTRManagedObject>](#), [RTRLockableObj](#)

Constructor & Destructor Documentation

RTRManagedObjectDirectory::RTRManagedObjectDirectory ()
 Constructor

RTRManagedObjectDirectory::~~RTRManagedObjectDirectory ()
 Destructor

RTRManagedObjectIterator Class Reference

Stateless iteration on an object's children. Multiple instances of this can be used (in a multi-thread enviroment) for read access.

```
#include <mo.h>
```

Public Member Functions

- [RTRManagedObjectIterator](#) ([RTRManagedObject](#) *)
- [~RTRManagedObjectIterator](#) ()
- int [count](#) () const
- RTRBOOL [off](#) () const
- RTRBOOL [empty](#) () const

- [RTRManagedObject](#) & [item](#) () const
- void [start](#) ()
- void [finish](#) ()
- void [forth](#) ()
- void [back](#) ()

Detailed Description

Stateless iteration on an object's children. Multiple instances of this can be used (in a multi-thread enviroment) for read access.

See Also:

[RTRManagedObject](#), [RTRManagedObjectClient](#)

Constructor & Destructor Documentation

RTRManagedObjectIterator::RTRManagedObjectIterator ([RTRManagedObject](#) *)
 Constructor

RTRManagedObjectIterator::~~RTRManagedObjectIterator ()
 Destructor

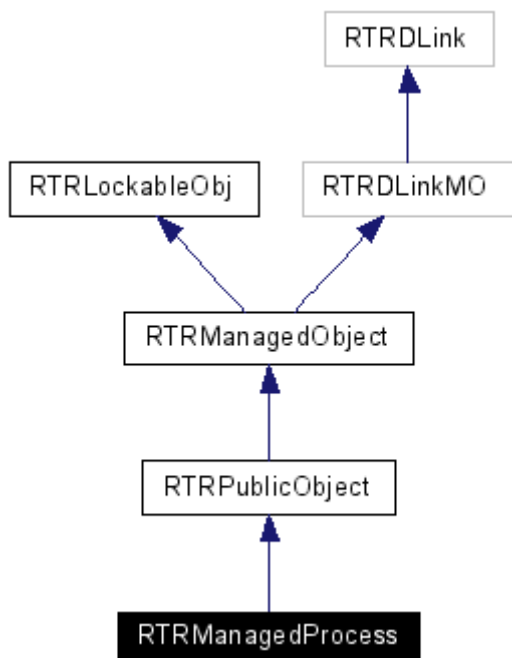
RTRManagedProcess Class Reference

RTRManagedProcess is a descendant of [RTRPublicObject](#) which provides a minimum set of variables relating to process state.

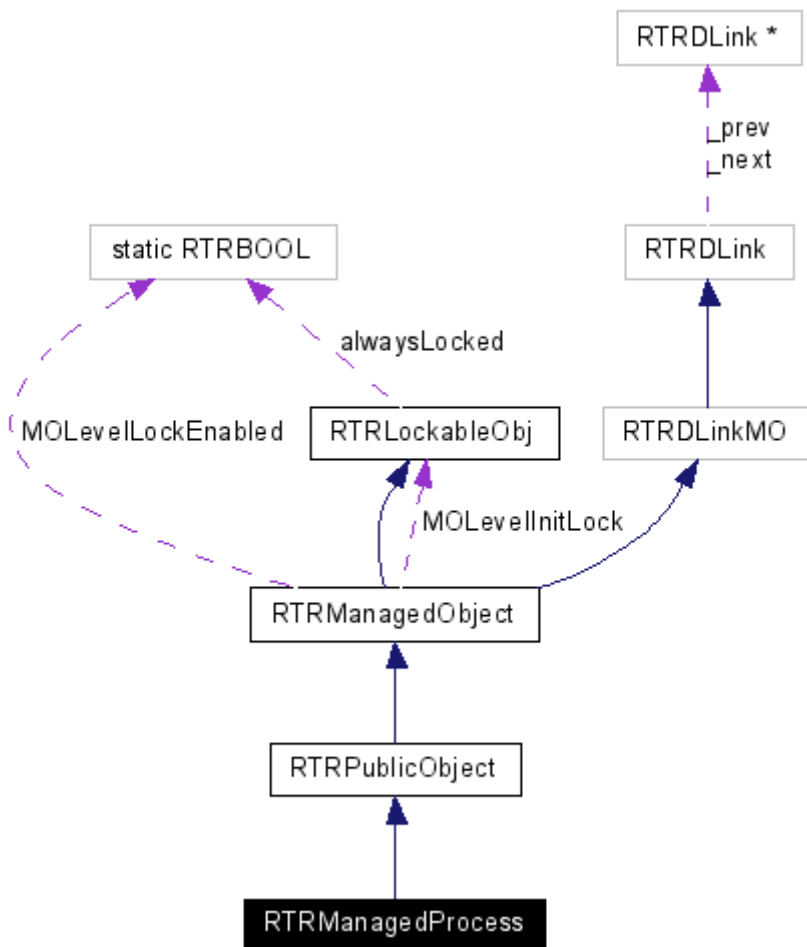
```
#include <mngdproc.h>
```

Inherits [RTRPublicObject](#).

Inheritance diagram for RTRManagedProcess:



Collaboration diagram for RTRManagedProcess:



Public Member Functions

- [RTRManagedProcess](#) (const [RTRObjectId](#) &appId, const char *execName, const char *subClassName, const char *description, const char *version, [MOState](#) startState=RTRManagedObject::Normal)
- [RTRManagedProcess](#) (const [RTRObjectId](#) &appId, const char *execName, const char *description, const char *version, [MOState](#) startState=RTRManagedObject::Normal)
- [RTRManagedProcess](#) (int argc, char **argv, const char *subClassName, const [RTRApplicationId](#) &appId, const [RTRString](#) version)
- virtual [~RTRManagedProcess](#) ()

Detailed Description

RTRManagedProcess is a descendant of [RTRPublicObject](#) which: provides a minimum set of variables relating to process state.

See Also:

Constructor & Destructor Documentation

RTRManagedProcess::RTRManagedProcess (const [RTRObjectId](#) & *appld*, const char * *execName*, const char * *subClassName*, const char * *description*, const char * *version*, [MOState](#) *startState* = RTRManagedObject::Normal)
 Constructor

RTRManagedProcess::RTRManagedProcess (const [RTRObjectId](#) & *appld*, const char * *execName*, const char * *description*, const char * *version*, [MOState](#) *startState* = RTRManagedObject::Normal)
 Constructor

RTRManagedProcess::RTRManagedProcess (int *argc*, char ** *argv*, const char * *subClassName*, const [RTRApplicationId](#) & *appld*, const [RTRString](#) *version*)
 For backward-compatibility

virtual RTRManagedProcess::~~RTRManagedProcess () [virtual]
 Destructor

RTRManagedString Class Reference

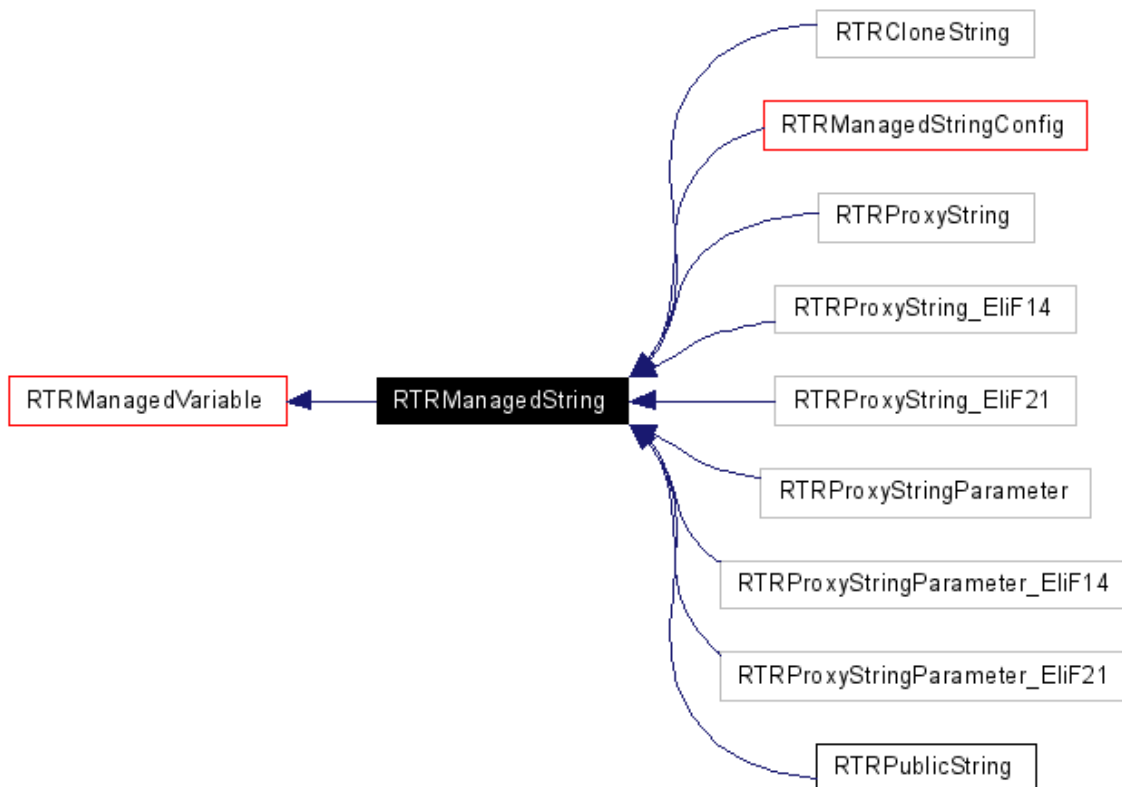
The base class for string variables and provides services for accessing and modifying the value of the ManagedString.

```
#include <mstrvar.h>
```

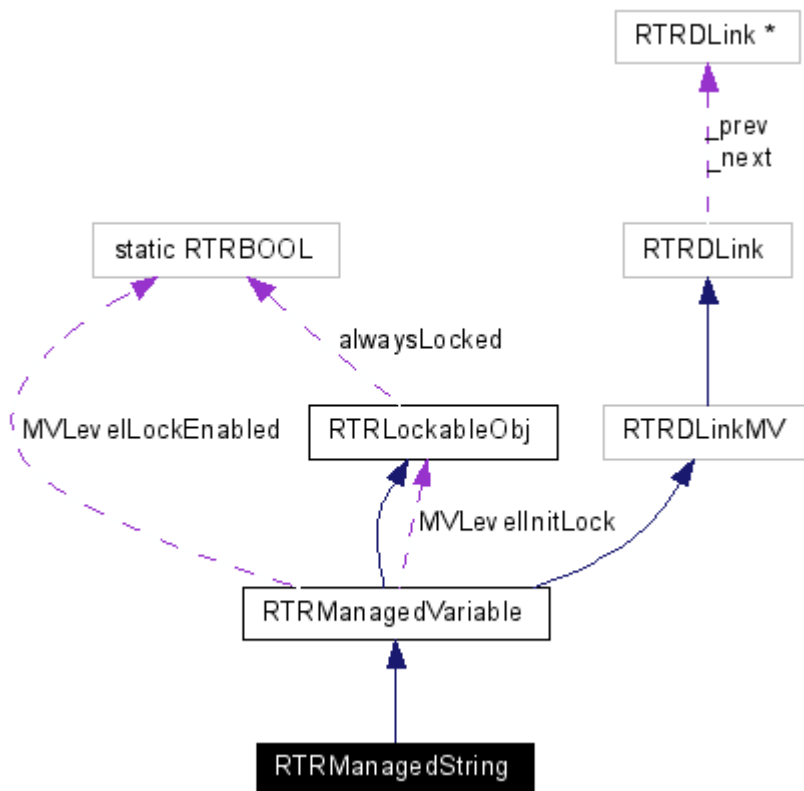
Inherits [RTRManagedVariable](#).

Inherited by [RTRCloneString](#), [RTRManagedStringConfig](#), [RTRProxyString](#), [RTRProxyString_EliF14](#), [RTRProxyString_EliF21](#), [RTRProxyStringParameter](#), [RTRProxyStringParameter_EliF14](#), [RTRProxyStringParameter_EliF21](#), and [RTRPublicString](#).

Inheritance diagram for RTRManagedString:



Collaboration diagram for RTRManagedString:



Public Member Functions

- virtual [~RTRManagedString](#) ()
- [RTRString value](#) () const
- [operator const char *](#) () const
- virtual [RTRString toString](#) () const
- RTRBOOL [operator==](#) (const char *) const
- RTRBOOL [modifyEnabled](#) () const
- [RTRManagedString](#) & [operator=](#) (const char *rhs)
- virtual void [set](#) (const char *newValue)

Friends

- std::ostream & [operator<<](#) (std::ostream &, const [RTRManagedString](#) &)

Detailed Description

The base class for string variables and provides services for accessing and modifying the value of the ManagedString.

Consumers can modify the variable if permitted by the publisher (`modifyEnabled == RTRTRUE`). Permission is granted/denied when the variable is created and cannot be changed during its life-cycle. The context (ManagedObject) will be notified of changes.

This class cannot be instantiated directly.

See Also:

[RTRManagedObject](#), [RTRManagedVariable](#), [RTRManagedNumeric](#), [RTRManagedBoolean](#), [RTRManagedCounter](#), [RTRManagedGauge](#), [RTRManagedNumericRange](#), [RTRManagedStringConfig](#), [RTRManagedNumericConfig](#), [RTRManagedBooleanConfig](#), [RTRManagedGaugeConfig](#)

Constructor & Destructor Documentation

virtual RTRManagedString::~~RTRManagedString () [virtual]
Destructor

RTRManagedStringConfig Class Reference

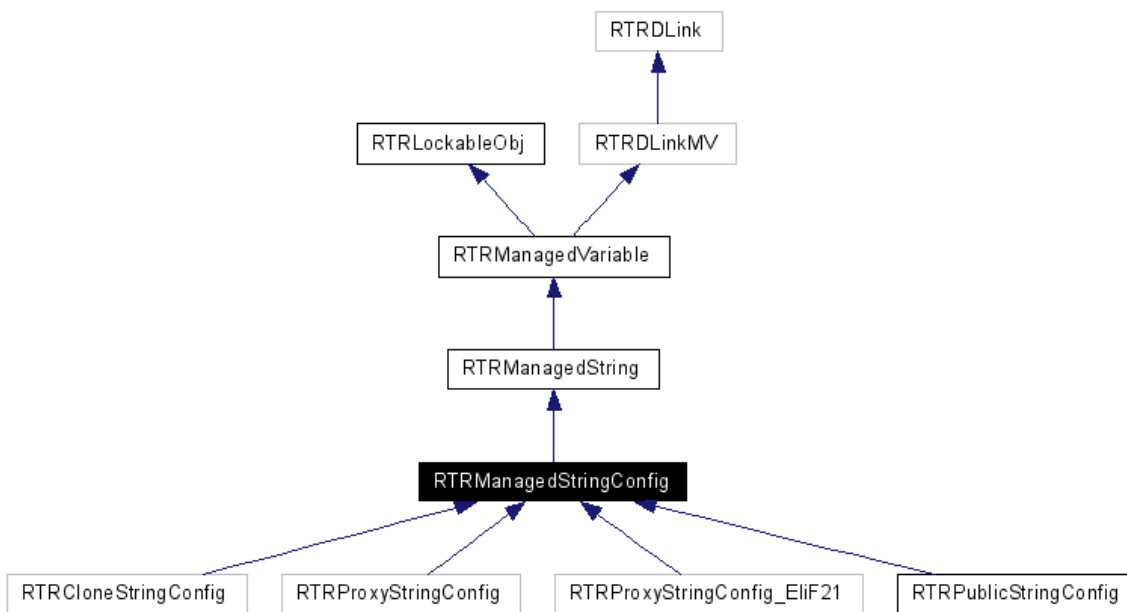
String config inherited from String and provide services for specifying configuration and default values.

```
#include <mstrcvar.h>
```

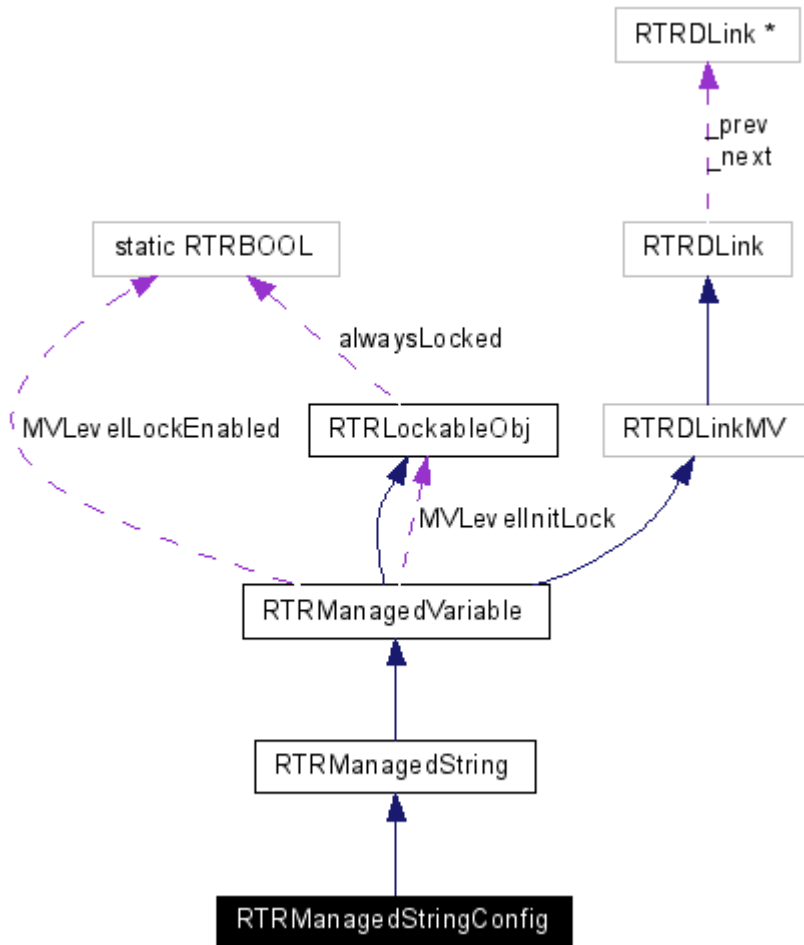
Inherits [RTRManagedString](#).

Inherited by [RTRCloneStringConfig](#), [RTRProxyStringConfig](#), [RTRProxyStringConfig_EliF21](#), and [RTRPublicStringConfig](#).

Inheritance diagram for RTRManagedStringConfig:



Collaboration diagram for RTRManagedStringConfig:



Public Member Functions

- virtual [~RTRManagedStringConfig](#) ()
- [RTRManagedStringConfig](#) & [operator=](#) (const char *rhs)
- [RTRString](#) [activeValue](#) () const
- [RTRString](#) [storeValue](#) () const
- [RTRString](#) [factoryDefault](#) () const
- RTRBOOL [hasStore](#) () const
- RTRBOOL [isStoreActive](#) () const
- RTRBOOL [isStoreClassConfig](#) () const
- RTRBOOL [isStoreInstanceConfig](#) () const

Friends

- class [RTRStrConfigImplPub](#)
- class [RTRManagedMemAllocator](#)

- `std::ostream & operator<< (std::ostream &, const RTRManagedStringConfig &)`

Detailed Description

String config inherit from String and provide services for specifying configuration and default values.

Consumers can modify the variable if permitted by the publisher (`modifyEnabled == RTRTRUE`). Permission is granted/denied when the variable is created and cannot be changed during its life-cycle. The context (ManagedObject) will be notified of changes.

This class cannot be instantiated directly.

See Also:

[RTRManagedObject](#), [RTRManagedVariable](#), [RTRManagedCounter](#), [RTRManagedNumeric](#), [RTRManagedGauge](#), [RTRManagedBoolean](#), [RTRManagedString](#), [RTRManagedNumericRange](#), [RTRManagedBooleanConfig](#), [RTRManagedGaugeConfig](#), [RTRManagedNumericConfig](#)

Constructor & Destructor Documentation

`virtual RTRManagedStringConfig::~RTRManagedStringConfig () [virtual]`
Destructor

RTRManagedVariable Class Reference

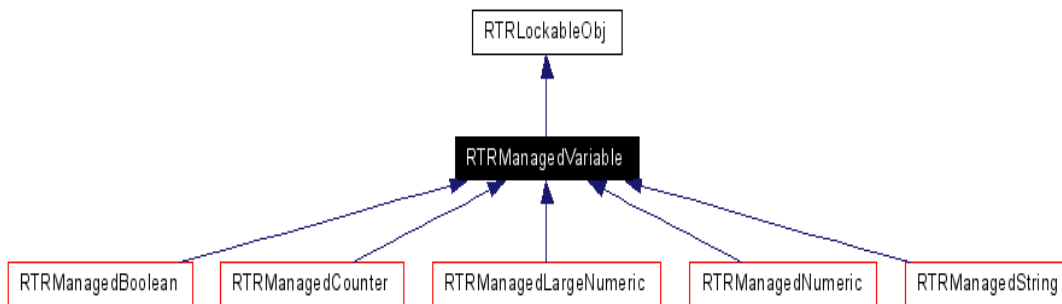
This is the base class for more specific types of managed variables. A managed variable has a name, type, and is contained by an instance of [RTRManagedObject](#). The name of the variable must be unique within the context of the containing object.

```
#include <mvar.h>
```

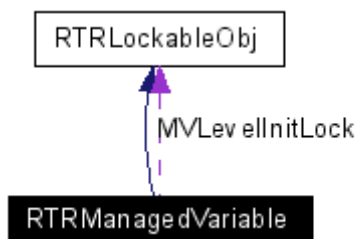
Inherits [RTRLockableObj](#), and [RTRDLinkMV](#).

Inherited by [RTRManagedBoolean](#), [RTRManagedCounter](#), [RTRManagedNumeric](#), and [RTRManagedString](#).

Inheritance diagram for RTRManagedVariable:



Collaboration diagram for RTRManagedVariable:



Public Types

- enum [MVType](#) { Counter, Numeric, NumericRange, Gauge, String, GaugeConfig, NumericConfig, StringConfig, Boolean, BooleanConfig }

Public Member Functions

- virtual [~RTRManagedVariable](#) ()
- const [RTRString](#) & [name](#) () const
- [MVType](#) [type](#) () const
- [RTRManagedObject](#) & [context](#) () const
- const [RTRString](#) & [description](#) () const
- [operator RTRManagedBoolean &](#) () const
- [operator RTRManagedBooleanConfig &](#) () const
- [operator RTRManagedCounter &](#) () const
- [operator RTRManagedGauge &](#) () const
- [operator RTRManagedGaugeConfig &](#) () const
- [operator RTRManagedNumeric &](#) () const
- [operator RTRManagedNumericRange &](#) () const
- [operator RTRManagedNumericConfig &](#) () const
- [operator RTRManagedStringConfig &](#) () const
- [operator RTRManagedString &](#) () const
- virtual [RTRString](#) [toString](#) () const =0
- virtual void [addClient](#) ([RTRManagedVariableClient](#) &newClient)
- virtual void [dropClient](#) ([RTRManagedVariableClient](#) &oldClient)
- RTRBOOL [hasClient](#) ([RTRManagedVariableClient](#) &client) const
- RTRBOOL [hasClients](#) () const
- virtual void [lock](#) ()
- virtual void [unlock](#) ()
- virtual RTRBOOL [locked](#) () const
- RTRMVImp * [storeImpl](#) () const
- RTRMVImpIPub * [storeImplPub](#) () const

Static Public Attributes

- static RTRBOOL [MVLevelLockEnabled](#)
- static [RTRLockableObj](#) [MVLevelInitLock](#)

Friends

- class RTRMVImpIPub

- class [RTRManagedObject](#)

Detailed Description

This is the base class for more specific types of managed variables. A managed variable has a name, type, and is contained by an instance of [RTRManagedObject](#). The name of the variable must be unique within the context of the containing object.

The set of supported variable types is fixed and defined by an enumeration.

Variables may change over time and will propagate changes to registered clients.

This class cannot be instantiated directly.

See Also:

[RTRManagedObject](#), [RTRManagedVariableClient](#), [RTRManagedString](#), [RTRManagedNumeric](#), [RTRManagedCounter](#), [RTRManagedBoolean](#), [RTRManagedNumericRange](#), [RTRManagedGauge](#), [RTRManagedNumericConfig](#), [RTRManagedStringConfig](#), [RTRManagedBooleanConfig](#), [RTRManagedGaugeConfig](#), [RTRPublicObject](#), [RTRString](#)

Member Enumeration Documentation

enum [RTRManagedVariable::MVType](#)

The type of managed variable

Constructor & Destructor Documentation

virtual [RTRManagedVariable::~~RTRManagedVariable](#) () [**virtual**]

Destructor

Member Data Documentation

RTRBOOL [RTRManagedVariable::MVLevelLockEnabled](#) [**static**]

Default is false. Enabled when an instance of variable could be accessed in multiple threads.

[RTRLockableObj](#) [RTRManagedVariable::MVLevelInitLock](#) [**static**]

To synchronize the initialization when MVLevelLockEnabled is true.

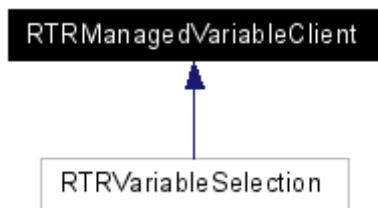
RTRManagedVariableClient Class Reference

The base class for components which can register with a variable to receive changed events from that variable

```
#include <mvar.h>
```

Inherited by [RTRVariableSelection](#).

Inheritance diagram for [RTRManagedVariableClient](#):



Public Member Functions

- virtual void [processVariableChange](#) ([RTRManagedVariable](#) &)=0
- virtual void [processVariableDelete](#) ([RTRManagedVariable](#) &)=0

Detailed Description

The base class for components which can register with a variable to receive change events from that variable.

See Also:

[RTRManagedObject](#), [RTRManagedVariable](#), [RTRManagedString](#), [RTRManagedNumeric](#), [RTRManagedCounter](#), [RTRManagedBoolean](#), [RTRManagedNumericRange](#), [RTRManagedGauge](#), [RTRManagedNumericConfig](#), [RTRManagedStringConfig](#), [RTRManagedBooleanConfig](#), [RTRManagedGaugeConfig](#), [RTRPublicObject](#), [RTRString](#)

RTRManagedVariableIterator Class Reference

Stateless iteration on an object's variables. Multiple instances of this can be used (in a multi-thread enviroment) for read access.

```
#include <mo.h>
```

Public Member Functions

- [RTRManagedVariableIterator](#) ([RTRManagedObject](#) *)
- [~RTRManagedVariableIterator](#) ()
- int [count](#) () const
- RTRBOOL [off](#) () const
- RTRBOOL [empty](#) () const
- [RTRManagedVariable](#) & [item](#) () const
- void [start](#) ()
- void [finish](#) ()
- void [forth](#) ()
- void [back](#) ()

Detailed Description

Stateless iteration on an object's variables. Multiple instances of this can be used (in a multi-thread enviroment) for read access.

See Also:

[RTRManagedObject](#), [RTRManagedObjectClient](#)

Constructor & Destructor Documentation

RTRManagedVariableIterator::RTRManagedVariableIterator ([RTRManagedObject](#) *)
Constructor

RTRManagedVariableIterator::~~RTRManagedVariableIterator ()
Destructor

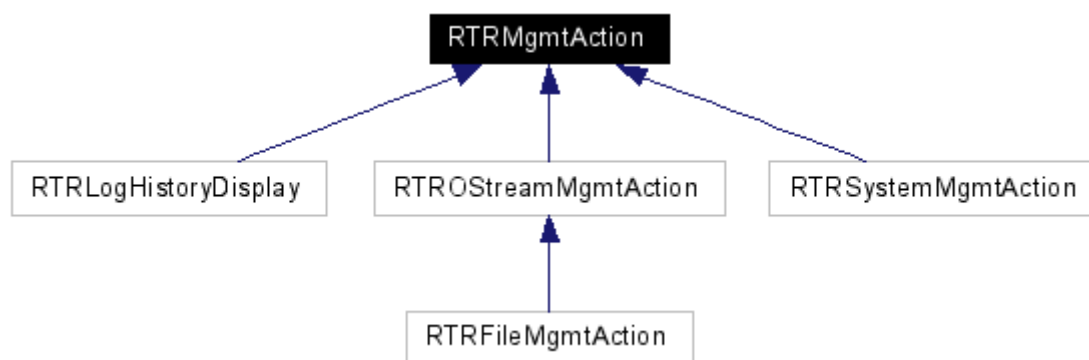
RTRMgmtAction Class Reference

RTRMgmtAction is the abstact base class for components which can be installed with an instance of RTRMgmtEventRouter in order to process application generated management events.

```
#include <mgmtact.h>
```

Inherited by [RTRLogHistoryDisplay](#), [RTROStreamMgmtAction](#), and [RTRSystemMgmtAction](#).

Inheritance diagram for RTRMgmtAction:



Public Member Functions

- [RTRMgmtAction](#) (const [RTRObjectId](#) &classOfAction, const [RTRObjectId](#) &id)
- [RTRMgmtAction](#) (const [RTRObjectId](#) &classOfAction, const [RTRObjectId](#) &id, const RTRMgmtEventFilter &filter)
- virtual [~RTRMgmtAction](#) ()
- const [RTRObjectId](#) & [classId](#) () const
- const [RTRObjectId](#) & [instanceId](#) () const
- const RTRMgmtEventFilter & [filter](#) () const
- RTRBOOL [installed](#) () const
- virtual void [processMgmtEvent](#) (const [RTRMgmtEvent](#) &)
- virtual void [processFilteredMgmtEvent](#) (const [RTRMgmtEvent](#) &)=0
- void [install](#) ()
- void [deinstall](#) ()
- void [setFilter](#) (const RTRMgmtEventFilter &)

Detailed Description

RTRMgmtAction is the abstract base class for components which can be installed with an instance of RTRMgmtEventRouter in order to process application generated management events.

Descendants must implement [processFilteredMgmtEvent\(\)](#), and may override the default implementation of [processMgmtEvent\(\)](#) in order to provide specialized filtering. RTRMgmtAction creates a default filter (passes RTRMgmtError::Error and above from all components) which may be overridden on the constructor or via the [setFilter\(\)](#) method.

An action must be installed with the router. The [install\(\)](#) method is provided for this purpose. The base class destructor calls [deinstall\(\)](#) if necessary.

See Also:

[RTRMgmtEvent](#), RTRMgmtEventRouter

Constructor & Destructor Documentation

RTRMgmtAction::RTRMgmtAction (const [RTRObjectId](#) & *classOfAction*, const [RTRObjectId](#) & *id*)
 Constructor

```
RTRMgmtAction::RTRMgmtAction (const RTRObjectld & classOfAction, const RTRObjectld & id, const RTRMgmtEventFilter
& filter)
    An action with the given filter (copied).

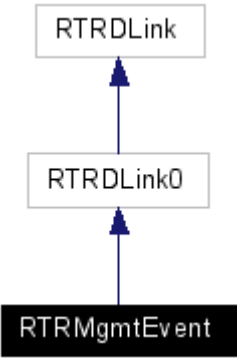
virtual RTRMgmtAction::~~RTRMgmtAction () [virtual]
    Destructor
```

RTRMgmtEvent Class Reference

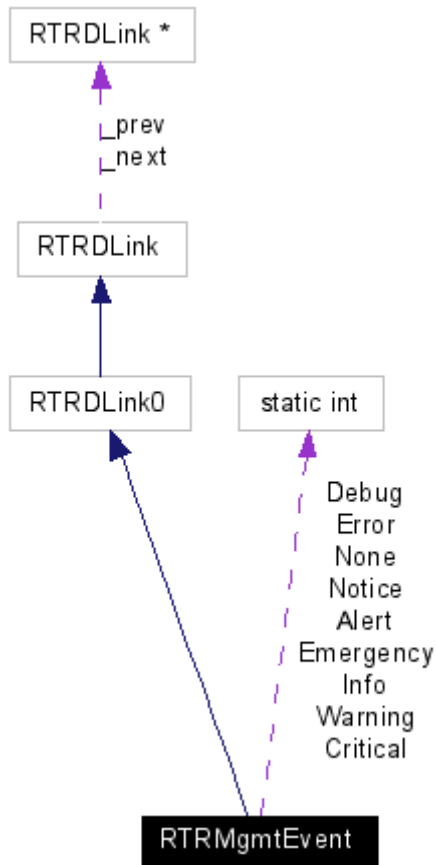
RTRMgmtEvent provides the means for managed applications to generate events for processing by managing applications. Events have an identifier (component), text, severity, and a timestamp. The identifier is that of the component generating the event. Text is descriptive information about the event. Severity is a value between RTRMgmtEvent::Emergency and RTRMgmtEvent::Debug. A timestamp will be generated automatically if none is set.

```
#include <mgmtevt.h>
Inherits RTRDLink0.
```

Inheritance diagram for RTRMgmtEvent:



Collaboration diagram for RTRMgmtEvent:



Public Member Functions

- [RTRMgmtEvent](#) ()
- [RTRMgmtEvent](#) (const [RTRObjectId](#) &, const [RTRString](#) &, int)
- [RTRMgmtEvent](#) (const [RTRObjectId](#) &, const [RTRString](#) &, int, unsigned long)
- [RTRMgmtEvent](#) (const [RTRMgmtEvent](#) &)
- const [RTRObjectId](#) & [instanceId](#) () const
- const [RTRString](#) & [text](#) () const
- int [severity](#) () const
- const [RTRDateTime](#) & [timestamp](#) () const
- [RTRBOOL](#) [operator<](#) (const [RTRMgmtEvent](#) &) const
- [RTRBOOL](#) [operator>](#) (const [RTRMgmtEvent](#) &) const
- [RTRBOOL](#) [operator==](#) (const [RTRMgmtEvent](#) &) const
- [RTRBOOL](#) [operator<=](#) (const [RTRMgmtEvent](#) &) const
- [RTRBOOL](#) [operator>=](#) (const [RTRMgmtEvent](#) &) const
- void [setIdIdentifier](#) (const [RTRObjectId](#) &)

- void [setSeverity](#) (int)
- void [setSeverity](#) (const [RTRString](#) &)
- void [setText](#) (const char *)
- void [setText](#) (const [RTRString](#) &)
- void [setTimestamp](#) ()
- void [log](#) ()
- [RTRMgmtEvent](#) & **operator=** (const [RTRMgmtEvent](#) &)
- void [setComponent](#) (const char *c)
- void [setComponent](#) (const [RTRString](#) &c)

Static Public Member Functions

- static int [severity](#) (const [RTRString](#) &)
- static [RTRString severityString](#) (int)

Static Public Attributes

- static int **Emergency**
- static int **Alert**
- static int **Critical**
- static int **Error**
- static int **Warning**
- static int **Notice**
- static int **Info**
- static int **Debug**
- static int **None**

Detailed Description

RTRMgmtEvent provides the means for managed applications to generate events for processing by managing applications. Events have an identifier (component), text, severity, and a timestamp. The identifier is that of the component generating the event. Text is descriptive information about the event. Severity is a value between RTRMgmtEvent::Emergency and RTRMgmtEvent::Debug. A timestamp will be generated automatically if none is set.

RTRMgmtEvent provides comparison operators based on the timestamp attribute. Newer events are greater than older events.

```
RTRMgmtEvent event;
event.setIdentifier("instanceId_or_string");
event.setText("Event text");
event.setSeverity(RTRMgmtEvent::Alert);
event.log();
```

See Also:

RTRLogEvent, [RTRObjectId](#), [RTRString](#)

Constructor & Destructor Documentation

RTRMgmtEvent::RTRMgmtEvent ()
 A blank event.

ENSURE: [severity\(\)](#) == Info

RTRMgmtEvent::RTRMgmtEvent (const [RTRObjectld](#) &, const [RTRString](#) &, int)
 A new event with the given attributes

RTRMgmtEvent::RTRMgmtEvent (const [RTRObjectld](#) &, const [RTRString](#) &, int, unsigned *long*)
 A new event with the given attributes (and timestamp).

RTRMgmtEvent::RTRMgmtEvent (const [RTRMgmtEvent](#) &)
 Constructor

Member Function Documentation

static int RTRMgmtEvent::severity (const [RTRString](#) &) [static]
 The integer value of the given severity string. Result will be "None" if string is otherwise invalid.

static [RTRString](#) RTRMgmtEvent::severityString (int) [static]
 The string value of the given severity. Result will be "None" if string is invalid.

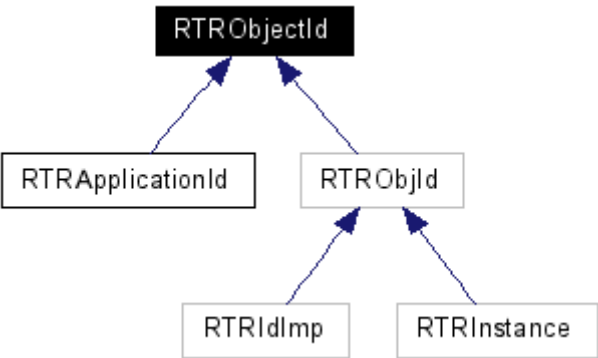
RTRObjectld Class Reference

An object identifier. Both instance identifiers and class identifiers can be represented by instances of RTRObjectld.

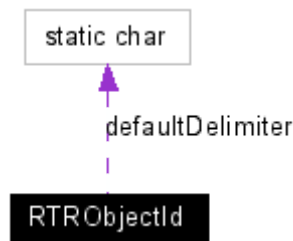
```
#include <objid.h>
```

Inherited by [RTRApplicationId](#), and [RTRObjld](#).

Inheritance diagram for RTRObjectld:



Collaboration diagram for RTRObjectld:



Public Member Functions

- [RTRObjctId](#) ()
- [RTRObjctId](#) (const [RTRObjctId](#) &cntxt, const [RTRString](#) &nm)
- [RTRObjctId](#) (const char *value)
- [RTRObjctId](#) (const char *value, unsigned int l)
- [RTRObjctId](#) (const [RTRObjctId](#) &id)
- virtual [~RTRObjctId](#) ()
- [RTRString name](#) () const
- [RTRString base](#) () const
- int [numberOfElements](#) () const
- int [count](#) () const
- unsigned long [hash](#) () const
- RTRBOOL [isEmpty](#) () const
- RTRLinkedList< [RTRString](#) > * [lineage](#) () const
- [RTRString iTh](#) (int i) const
- RTRBOOL [isDescendant](#) (const [RTRObjctId](#) &other) const
- RTRBOOL [conformsTo](#) (const [RTRObjctId](#) &other) const
- RTRBOOL [operator==](#) (const [RTRObjctId](#) &other) const
- RTRBOOL [operator!=](#) (const [RTRObjctId](#) &other) const
- RTRBOOL [operator==](#) (const [RTRString](#) &other) const
- RTRBOOL [operator==](#) (const char *other) const
- operator const char * () const
- [RTRString string](#) () const
- [RTRString delimitedString](#) (char delimiter) const
- [RTRObjctId firstN](#) (int n) const
- [RTRObjctId lastN](#) (int n) const
- [RTRObjctId parent](#) () const
- [RTRObjctId commonRoot](#) (const [RTRObjctId](#) &other) const

- [RTRObjectId](#) & [operator=](#) (const [RTRObjectId](#) &rhs)
- void [set](#) ([RTRString](#) &s, int n1, int n2)
- const char * [to_c](#) () const
- RTRBOOL [equivalent](#) (const [RTRObjectId](#) &other) const

Static Public Attributes

- static char defaultDelimiter

Friends

- std::ostream & [operator<<](#) (std::ostream &, const [RTRObjectId](#) &)

Detailed Description

An object identifier. Both instance identifiers and class identifiers can be represented by instances of [RTRObjectId](#).

See Also:

[RTRString](#)

Constructor & Destructor Documentation

RTRObjectId::RTRObjectId ()
Construct a blank object id.

RTRObjectId::RTRObjectId (const [RTRObjectId](#) & cntxt, const [RTRString](#) & nm)
Construct an object id composed of a context and a name. The final object id string will be "<cntxt><delimiter><nm>". cntxt and nm may be empty values.

RTRObjectId::RTRObjectId (const char * value)
Construct an object id using the given character string. The string representation of the object id is <value>.

RTRObjectId::RTRObjectId (const char * value, unsigned int l)
Construct an object id using the given character string. The string representation of the object id is the first l characters of <value>.

RTRObjectId::RTRObjectId (const [RTRObjectId](#) & id)
Copy constructor.

virtual RTRObjectId::~~RTRObjectId () [virtual]
Destructor

RTRProxyManagedBoolean Class Reference

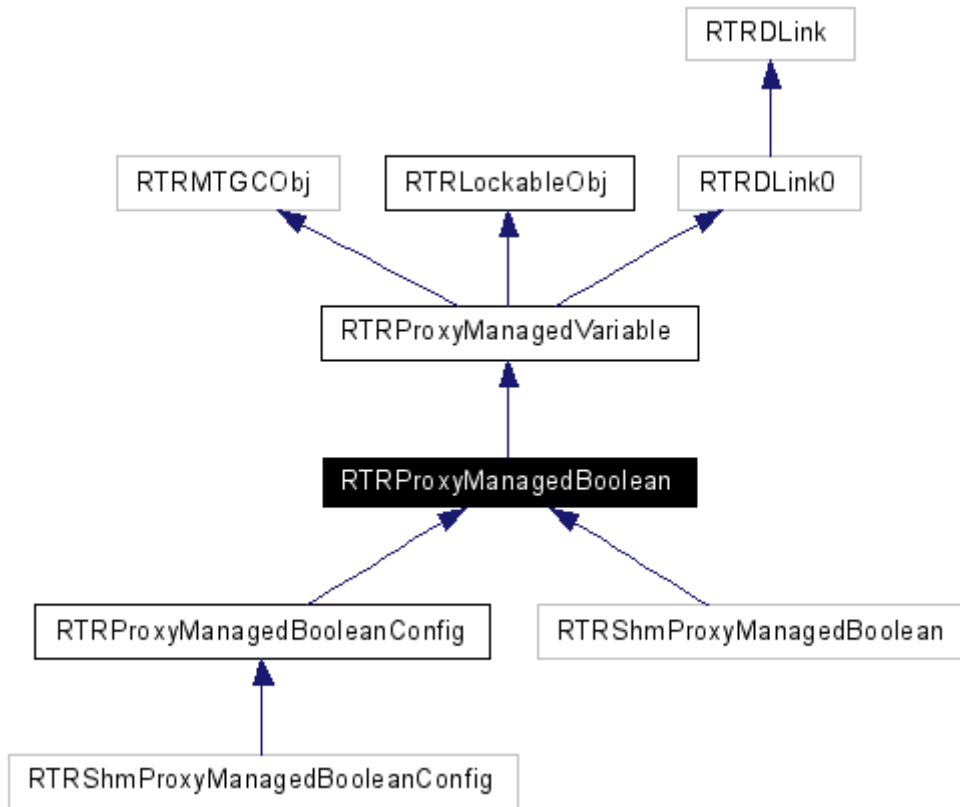
A cloned (proxy) representation of a Boolean variable. The base class for proxy boolean managed variables. Inherits from [RTRProxyManagedVariable](#) and provides services for accessing and (conditionally) modifying a managed variable of type boolean. The managed application will accept modifications to this variable if the [modifyEnabled\(\)](#) attribute is true. Accepted modifications are limited to setting the value of the variable to true or false.

```
#include <prxymb.h>
```

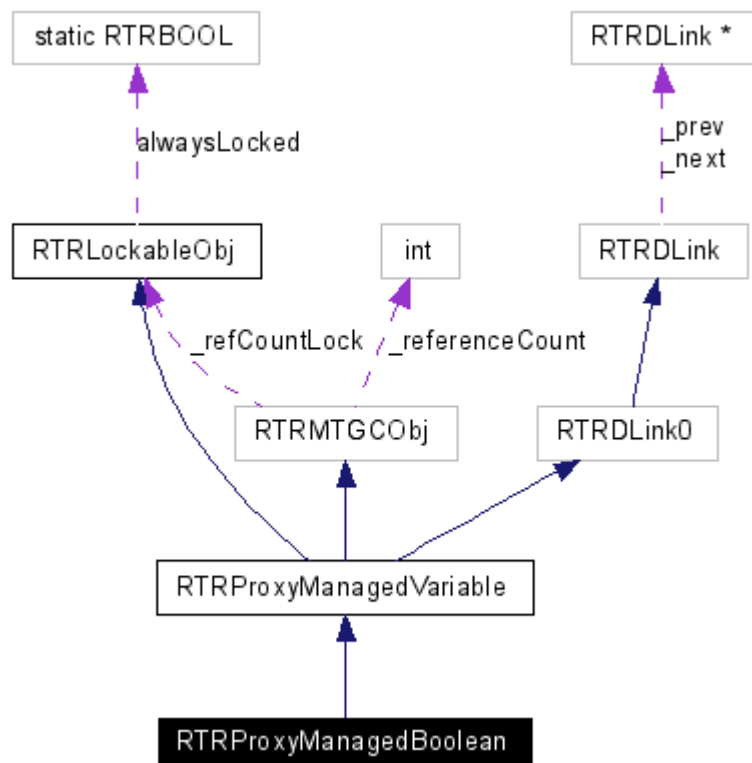
Inherits [RTRProxyManagedVariable](#).

Inherited by [RTRProxyManagedBooleanConfig](#), and [RTRShmProxyManagedBoolean](#).

Inheritance diagram for [RTRProxyManagedBoolean](#):



Collaboration diagram for RTRProxyManagedBoolean:



Public Member Functions

- [RTRProxyManagedBoolean](#) ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
- virtual [~RTRProxyManagedBoolean](#) ()
- [RTRBOOL](#) [operator==](#) ([RTRBOOL](#) rhs) const
- [RTRBOOL](#) [value](#) () const
- virtual [RTRString](#) [toString](#) () const
- operator [RTRProxyManagedBooleanConfig](#) & ()
- operator const [RTRProxyManagedBooleanConfig](#) & () const
- [RTRBOOL](#) [modifyEnabled](#) () const
- [RTRProxyManagedBoolean](#) & [operator=](#) ([RTRBOOL](#) rhs)
- virtual void [set](#) ()=0
- virtual void [clear](#) ()=0

Detailed Description

A cloned (proxy) representation of a Boolean variable. The base class for proxy boolean managed variables. Inherits from [RTRProxyManagedVariable](#) and provides services for accessing and (conditionally) modifying a managed variable of type boolean. The managed application will accept modifications to this variable if the [modifyEnabled\(\)](#) attribute is true. Accepted modifications are limited to setting the value of the variable to true or false.

See Also:

[RTRProxyManagedVariableClient](#), [RTRProxyManagedCounter](#), [RTRProxyManagedNumeric](#), [RTRProxyManagedString](#), [RTRProxyManagedGauge](#), [RTRProxyManagedNumericRange](#), [RTRProxyManagedNumericConfig](#), [RTRProxyManagedStringConfig](#), [RTRProxyManagedGaugeConfig](#), [RTRProxyManagedBooleanConfig](#)

Constructor & Destructor Documentation

RTRProxyManagedBoolean::RTRProxyManagedBoolean ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
Constructor

virtual RTRProxyManagedBoolean::~~RTRProxyManagedBoolean () [virtual]
Destructor

RTRProxyManagedBooleanConfig Class Reference

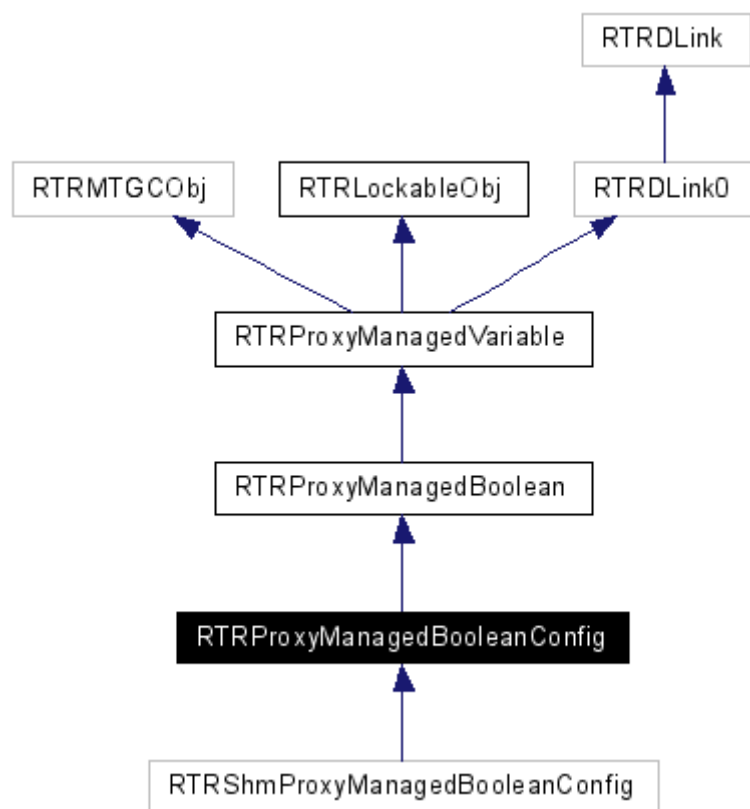
A cloned (proxy) representation of a BooleanConfig variable. The base class for proxy boolean managed configuration variables. Inherits from [RTRProxyManagedBoolean](#) and provides additional services for accessing the stored and default values of a managed variable of type boolean config. The managed application will accept modifications to this variable if the [modifyEnabled\(\)](#) attribute is true. Accepted modifications are limited to setting the active value of the variable to true or false (the stored and default values cannot be modified).

```
#include <prxymbc.h>
```

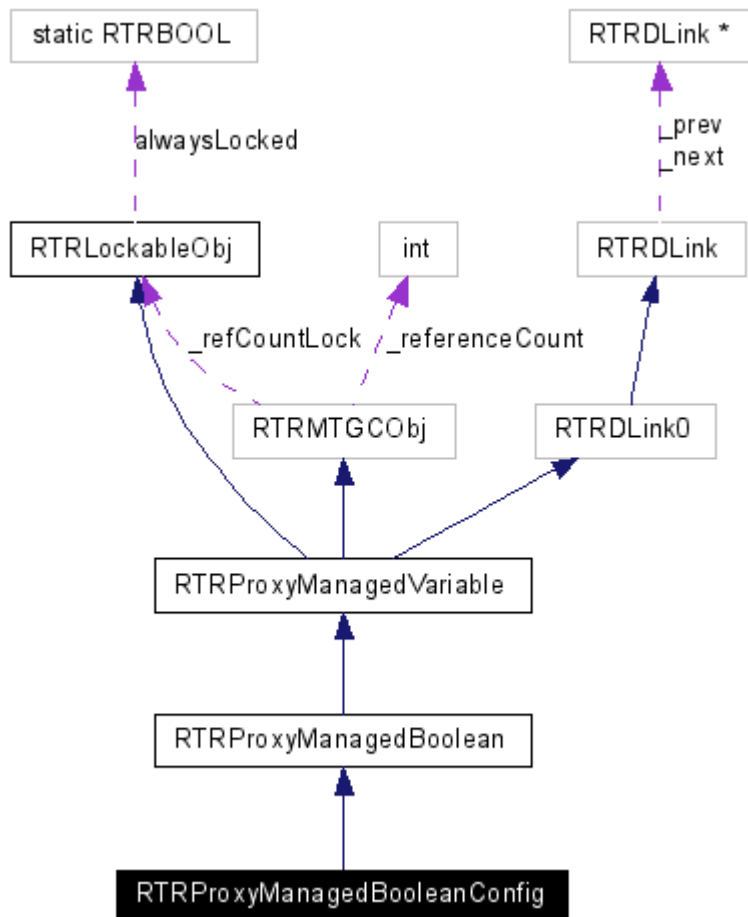
Inherits [RTRProxyManagedBoolean](#).

Inherited by RTRShmProxyManagedBooleanConfig.

Inheritance diagram for RTRProxyManagedBooleanConfig:



Collaboration diagram for RTRProxyManagedBooleanConfig:



Public Member Functions

- [RTRProxyManagedBooleanConfig](#) ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
- virtual [~RTRProxyManagedBooleanConfig](#) ()
- [RTRProxyManagedBooleanConfig](#) & [operator=](#) ([RTRBOOL](#) rhs)
- [RTRBOOL](#) [activeValue](#) () const
- [RTRBOOL](#) [storeValue](#) () const
- [RTRBOOL](#) [factoryDefault](#) () const

Detailed Description

A cloned (proxy) representation of a BooleanConfig variable. The base class for proxy boolean managed configuration variables. Inherits from [RTRProxyManagedBoolean](#) and provides additional services for accessing the stored and default values of a managed variable of type boolean config. The managed application will accept modifications to this variable if the [modifyEnabled\(\)](#) attribute is true. Accepted modifications are limited to setting the active value of the variable to true or false (the stored and default values cannot be modified).

See Also:

[RTRProxyManagedVariableClient](#), [RTRProxyManagedCounter](#), [RTRProxyManagedNumeric](#), [RTRProxyManagedString](#), [RTRManagedBoolean](#), [RTRManagedGauge](#), [RTRProxyManagedNumericRange](#), [RTRProxyManagedNumericConfig](#), [RTRProxyManagedStringConfig](#), [RTRProxyManagedGaugeConfig](#)

Constructor & Destructor Documentation

RTRProxyManagedBooleanConfig::RTRProxyManagedBooleanConfig ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)

Constructor

virtual RTRProxyManagedBooleanConfig::~RTRProxyManagedBooleanConfig () [virtual]

Destructor

RTRProxyManagedCounter Class Reference

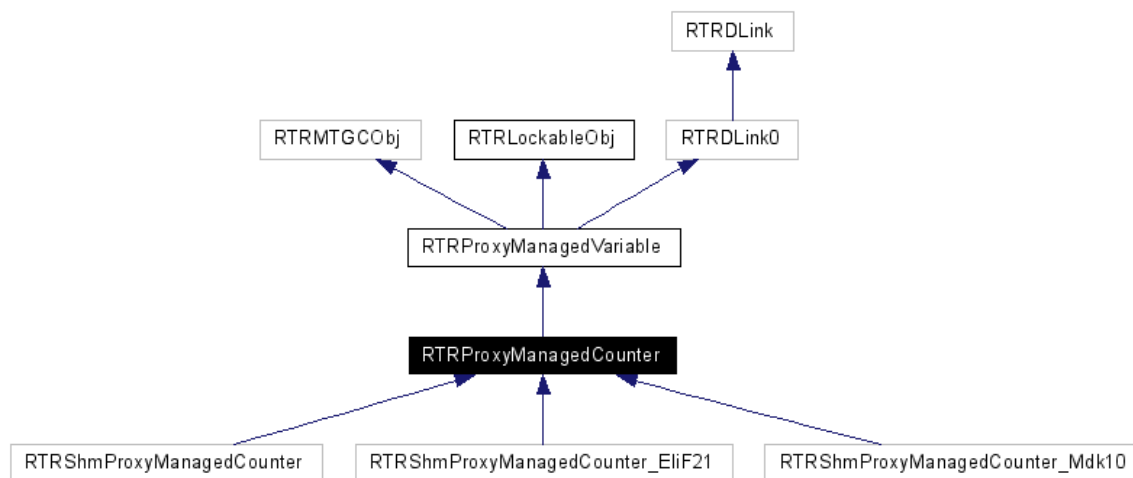
A cloned (proxy) representation of a Counter variable. The base class for proxy counter managed variables. Inherits from [RTRProxyManagedVariable](#) and provides services for accessing and resetting (to 0) the value of a managed variable of type counter. The managed application will always accept reset modification requests.

```
#include <prxymc.h>
```

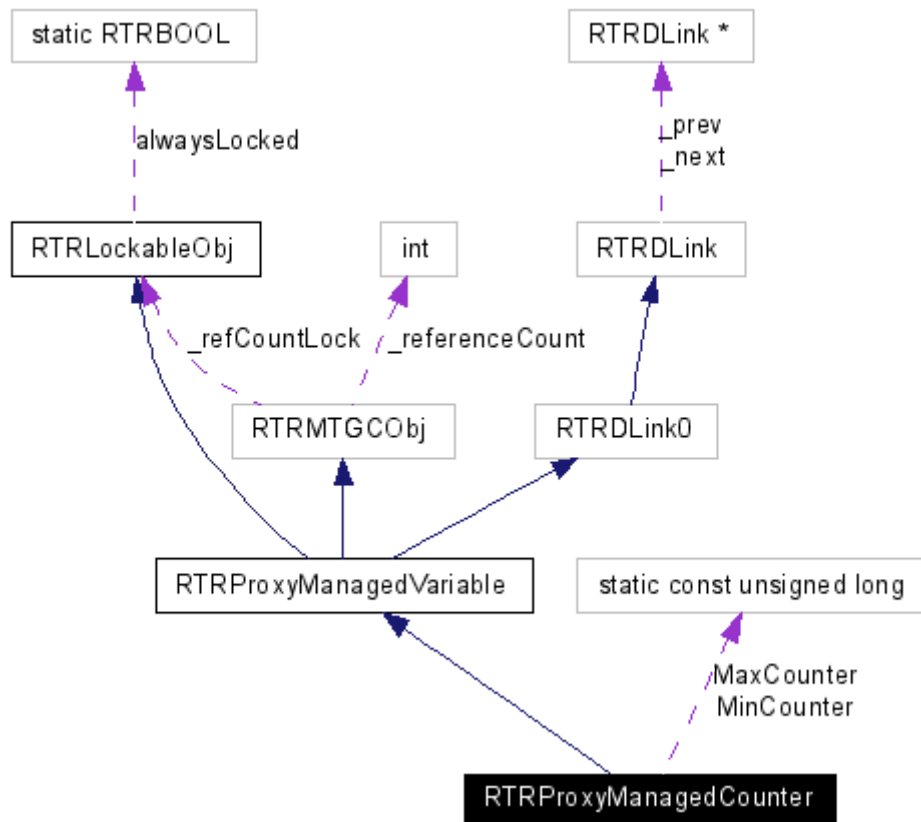
Inherits [RTRProxyManagedVariable](#).

Inherited by RTRShmProxyManagedCounter, RTRShmProxyManagedCounter_EliF21, and RTRShmProxyManagedCounter_Mdk10.

Inheritance diagram for RTRProxyManagedCounter:



Collaboration diagram for RTRProxyManagedCounter:



Public Member Functions

- [RTRProxyManagedCounter](#) ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
- virtual [~RTRProxyManagedCounter](#) ()
- RTRBOOL [operator==](#) (unsigned long rhs) const
- unsigned long [value](#) () const
- virtual [RTRString toString](#) () const
- [operator unsigned long](#) () const
- virtual void [reset](#) ()=0

Static Public Attributes

- static const unsigned long **MinCounter**
- static const unsigned long **MaxCounter**

Detailed Description

A cloned (proxy) representation of a Counter variable. The base class for proxy counter managed variables. Inherits from [RTRProxyManagedVariable](#) and provides services for accessing and resetting (to 0) the value of a managed variable of type counter. The managed application will always accept reset modification requests.

See Also:

[RTRProxyManagedVariableClient](#), [RTRProxyManagedGauge](#), [RTRProxyManagedNumeric](#), [RTRProxyManagedBoolean](#), [RTRProxyManagedString](#), [RTRProxyManagedNumericRange](#), [RTRProxyManagedNumericConfig](#), [RTRProxyManagedStringConfig](#), [RTRProxyManagedBooleanConfig](#), [RTRProxyManagedGaugeConfig](#)

Constructor & Destructor Documentation

RTRProxyManagedCounter::RTRProxyManagedCounter ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
Constructor

virtual RTRProxyManagedCounter::~RTRProxyManagedCounter () [virtual]
Destructor

RTRProxyManagedGauge Class Reference

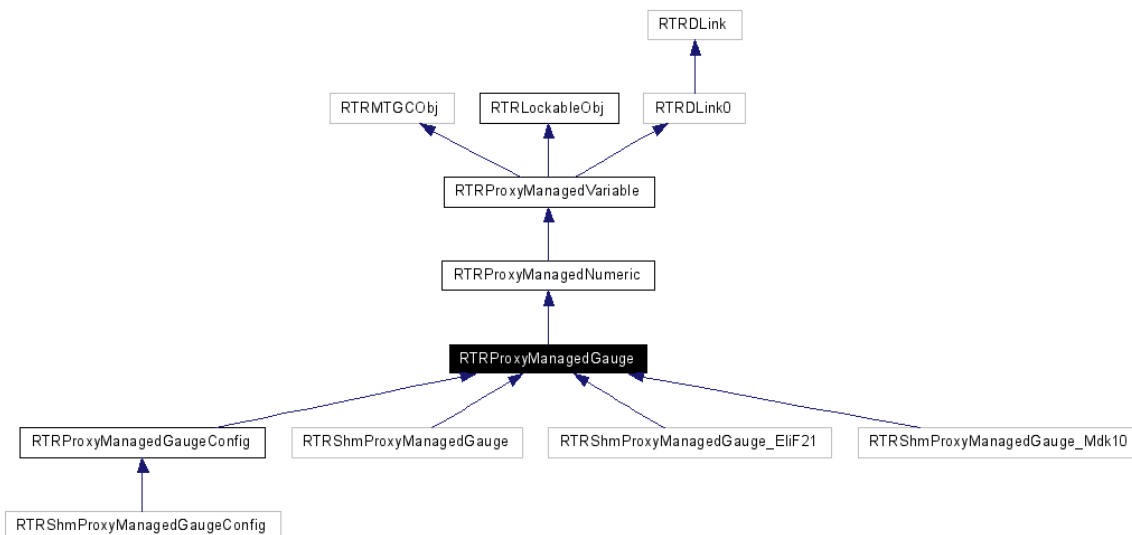
A cloned (proxy) representation of a Gauge variable. The base class for proxy gauge managed variables. Inherits from [RTRProxyManagedNumeric](#) and provides additional services for accessing and modifying a managed variable of type gauge.

```
#include <prxymg.h>
```

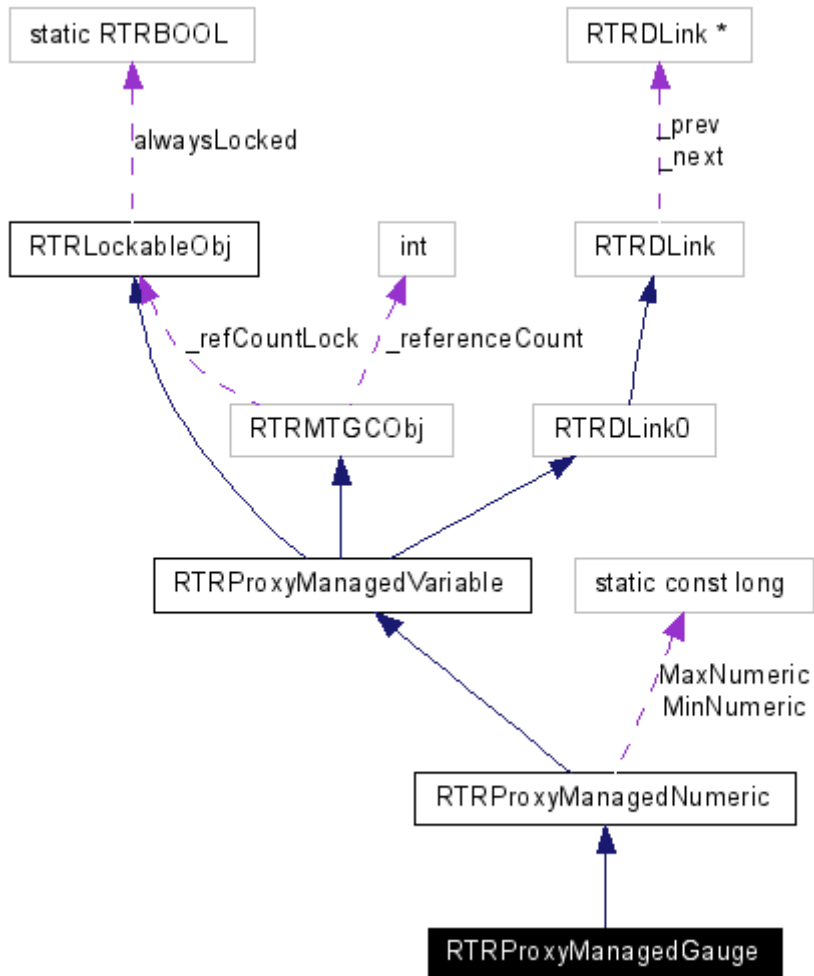
Inherits [RTRProxyManagedNumeric](#).

Inherited by [RTRProxyManagedGaugeConfig](#), [RTRShmProxyManagedGauge](#), [RTRShmProxyManagedGauge_EliF21](#), and [RTRShmProxyManagedGauge_Mdk10](#).

Inheritance diagram for RTRProxyManagedGauge:



Collaboration diagram for RTRProxyManagedGauge:



Public Member Functions

- [RTRProxyManagedGauge](#) ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
- virtual [~RTRProxyManagedGauge](#) ()
- operator [RTRProxyManagedGaugeConfig](#) & ()
- [operator const RTRProxyManagedGaugeConfig &](#) () const
- long [minValue](#) () const
- long [maxValue](#) () const
- long [lowWaterMark](#) () const
- long [highWaterMark](#) () const
- [RTRBOOL modifyEnabled](#) () const
- virtual void [setRange](#) (long newMin, long newMax)=0

Detailed Description

A cloned (proxy) representation of a Gauge variable. The base class for proxy gauge managed variables. Inherits from [RTRProxyManagedNumeric](#) and provides additional services for accessing and modifying a managed variable of type gauge.

In addition to the current value, this variable also contains a minimum, maximum, low water mark and high water mark values. The low and high water marks indicate the lowest and highest values that the variable has assumed since the variable was created.

The minimum and maximum values specify a desired range of values that the variable can assume.

If [modifyEnabled\(\)](#) is False, then the values that the variable can assume are strictly enforced. The value will always be between [minValue\(\)](#) and [maxValue\(\)](#) (inclusive). Note, however, that the min and max values can be changed by the producer, but the current value will be within the new range. When [modifyEnabled\(\)](#) is False, then consumers are not permitted to modify the variable.

If [modifyEnabled\(\)](#) is True, then the consumer is permitted to change the range (the min/max values). In this case, the current value does not have to be within the min/max range. Note: The consumer cannot modify the current value.

See Also:

[RTRProxyManagedVariableClient](#), [RTRProxyManagedCounter](#), [RTRProxyManagedNumeric](#), [RTRProxyManagedString](#), [RTRProxyManagedBoolean](#), [RTRProxyManagedNumericRange](#), [RTRProxyManagedNumericConfig](#), [RTRProxyManagedStringConfig](#), [RTRProxyManagedBooleanConfig](#), [RTRProxyManagedGaugeConfig](#)

Constructor & Destructor Documentation

RTRProxyManagedGauge::RTRProxyManagedGauge ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
 Constructor

virtual RTRProxyManagedGauge::~RTRProxyManagedGauge () [virtual]
 Destructor

RTRProxyManagedGaugeConfig Class Reference

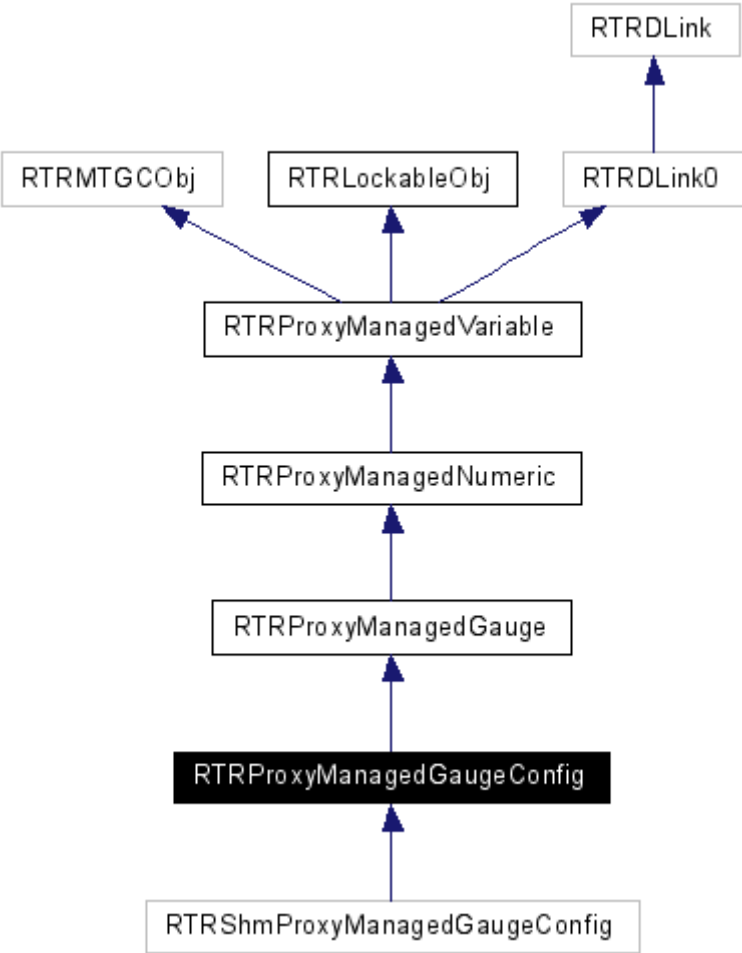
A cloned (proxy) representation of a GaugeConfig variable. The base class for proxy gauge managed configuration variables. Inherits from [RTRProxyManagedGauge](#) and provides additional services for accessing the stored and default values for the minimum and maximum values.

```
#include <prxymgc.h>
```

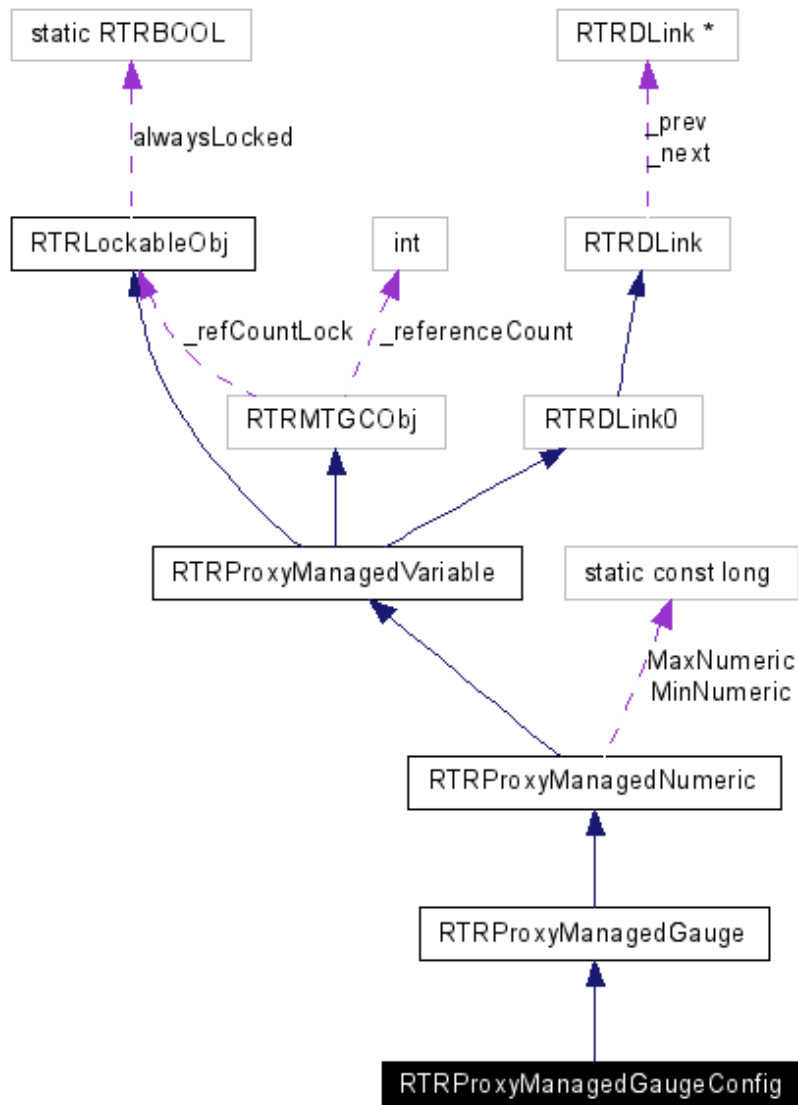
Inherits [RTRProxyManagedGauge](#).

Inherited by [RTRShmProxyManagedGaugeConfig](#).

Inheritance diagram for [RTRProxyManagedGaugeConfig](#):



Collaboration diagram for RTRProxyManagedGaugeConfig:



Public Member Functions

- [RTRProxyManagedGaugeConfig](#) ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
- virtual [~RTRProxyManagedGaugeConfig](#) ()
- long [minStoreValue](#) () const
- long [minFactoryDefault](#) () const
- long [maxStoreValue](#) () const
- long [maxFactoryDefault](#) () const

Detailed Description

A cloned (proxy) representation of a GaugeConfig variable. The base class for proxy gauge managed configuration variables. Inherits from [RTRProxyManagedGauge](#) and provides additional services for accessing the stored and default values for the minimum and maximum values.

See Also:

[RTRProxyManagedVariableClient](#), [RTRProxyManagedCounter](#), [RTRProxyManagedNumeric](#), [RTRProxyManagedString](#), [RTRProxyManagedBoolean](#), [RTRProxyManagedGauge](#), [RTRProxyManagedNumericRange](#), [RTRProxyManagedNumericConfig](#), [RTRProxyManagedStringConfig](#), [RTRProxyManagedBooleanConfig](#), [RTRProxyManagedGaugeConfig](#)

Constructor & Destructor Documentation

RTRProxyManagedGaugeConfig::RTRProxyManagedGaugeConfig ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
 Constructor

virtual RTRProxyManagedGaugeConfig::~RTRProxyManagedGaugeConfig () [virtual]
 Destructor

RTRProxyManagedLargeNumeric Class Reference

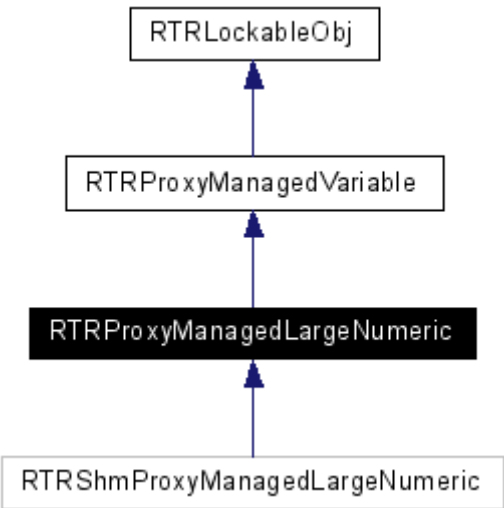
A cloned (proxy) representation of a Large Numeric variable. The base class for large proxy numeric managed variables. Inherits from [RTRProxyManagedVariable](#) and provides services for accessing the current value of the variable.

```
#include <prxym1n.h>
```

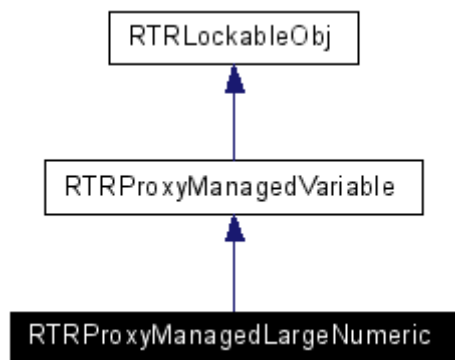
Inherits [RTRProxyManagedVariable](#).

Inherited by RTRShmProxyManagedLargeNumeric

Inheritance diagram for RTRProxyManagedLargeNumeric:



Collaboration diagram for RTRProxyManagedLargeNumeric:



Public Member Functions

- [RTRProxyManagedLargeNumeric](#) ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
- virtual [~RTRProxyManagedLargeNumeric](#) ()
- RTRBOOL [operator==](#) (RTR_I64 rhs) const
- RTR_I64 [value](#) () const
- virtual [RTRString toString](#) () const
- [operator RTR_I64](#) () const

Static Public Attributes

- static const RTR_I64 **LongMinNumeric**
- static const RTR_I64 **LongMaxNumeric**

Detailed Description

A cloned (proxy) representation of a Large Numeric variable. The base class for large proxy numeric managed variables. Inherits from [RTRProxyManagedVariable](#) and provides services for accessing the current value of the variable.

See Also:

[RTRProxyManagedVariableClient](#), [RTRProxyManagedCounter](#), [RTRProxyManagedGauge](#), [RTRProxyManagedString](#), [RTRProxyManagedBoolean](#), [RTRProxyManagedNumericRange](#), [RTRProxyManagedNumericConfig](#), [RTRProxyManagedStringConfig](#), [RTRProxyManagedNumeric](#), [RTRProxyManagedBooleanConfig](#), [RTRProxyManagedGaugeConfig](#)

Constructor & Destructor Documentation

RTRProxyManagedLargeNumeric::RTRProxyManagedLargeNumeric ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
 Constructor

virtual RTRProxyManagedLargeNumeric::~~RTRProxyManagedLargeNumeric () [virtual]
 Destructor

RTRProxyManagedNumeric Class Reference

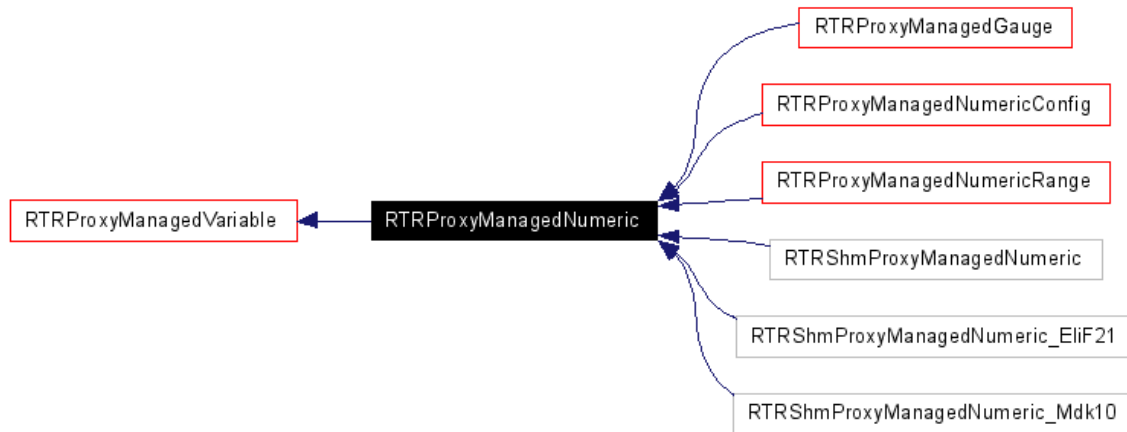
A cloned (proxy) representation of a Numeric variable. The base class for proxy numeric managed variables. Inherits from [RTRProxyManagedVariable](#) and provides services for accessing the current value of the variable.

```
#include <prxymn.h>
```

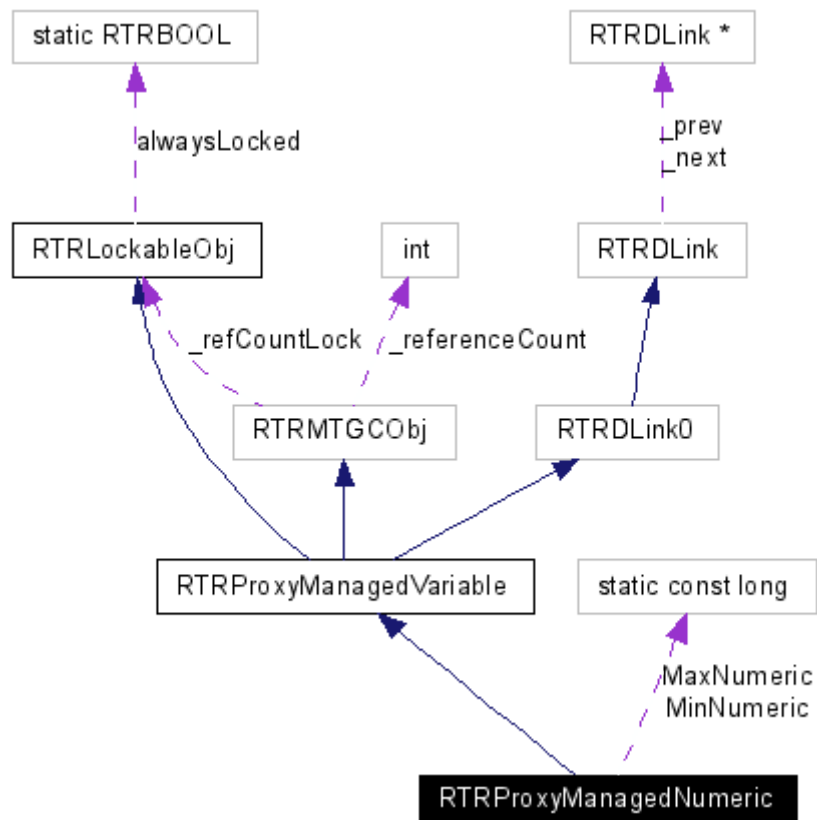
Inherits [RTRProxyManagedVariable](#).

Inherited by [RTRProxyManagedGauge](#), [RTRProxyManagedNumericConfig](#), [RTRProxyManagedNumericRange](#), [RTRShmProxyManagedNumeric](#), [RTRShmProxyManagedNumeric_Elif21](#), and [RTRShmProxyManagedNumeric_Mdk10](#).

Inheritance diagram for RTRProxyManagedNumeric:



Collaboration diagram for RTRProxyManagedNumeric:



Public Member Functions

- [RTRProxyManagedNumeric](#) ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
- virtual [~RTRProxyManagedNumeric](#) ()
- RTRBOOL [operator==](#) (long rhs) const
- long [value](#) () const
- virtual [RTRString toString](#) () const
- [operator long](#) () const
- [operator RTRProxyManagedGauge &](#) ()
- [operator const RTRProxyManagedGauge &](#) () const
- operator RTRProxyManagedGaugeConfig & ()
- [operator const RTRProxyManagedGaugeConfig &](#) () const
- operator RTRProxyManagedNumericConfig & ()
- operator const RTRProxyManagedNumericConfig & () const
- operator RTRProxyManagedNumericRange & ()
- operator const RTRProxyManagedNumericRange & () const

Static Public Attributes

- static const long **MinNumeric**
- static const long **MaxNumeric**

Detailed Description

A cloned (proxy) representation of a Numeric variable. The base class for proxy numeric managed variables. Inherits from [RTRProxyManagedVariable](#) and provides services for accessing the current value of the variable.

See Also:

[RTRProxyManagedVariableClient](#), [RTRProxyManagedCounter](#), [RTRProxyManagedGauge](#), [RTRProxyManagedString](#), [RTRProxyManagedBoolean](#), [RTRProxyManagedNumericRange](#), [RTRProxyManagedNumericConfig](#), [RTRProxyManagedStringConfig](#), [RTRProxyManagedBooleanConfig](#), [RTRProxyManagedGaugeConfig](#)

Constructor & Destructor Documentation

RTRProxyManagedNumeric::RTRProxyManagedNumeric ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
Constructor

virtual RTRProxyManagedNumeric::~~RTRProxyManagedNumeric () [**virtual**]
Destructor

RTRProxyManagedNumericConfig Class Reference

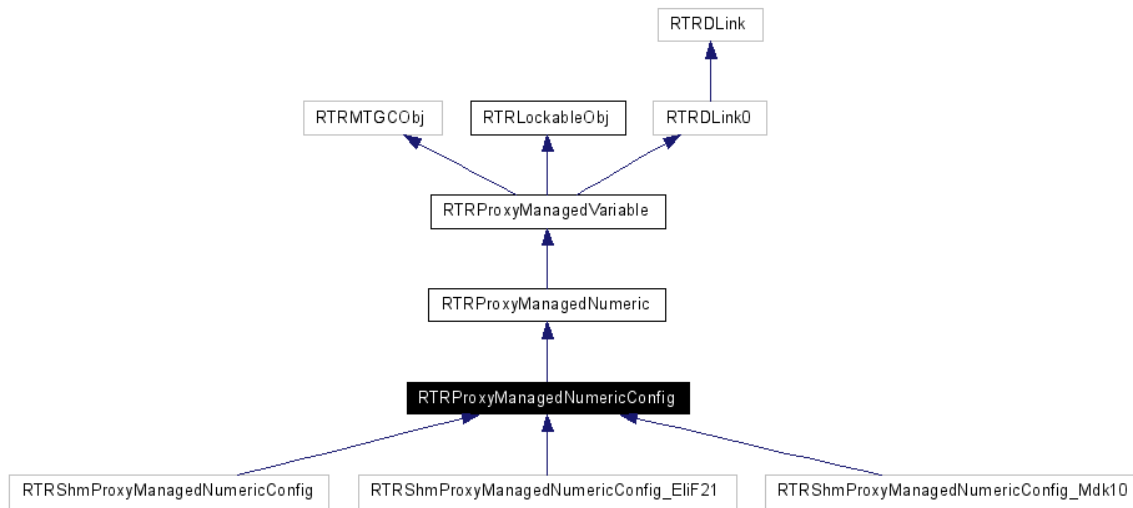
A cloned (proxy) representation of a NumericConfig variable. The base class for proxy numeric managed configuration variables. Inherits from [RTRProxyManagedNumeric](#) and provides additional services for accessing the stored and default values and for modifying (conditionally) the active value.

```
#include <prxymnc.h>
```

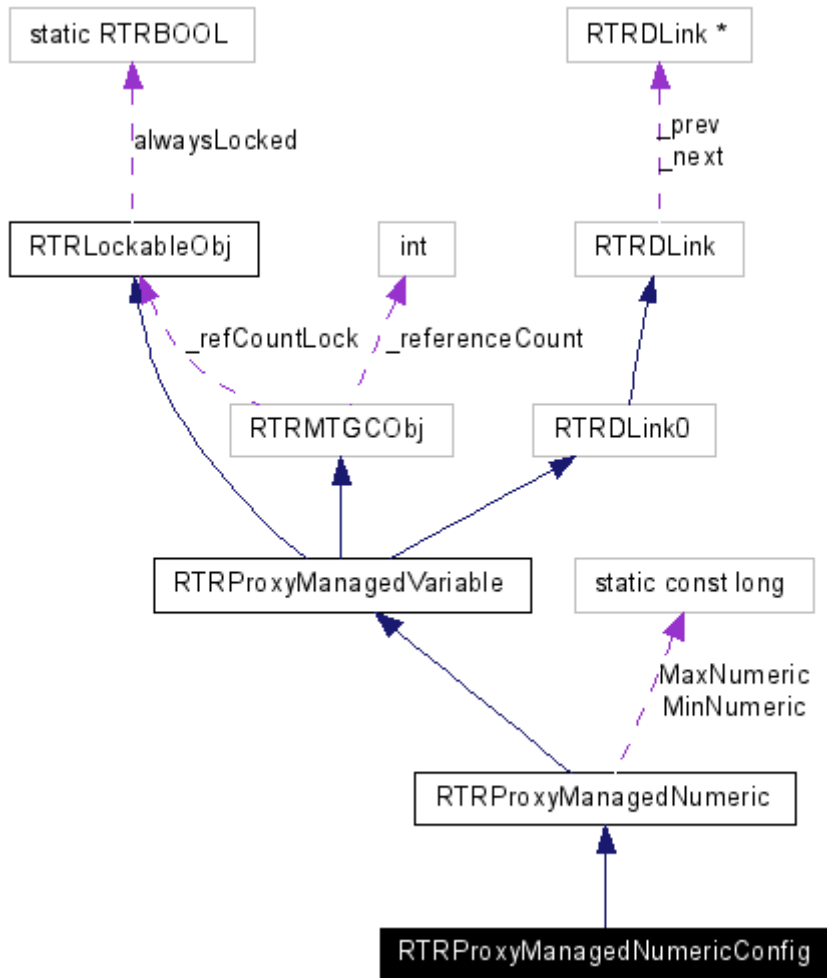
Inherits [RTRProxyManagedNumeric](#).

Inherited by RTRShmProxyManagedNumericConfig, RTRShmProxyManagedNumericConfig_Elif21, and RTRShmProxyManagedNumericConfig_Mdk10.

Inheritance diagram for RTRProxyManagedNumericConfig:



Collaboration diagram for RTRProxyManagedNumericConfig:



Public Member Functions

- [RTRProxyManagedNumericConfig](#) ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
- virtual [~RTRProxyManagedNumericConfig](#) ()
- long [activeValue](#) () const
- long [minValue](#) () const
- long [maxValue](#) () const
- long [storeValue](#) () const
- long [factoryDefault](#) () const
- RTRBOOL [modifyEnabled](#) () const
- RTRBOOL [hasStore](#) () const
- RTRBOOL [isStoreActive](#) () const
- RTRBOOL [isStoreClassConfig](#) () const
- RTRBOOL [isStoreInstanceConfig](#) () const

- [RTRProxyManagedNumericConfig](#) & [operator=](#) (long rhs)
- virtual void [set](#) (long newValue)=0

Detailed Description

A cloned (proxy) representation of a NumericConfig variable. The base class for proxy numeric managed configuration variables. Inherits from [RTRProxyManagedNumeric](#) and provides additional services for accessing the stored and default values and for modifying (conditionally) the active value.

If [modifyEnabled\(\)](#) is True, then managing applications are permitted to modify the active value. Note: The active values of configuration variables are not persistent.

See Also:

[RTRProxyManagedVariableClient](#), [RTRProxyManagedCounter](#), [RTRProxyManagedGauge](#), [RTRProxyManagedNumeric](#), [RTRProxyManagedString](#), [RTRProxyManagedBoolean](#), [RTRProxyManagedNumericRange](#), [RTRProxyManagedStringConfig](#), [RTRProxyManagedBooleanConfig](#), [RTRProxyManagedGaugeConfig](#)

Constructor & Destructor Documentation

RTRProxyManagedNumericConfig::RTRProxyManagedNumericConfig ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
 Constructor

virtual RTRProxyManagedNumericConfig::~RTRProxyManagedNumericConfig () [virtual]
 Destructor

RTRProxyManagedNumericRange Class Reference

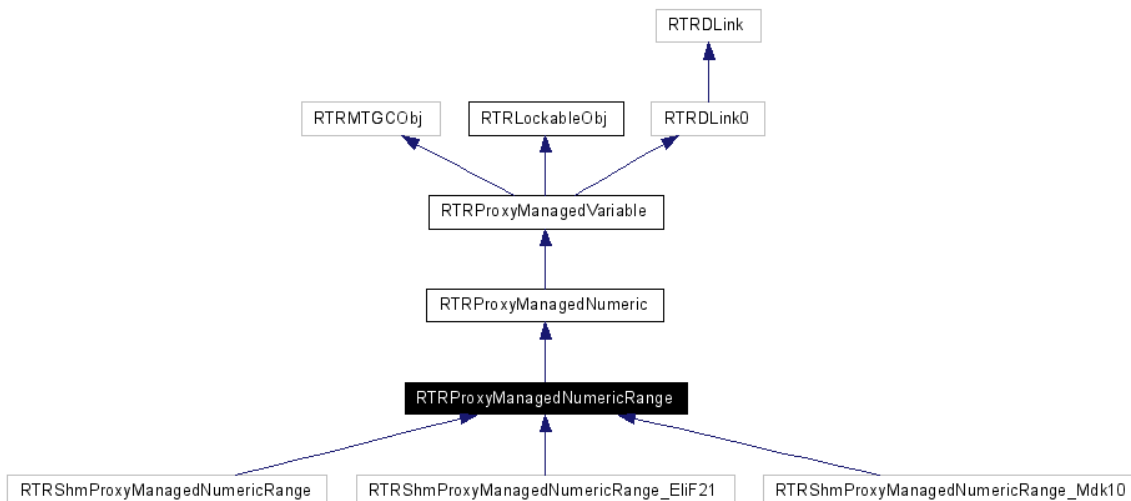
A cloned (proxy) representation of a NumericRange variable. The base class for proxy numeric range managed variables. Inherits from [RTRProxyManagedNumeric](#) and provides additional services for accessing and modifying (conditionally) the current value. It also provides a fixed range of values that the variable can assume.

```
#include <prxymnr.h>
```

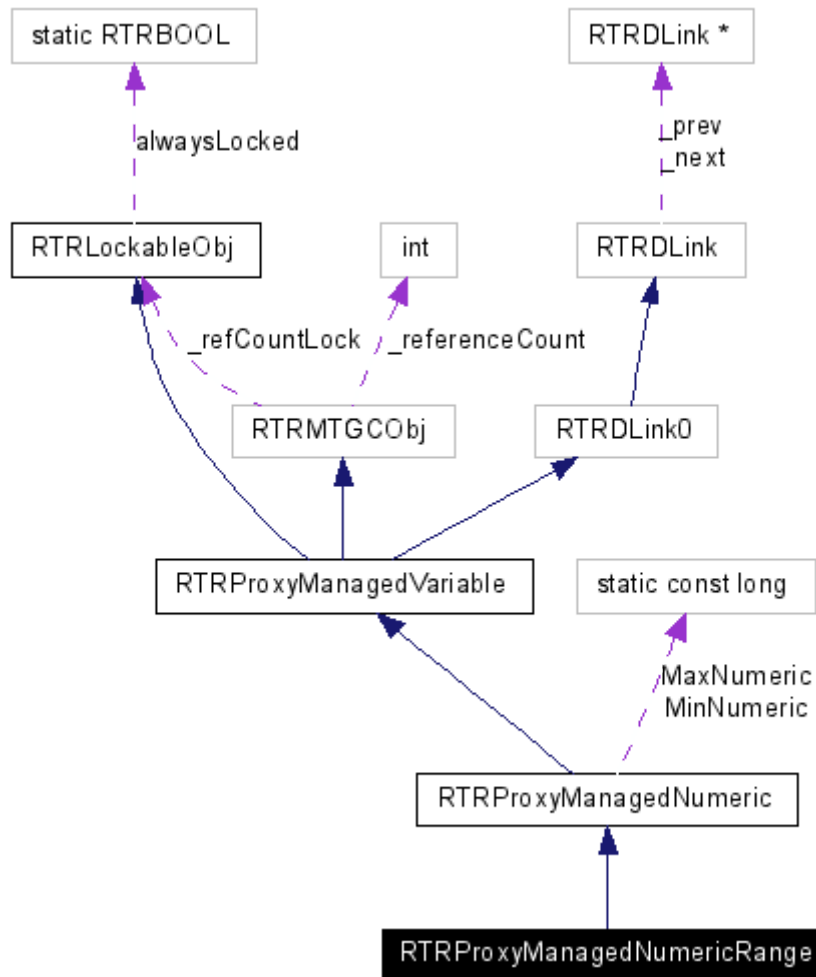
Inherits [RTRProxyManagedNumeric](#).

Inherited by [RTRShmProxyManagedNumericRange](#), [RTRShmProxyManagedNumericRange_Elif21](#), and [RTRShmProxyManagedNumericRange_Mdk10](#).

Inheritance diagram for RTRProxyManagedNumericRange:



Collaboration diagram for RTRProxyManagedNumericRange:



Public Member Functions

- [RTRProxyManagedNumericRange](#) ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
- virtual [~RTRProxyManagedNumericRange](#) ()
- long [minValue](#) () const
- long [maxValue](#) () const
- [RTRProxyManagedNumericRange](#) & [operator=](#) (long rhs)
- virtual void [set](#) (long newValue)=0

Detailed Description

A cloned (proxy) representation of a `NumericRange` variable. The base class for proxy numeric range managed variables. Inherits from [RTRProxyManagedNumeric](#) and provides additional services for accessing and modifying (conditionally) the current value. It also provides a fixed range of values that the variable can assume.

If `modifyEnabled()` is `True`, the managing applications are permitted to modify the current value. The new value must be within the specified range (`minValue()`/`maxValue()`).

See Also:

[RTRProxyManagedVariableClient](#), [RTRProxyManagedCounter](#), [RTRProxyManagedGauge](#), [RTRProxyManagedNumeric](#), [RTRProxyManagedBoolean](#), [RTRProxyManagedString](#), [RTRProxyManagedNumericConfig](#), [RTRProxyManagedStringConfig](#), [RTRProxyManagedBooleanConfig](#), [RTRProxyManagedGaugeConfig](#)

Constructor & Destructor Documentation

RTRProxyManagedNumericRange::RTRProxyManagedNumericRange ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)

Constructor

virtual RTRProxyManagedNumericRange::~~RTRProxyManagedNumericRange () [`virtual`]

Destructor

RTRProxyManagedObject Class Reference

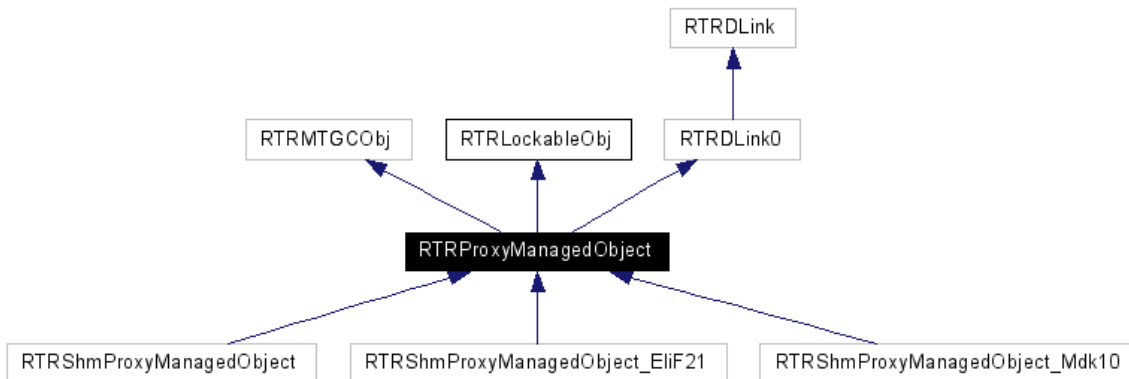
A cloned (proxy) representation of a managed object. `RTRProxyManagedObject` is an abstract base class representing application components which can be accessed and managed by external management entities. Management is accomplished by monitoring and possibly modifying variables made available by the application component to be managed.

```
#include <prxymo.h>
```

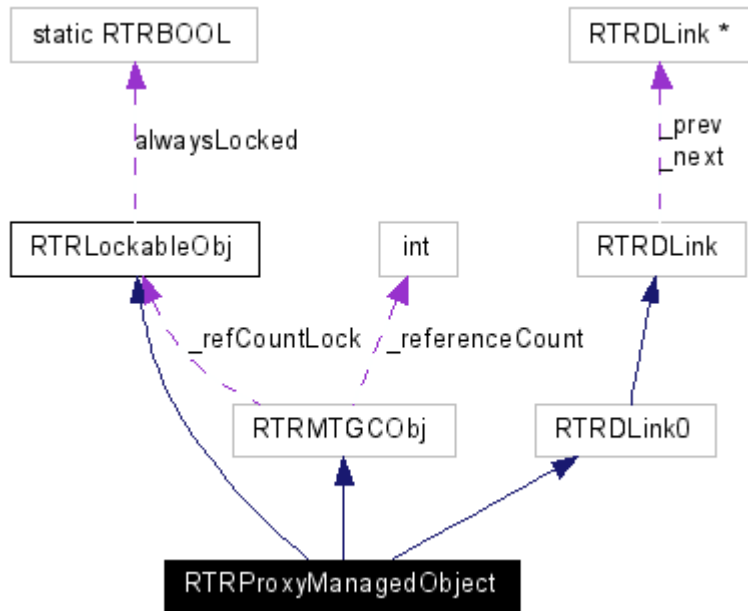
Inherits `RTRMTGCObj`, [RTRLockableObj](#), and `RTRDLink0`.

Inherited by `RTRShmProxyManagedObject`, `RTRShmProxyManagedObject_EliF21`, and `RTRShmProxyManagedObject_Mdk10`.

Inheritance diagram for `RTRProxyManagedObject`:



Collaboration diagram for `RTRProxyManagedObject`:



Public Types

- enum [PMOState](#) { Init, Normal, ManualRecovery, AutoRecovery, Dead, Invalid, LastValueDummy }

Public Member Functions

- [RTRProxyManagedObject](#) (const [RTRProxyManagedObjectServer](#) &, const [RTRProxyManagedObjectHandle](#) &)
- virtual [~RTRProxyManagedObject](#) ()
- const [RTRObjectId](#) & [instanceId](#) () const
- const [RTRString](#) & [name](#) () const
- const [RTRObjectId](#) & [classId](#) () const
- const [RTRString](#) & [description](#) () const
- [PMOState](#) [state](#) () const
- [PMOState](#) [previousState](#) () const
- const [RTRString](#) & [text](#) () const
- RTRBOOL [error](#) () const
- RTRBOOL [inSync](#) () const
- RTRBOOL [hasChild](#) (const [RTRString](#) &) const
- RTRBOOL [hasVariable](#) (const [RTRString](#) &) const
- [RTRProxyManagedVarHandleIterator](#) [variableHandles](#) () const
- [RTRProxyManagedObjectHandleIterator](#) [childHandles](#) () const
- virtual RTRObjRef< [RTRProxyManagedObject](#) > [childByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedVariablePtr [variableByName](#) (const [RTRString](#) &name) const

- RTRProxyManagedBooleanPtr [booleanByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedBooleanConfigPtr [booleanConfigByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedCounterPtr [counterByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedGaugePtr [gaugeByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedGaugeConfigPtr [gaugeConfigByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedNumericPtr [numericByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedNumericConfigPtr [numericConfigByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedNumericRangePtr [numericRangeByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedStringPtr [stringByName](#) (const [RTRString](#) &name) const
- RTRProxyManagedStringConfigPtr [stringConfigByName](#) (const [RTRString](#) &name) const
- RTRBOOL [hasClient](#) ([RTRProxyManagedObjectClient](#) &) const
- void [addClient](#) ([RTRProxyManagedObjectClient](#) &)
- void [dropClient](#) ([RTRProxyManagedObjectClient](#) &)
- virtual void [lock](#) ()
- virtual void [unlock](#) ()
- virtual RTRBOOL [locked](#) () const

Friends

- class [RTRProxyManagedObjectServer](#)
- class [RTRProxyManagedVariable](#)

Detailed Description

A cloned (proxy) representation of a managed object. RTRProxyManagedObject is an abstract base class representing application components which can be accessed and managed by external management entities. Management is accomplished by monitoring and possibly modifying variables made available by the application component to be managed.

Managed applications are perceived by management entities as a collection of proxy managed objects. These objects provide some number of variables of interest. The exact variables provided by an instance of managed object depends on the type of that object. All variables conform to one of a limited number of types; all variable types are specializations of [RTRProxyManagedVariable](#).

The managed objects within an application have relationships with other managed objects, i.e. objects may refer to other objects. These relationships form one or more trees (whose nodes are objects) and can be of interest to management entities. Managed objects contained by other objects are children. Managed objects not contained by other objects are the roots of object trees. Given the set of roots in an application, or system, all other objects can be reached by traversing the trees defined by those roots.

The instance tree represents the composition of an application as represented to external management entities, and provides the means for those entities to access and possibly modify variables within the application.

Managed objects have an instance identifier which uniquely identifies that object. Managed objects also have a class identifier which identifies the type (semantics) of that object. In principle, all managed objects with a given class identifier (type) will provide the same set of variables.

Application components which wish to be notified of changes in the composition of a particular managed object may register with that object in order to receive object level events. To do so, they must be descendants of [RTRProxyManagedObjectClient](#).

See Also:

[RTRProxyManagedObjectClient](#)

Member Enumeration Documentation

enum [RTRProxyManagedObject::PMOState](#)
The state attribute values.

Constructor & Destructor Documentation

[RTRProxyManagedObject::RTRProxyManagedObject](#) (const [RTRProxyManagedObjectServer](#) &, const [RTRProxyManagedObjectHandle](#) &)
Constructor

[virtual RTRProxyManagedObject::~RTRProxyManagedObject](#) () [virtual]
Destructure

RTRProxyManagedObjectClassDirectory Class Reference

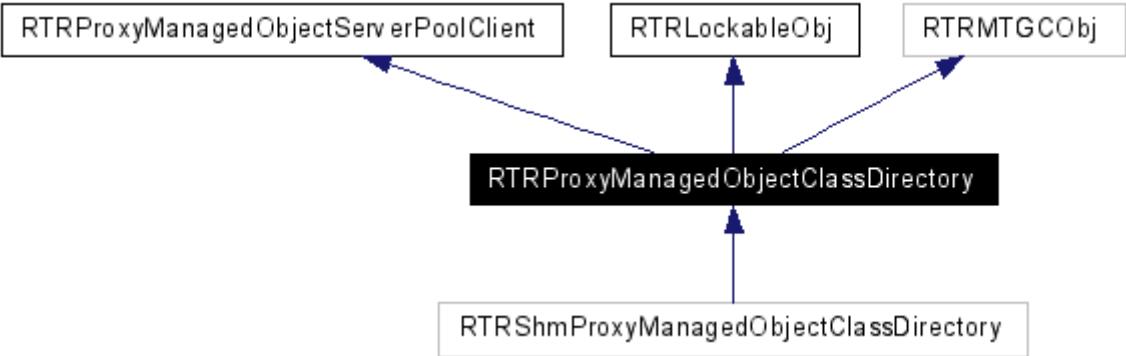
RTRProxyManagedObjectClassDirectory is a directory of all object handles for all managed objects of a particular type as published by a pool of [RTRProxyManagedObjectServer](#). The handles can be used to retrieve instances of [RTRProxyManagedObject](#) from a [RTRProxyManagedObjectServer](#). The pool of servers which contribute to a directory is specified when the directory is constructed. The directory will dynamically adjust its contents according to changes in the server pool and changes in the set of objects published by the servers.

```
#include <pmocd.h>
```

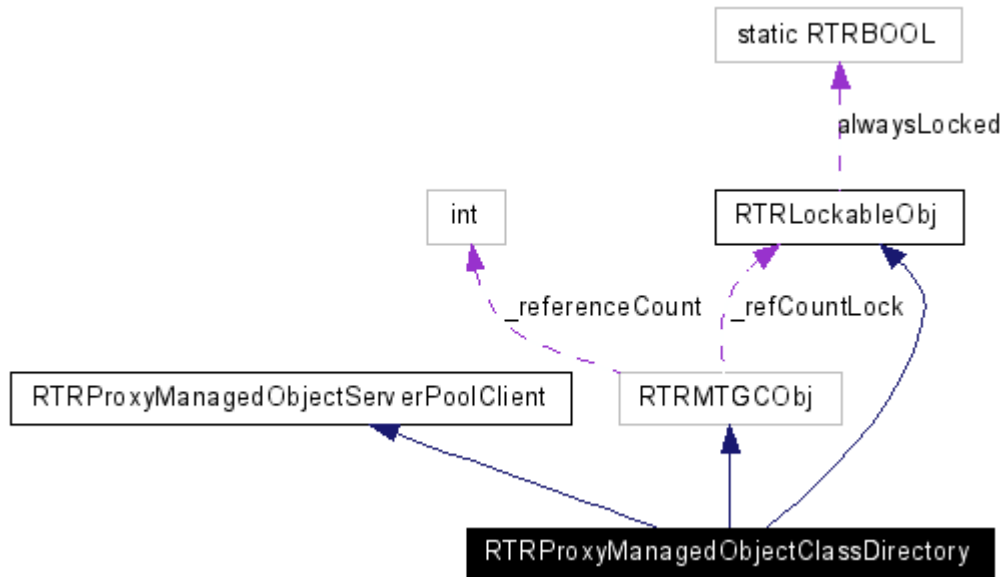
Inherits [RTRProxyManagedObjectServerPoolClient](#), [RTRLockableObj](#), and [RTRMTGCObj](#).

Inherited by [RTRShmProxyManagedObjectClassDirectory](#).

Inheritance diagram for RTRProxyManagedObjectClassDirectory:



Collaboration diagram for RTRProxyManagedObjectClassDirectory:



Public Member Functions

- [RTRProxyManagedObjectClassDirectory](#) ([RTRProxyManagedObjectServerPool](#) &pool, const [RTRObjectId](#) &filter)
- virtual [~RTRProxyManagedObjectClassDirectory](#) ()
- const [RTRObjectId](#) & [classFilter](#) () const
- const [RTRProxyManagedObjectServerPool](#) & [serverPool](#) () const
- RTRBOOL [hasHandle](#) (const [RTRObjectId](#) &) const
- RTRBOOL [hasHandle](#) (const [RTRProxyManagedObjectHandle](#) &) const
- const [RTRProxyManagedObjectHandle](#) * [handle](#) (const [RTRObjectId](#) &iid) const
- [RTRProxyManagedObjectHandleIterator](#) [handles](#) () const
- virtual void [processProxyManagedObjectServerAdded](#) ([RTRProxyManagedObjectServerPool](#) &, [RTRProxyManagedObjectServer](#) &)
- RTRBOOL [hasClient](#) (const [RTRProxyManagedObjectClassDirectoryClient](#) &) const
- void [addClient](#) (const [RTRProxyManagedObjectClassDirectoryClient](#) &client)
- void [dropClient](#) (const [RTRProxyManagedObjectClassDirectoryClient](#) &client)

Detailed Description

[RTRProxyManagedObjectClassDirectory](#) is a directory of all object handles for all managed objects of a particular type as published by a pool of [RTRProxyManagedObjectServer](#). The handles can be used to retrieve instances of [RTRProxyManagedObject](#) from a [RTRProxyManagedObjectServer](#). The pool of servers which contribute to a directory is specified when the directory is constructed. The directory will dynamically adjust its contents according to changes in the server pool and changes in the set of objects published by the servers.

Application components which want to monitor events associated with one or more instances of [RTRProxyManagedObjectClassDirectory](#) must be descendants of [RTRProxyManagedObjectClassDirectoryClient](#).

See Also:

[RTRProxyManagedObjectClassDirectoryClient](#)

Constructor & Destructor Documentation

RTRProxyManagedObjectClassDirectory::RTRProxyManagedObjectClassDirectory ([RTRProxyManagedObjectServerPool](#) & *pool*, const [RTRObjectId](#) & *filter*)

Constructor

virtual RTRProxyManagedObjectClassDirectory::~RTRProxyManagedObjectClassDirectory () [virtual]

Destructor

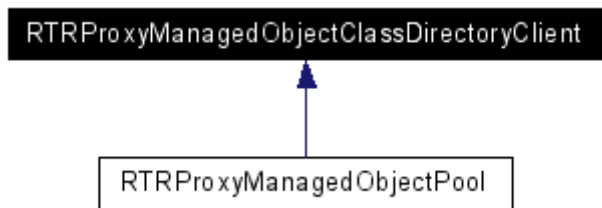
RTRProxyManagedObjectClassDirectoryClient Class Reference

RTRProxyManagedObjectClassDirectoryClient is the abstract base class for application components which can register to receive events from one or more instances of [RTRProxyManagedObjectClassDirectory](#).

```
#include <pmocdc.h>
```

Inherited by [RTRProxyManagedObjectPool](#).

Inheritance diagram for RTRProxyManagedObjectClassDirectoryClient:



Public Member Functions

- **RTRProxyManagedObjectClassDirectoryClient** ()
- virtual **~RTRProxyManagedObjectClassDirectoryClient** ()
- virtual void **processDirectoryHandleAdded** ([RTRProxyManagedObjectClassDirectory](#) &, [RTRProxyManagedObjectServer](#) &, const [RTRProxyManagedObjectHandle](#) &)=0
- virtual void **processDirectoryHandleRemoved** ([RTRProxyManagedObjectClassDirectory](#) &, [RTRProxyManagedObjectServer](#) &, const [RTRProxyManagedObjectHandle](#) &)=0

Detailed Description

RTRProxyManagedObjectClassDirectoryClient is the abstract base class for application components which can register to receive events from one or more instances of [RTRProxyManagedObjectClassDirectory](#).

See Also:

[RTRProxyManagedObjectClassDirectory](#)

Constructor & Destructor Documentation

RTRProxyManagedObjectClassDirectoryClient::RTRProxyManagedObjectClassDirectoryClient ()

Constructor

virtual RTRProxyManagedObjectClassDirectoryClient::~RTRProxyManagedObjectClassDirectoryClient () [virtual]

Destructor

RTRProxyManagedObjectClient Class Reference

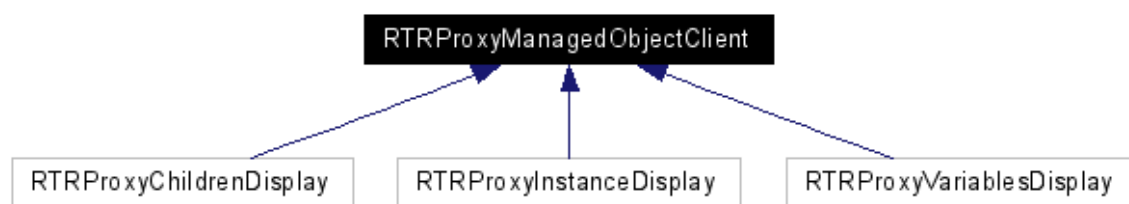
The base class for components which can register with a proxy managed object to receive change events from that proxy managed object. The notifications are grouped into five categories:

1. proxy managed object state changes,
2. the managed object has been deleted by the producer,
3. the state attribute has changed,
4. a child managed object has been added/removed, and
5. a contained managed variable has been added/removed.

```
#include <prxymoc.h>
```

Inherited by RTRProxyChildrenDisplay, RTRProxyInstanceDisplay, and RTRProxyVariablesDisplay.

Inheritance diagram for RTRProxyManagedObjectClient:



Public Member Functions

- virtual [~RTRProxyManagedObjectClient](#) ()
- virtual void [processProxyManagedObjectError](#) (const [RTRProxyManagedObject](#) &)=0
- virtual void [processProxyManagedObjectSync](#) (const [RTRProxyManagedObject](#) &)=0
- virtual void [processProxyManagedObjectDeleted](#) (const [RTRProxyManagedObject](#) &)=0
- virtual void [processProxyManagedObjectInfo](#) (const [RTRProxyManagedObject](#) &)=0
- virtual void [processProxyManagedObjectInService](#) (const [RTRProxyManagedObject](#) &)=0
- virtual void [processProxyManagedObjectRecovering](#) (const [RTRProxyManagedObject](#) &)=0
- virtual void [processProxyManagedObjectWaiting](#) (const [RTRProxyManagedObject](#) &)=0
- virtual void [processProxyManagedObjectDead](#) (const [RTRProxyManagedObject](#) &)=0
- virtual void [processProxyManagedObjectChildAdded](#) (const [RTRProxyManagedObject](#) &, const [RTRProxyManagedObjectHandle](#) &)=0
- virtual void [processProxyManagedObjectChildRemoved](#) (const [RTRProxyManagedObject](#) &, const [RTRProxyManagedObjectHandle](#) &)=0
- virtual void [processProxyManagedObjectVariableAdded](#) (const [RTRProxyManagedObject](#) &, const [RTRProxyManagedVariableHandle](#) &)=0
- virtual void [processProxyManagedObjectVariableRemoved](#) (const [RTRProxyManagedObject](#) &, const [RTRProxyManagedVariableHandle](#) &)=0

Detailed Description

The base class for components which can register with a proxy managed object to receive change events from that proxy managed object. The notifications are grouped into five categories:

1. proxy managed object state changes,
2. the managed object has been deleted by the producer,
3. the state attribute has changed,
4. a child managed object has been added/removed, and
5. a contained managed variable has been added/removed.

See Also:

[RTRProxyManagedObject](#)

Constructor & Destructor Documentation

virtual RTRProxyManagedObjectClient::~~RTRProxyManagedObjectClient () [virtual]
 Destructor

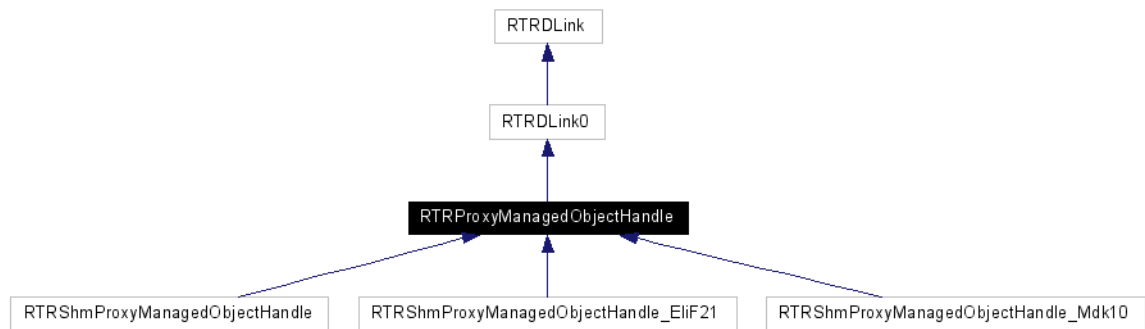
RTRProxyManagedObjectHandle Class Reference

A RTRProxyManagedObjectHandle uniquely identifies a Proxy Managed Object. The handle is used to request a clone (proxy) of a particular managed object.

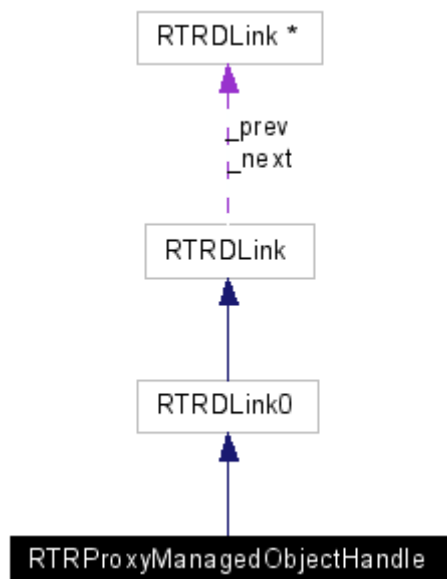
```
#include <prxyh.h>
Inherits RTRDLink0.
```

Inherited by RTRShmProxyManagedObjectHandle, RTRShmProxyManagedObjectHandle_EliF21, and RTRShmProxyManagedObjectHandle_Mdk10.

Inheritance diagram for RTRProxyManagedObjectHandle:



Collaboration diagram for RTRProxyManagedObjectHandle:



Public Member Functions

- [RTRProxyManagedObjectHandle](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &name)
- [RTRProxyManagedObjectHandle](#) (const [RTRProxyManagedObjectHandle](#) &)
- virtual [~RTRProxyManagedObjectHandle](#) ()
- const [RTRObjectId](#) & [classId](#) () const
- const [RTRObjectId](#) & [instanceId](#) () const
- const [RTRString](#) & [name](#) () const

Friends

- std::ostream & [operator<<](#) (std::ostream &, const [RTRProxyManagedObjectHandle](#) &)

Detailed Description

A [RTRProxyManagedObjectHandle](#) uniquely identifies a Proxy Managed Object. The handle is used to request a clone (proxy) of a particular managed object.

See Also:

[RTRProxyManagedVariableHandle](#), [RTRProxyManagedObjectHandleIterator](#), [RTRProxyManagedVarHandleIterator](#)

Constructor & Destructor Documentation

RTRProxyManagedObjectHandle::RTRProxyManagedObjectHandle (const [RTRObjectId](#) & *classId*, const [RTRObjectId](#) & *instanceId*, const [RTRString](#) & *name*)
 Constructor

RTRProxyManagedObjectHandle::RTRProxyManagedObjectHandle (const [RTRProxyManagedObjectHandle](#) &)
 Copy constructor

virtual RTRProxyManagedObjectHandle::~~RTRProxyManagedObjectHandle () [*inline, virtual*]
 Destructor

RTRProxyManagedObjectHandleIterator Class Reference

A RTRProxyManagedObjectHandleIterator is used to sequentially traverse a set of Proxy Managed Object Handles.

```
#include <prxy.h>
```

Public Member Functions

- [RTRProxyManagedObjectHandleIterator](#) (const RTRDLinkList< [RTRProxyManagedObjectHandle](#), RTRDLink0 > &)
- int [count](#) () const
- RTRBOOL [off](#) () const
- RTRBOOL [empty](#) () const
- [RTRProxyManagedObjectHandle](#) & [item](#) () const
- void [start](#) ()
- void [finish](#) ()
- void [forth](#) ()
- void [back](#) ()

Detailed Description

A RTRProxyManagedObjectHandleIterator is used to sequentially traverse a set of Proxy Managed Object Handles.

See Also:

[RTRProxyManagedObjectHandle](#), [RTRProxyManagedVariableHandle](#), [RTRProxyManagedVarHandleIterator](#)

Constructor & Destructor Documentation

RTRProxyManagedObjectHandleIterator::RTRProxyManagedObjectHandleIterator (const RTRDLinkList<
[RTRProxyManagedObjectHandle](#), RTRDLink0 > &)
Constructor

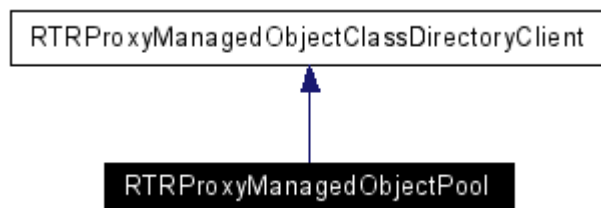
RTRProxyManagedObjectPool Class Reference

RTRProxyManagedObjectPool is a pool of objects matching the contents of a directory provided on the constructor. The directory in turn matches the contents of a pool of object servers.

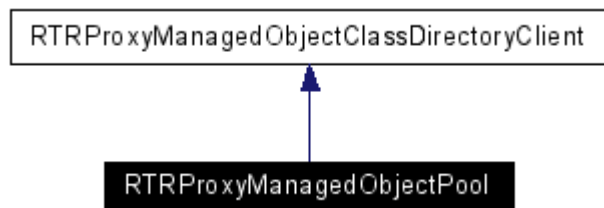
```
#include <pmop.h>
```

Inherits [RTRProxyManagedObjectClassDirectoryClient](#).

Inheritance diagram for RTRProxyManagedObjectPool:



Collaboration diagram for RTRProxyManagedObjectPool:



Public Member Functions

- [RTRProxyManagedObjectPool](#) ([RTRProxyManagedObjectClassDirectory](#) &)
- virtual [~RTRProxyManagedObjectPool](#) ()
- RTRBOOL [hasObject](#) (const [RTRObjectld](#) &) const
- [RTRProxyManagedObjectClassDirectory](#) & **directory** () const
- RTRProxyManagedObjectPtr [object](#) (const [RTRObjectld](#) &) const
- RTRLinkedListCursor< RTRProxyManagedObjectPtr > [objects](#) () const
- virtual void [processDirectoryHandleAdded](#) ([RTRProxyManagedObjectClassDirectory](#) &, [RTRProxyManagedObjectServer](#) &, const [RTRProxyManagedObjectHandle](#) &)
- virtual void [processDirectoryHandleRemoved](#) ([RTRProxyManagedObjectClassDirectory](#) &, [RTRProxyManagedObjectServer](#) &, const [RTRProxyManagedObjectHandle](#) &)
- RTRBOOL [hasClient](#) (const [RTRProxyManagedObjectPoolClient](#) &) const
- void [addClient](#) (const [RTRProxyManagedObjectPoolClient](#) &client)
- void [dropClient](#) (const [RTRProxyManagedObjectPoolClient](#) &client)

Detailed Description

`RTRProxyManagedObjectPool` is a pool of objects matching the contents of a directory provided on the constructor. The directory in turn matches the contents of a pool of object servers.

See Also:

[RTRProxyManagedObjectClassDirectory](#)

Constructor & Destructor Documentation

`RTRProxyManagedObjectPool::RTRProxyManagedObjectPool` ([RTRProxyManagedObjectClassDirectory](#) &)
Constructor

virtual `RTRProxyManagedObjectPool::~~RTRProxyManagedObjectPool` () [virtual]
Destructor

RTRProxyManagedObjectPoolClient Class Reference

`RTRProxyManagedObjectPoolClient` is the abstract base class for application components which wish to register with one or more instances of [RTRProxyManagedObjectPool](#) in order to be notified when objects are added to or removed from a pool.

```
#include <pmopc.h>
```

Public Member Functions

- [RTRProxyManagedObjectPoolClient](#) ()
- virtual [~RTRProxyManagedObjectPoolClient](#) ()
- virtual void [processProxyManagedObjectAdded](#) ([RTRProxyManagedObjectPool](#) &, [RTRProxyManagedObject](#) &)=0
- virtual void [processProxyManagedObjectRemoved](#) ([RTRProxyManagedObjectPool](#) &, [RTRProxyManagedObject](#) &)=0

Detailed Description

RTRProxyManagedObjectPoolClient is the abstract base class for application components which wish to register with one or more instances of [RTRProxyManagedObjectPool](#) in order to be notified when objects are added to or removed from a pool.

See Also:

[RTRProxyManagedObjectPool](#), [RTRProxyManagedObject](#)

Constructor & Destructor Documentation

RTRProxyManagedObjectPoolClient::RTRProxyManagedObjectPoolClient ()
 Constructor

virtual RTRProxyManagedObjectPoolClient::~~RTRProxyManagedObjectPoolClient () [virtual]
 Destructor

RTRProxyManagedObjectServer Class Reference

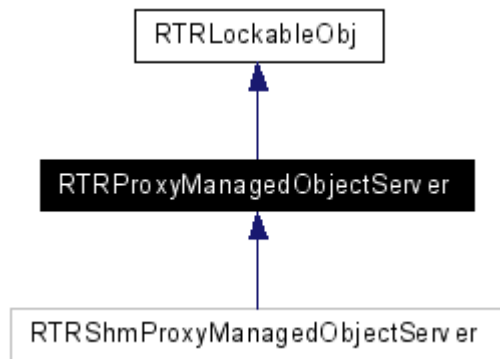
Provides access to the managed objects of a particular RTRManagedObjectServer. The set of available root proxy managed objects is also maintained.

```
#include <prxymos.h>
```

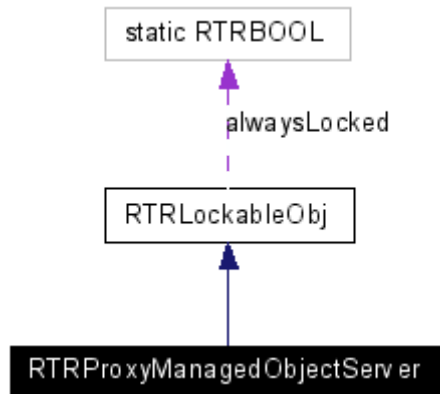
Inherits [RTRLockableObj](#).

Inherited by RTRShmProxyManagedObjectServer.

Inheritance diagram for RTRProxyManagedObjectServer:



Collaboration diagram for RTRProxyManagedObjectServer:



Public Member Functions

- [RTRProxyManagedObjectServer \(\)](#)
- virtual [~RTRProxyManagedObjectServer \(\)](#)
- const [RTRString](#) & [text](#) () const
- RTRBOOL [error](#) () const
- RTRBOOL [inSync](#) () const
- [RTRProxyManagedObjectHandleIterator](#) [roots](#) () const
- virtual [RTRProxyManagedObjectPtr](#) [object](#) (const [RTRProxyManagedObjectHandle](#) &id) const
- RTRBOOL [hasClient](#) (const [RTRProxyManagedObjectServerClient](#) &) const
- void [addClient](#) (const [RTRProxyManagedObjectServerClient](#) &client)
- void [dropClient](#) (const [RTRProxyManagedObjectServerClient](#) &client)

Friends

- class [RTRProxyManagedObject](#)

Detailed Description

Provides access to the managed objects of a particular `RTRManagedObjectServer`. The set of available root proxy managed objects is also maintained.

See Also:

[RTRProxyManagedObjectServerClient](#)

Constructor & Destructor Documentation

`RTRProxyManagedObjectServer::RTRProxyManagedObjectServer ()`
 Constructor

`virtual RTRProxyManagedObjectServer::~~RTRProxyManagedObjectServer () [virtual]`
 Destructor

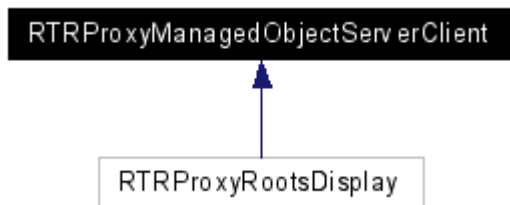
RTRProxyManagedObjectServerClient Class Reference

RTRProxyManagedObjectServerClient is the abstract base class for application components which wish to register with one or more instances of [RTRProxyManagedObjectServer](#) in order to be notified when root managed objects are added to or removed from a server.

```
#include <pmosc.h>
```

Inherited by RTRProxyRootsDisplay.

Inheritance diagram for RTRProxyManagedObjectServerClient:



Public Member Functions

- virtual [~RTRProxyManagedObjectServerClient](#) ()
- virtual void [processObjectServerError](#) ([RTRProxyManagedObjectServer](#) &)
- virtual void [processObjectServerSync](#) ([RTRProxyManagedObjectServer](#) &)
- virtual void [processObjectServerRootAdded](#) ([RTRProxyManagedObjectServer](#) &, const [RTRProxyManagedObjectHandle](#) &)
- virtual void [processObjectServerRootRemoved](#) ([RTRProxyManagedObjectServer](#) &, const [RTRProxyManagedObjectHandle](#) &)

Detailed Description

RTRProxyManagedObjectServerClient is the abstract base class for application components which wish to register with one or more instances of [RTRProxyManagedObjectServer](#) in order to be notified when root managed objects are added to or removed from a server.

See Also:

[RTRProxyManagedObjectServer](#)

Constructor & Destructor Documentation

virtual RTRProxyManagedObjectServerClient::~RTRProxyManagedObjectServerClient () [virtual]
Destructor

RTRProxyManagedObjectServerPool Class Reference

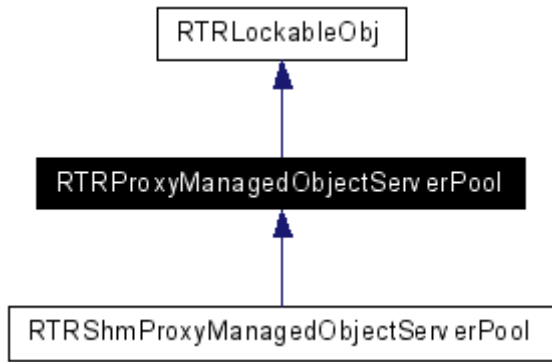
RTRProxyManagedObjectServerPool acts as a factory for instances of the class [RTRProxyManagedObjectServer](#).

```
#include <pmosp.h>
```

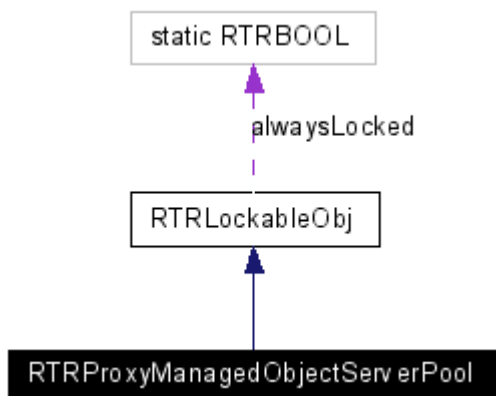
Inherits [RTRLockableObj](#).

Inherited by [RTRShmProxyManagedObjectServerPool](#).

Inheritance diagram for RTRProxyManagedObjectServerPool:



Collaboration diagram for RTRProxyManagedObjectServerPool:



Public Member Functions

- [RTRProxyManagedObjectServerPool \(\)](#)
- virtual [~RTRProxyManagedObjectServerPool \(\)](#)
- RTRLinkedListCursor< [RTRProxyManagedObjectServer](#) > [servers](#) () const
- RTRBOOL [hasClient](#) (const [RTRProxyManagedObjectServerPoolClient](#) &) const
- void [addClient](#) (const [RTRProxyManagedObjectServerPoolClient](#) &client)
- void [dropClient](#) (const [RTRProxyManagedObjectServerPoolClient](#) &client)

Detailed Description

RTRProxyManagedObjectServerPool acts as a factory for instances of the class [RTRProxyManagedObjectServer](#).

See Also:

[RTRProxyManagedObjectServer](#)

Constructor & Destructor Documentation

RTRProxyManagedObjectServerPool::RTRProxyManagedObjectServerPool ()
 Constructor

virtual RTRProxyManagedObjectServerPool::~RTRProxyManagedObjectServerPool () [virtual]
 Destructor

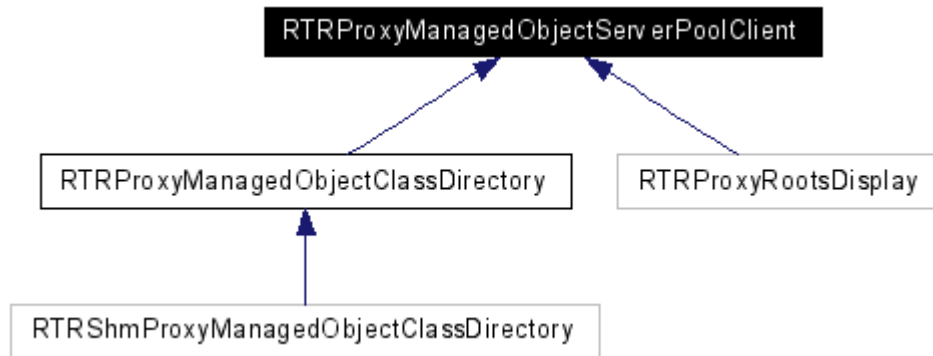
RTRProxyManagedObjectServerPoolClient Class Reference

The base class for components which can register with a proxy managed object server pool to receive change events from that server pool. The notifications are grouped into a single category. (1) A proxy managed object server has been added/removed from the pool.

```
#include <pmospc.h>
```

Inherited by [RTRProxyManagedObjectClassDirectory](#) and [RTRProxyRootsDisplay](#).

Inheritance diagram for RTRProxyManagedObjectServerPoolClient:



Public Member Functions

- [RTRProxyManagedObjectServerPoolClient \(\)](#)
- [virtual ~RTRProxyManagedObjectServerPoolClient \(\)](#)
- [virtual void processProxyManagedObjectServerAdded \(RTRProxyManagedObjectServerPool &, RTRProxyManagedObjectServer &\)](#)
- [virtual void processProxyManagedObjectServerRemoved \(RTRProxyManagedObjectServerPool &, RTRProxyManagedObjectServer &\)](#)

Detailed Description

The base class for components which can register with a proxy managed object server pool to receive change events from that server pool. The notifications are grouped into a single category. (1) A proxy managed object server has been added/removed from the pool.

See Also:

[RTRProxyManagedObjectServerPool](#)

Constructor & Destructor Documentation

RTRProxyManagedObjectServerPoolClient::RTRProxyManagedObjectServerPoolClient ()
 Constructor

virtual RTRProxyManagedObjectServerPoolClient::~RTRProxyManagedObjectServerPoolClient () [virtual]
 Destructor

RTRProxyManagedString Class Reference

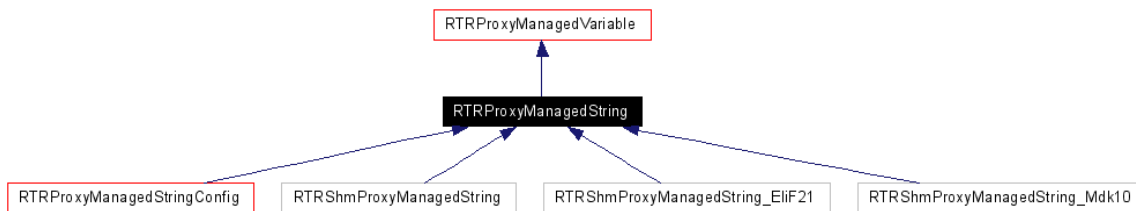
A cloned (proxy) representation of a String variable. The base class for proxy string managed variables. Inherits from [RTRProxyManagedVariable](#) and provides additional services for accessing and modifying (conditionally) the current value.

```
#include <prxyms.h>
```

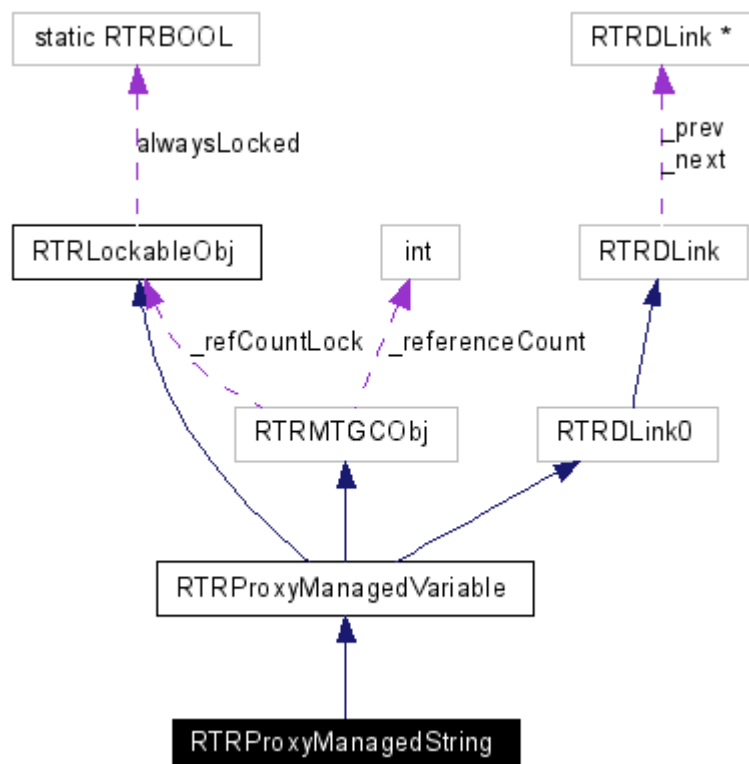
Inherits [RTRProxyManagedVariable.](#)

Inherited by [RTRProxyManagedStringConfig](#), RTRShmProxyManagedString, RTRShmProxyManagedString_EliF21, and RTRShmProxyManagedString_Mdk10.

Inheritance diagram for RTRProxyManagedString:



Collaboration diagram for RTRProxyManagedString:



Public Member Functions

- [RTRProxyManagedString](#) ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
- virtual [~RTRProxyManagedString](#) ()
- const [RTRString](#) & [value](#) () const

- virtual [RTRString toString](#) () const
- [operator const char *](#) () const
- operator RTRProxyManagedStringConfig & ()
- operator const RTRProxyManagedStringConfig & () const
- RTRBOOL [operator==](#) (const char *) const
- RTRBOOL [modifyEnabled](#) () const
- [RTRProxyManagedString](#) & [operator=](#) (const [RTRString](#) &rhs)
- [RTRProxyManagedString](#) & [operator=](#) (const char *rhs)
- virtual void [set](#) (const [RTRString](#) &newValue)=0
- virtual void [set](#) (const char *newValue)=0

Detailed Description

A cloned (proxy) representation of a String variable. The base class for proxy string managed variables. Inherits from [RTRProxyManagedVariable](#) and provides additional services for accessing and modifying (conditionally) the current value.

If [modifyEnabled\(\)](#) is True, then managing applications are permitted to modify the current value.

See Also:

[RTRProxyManagedVariableClient](#), [RTRProxyManagedCounter](#), [RTRProxyManagedGauge](#), [RTRProxyManagedNumeric](#), [RTRProxyManagedBoolean](#), [RTRProxyManagedNumericRange](#), [RTRProxyManagedNumericConfig](#), [RTRProxyManagedStringConfig](#), [RTRProxyManagedBooleanConfig](#), [RTRProxyManagedGaugeConfig](#)

Constructor & Destructor Documentation

RTRProxyManagedString::RTRProxyManagedString ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
Constructor

virtual RTRProxyManagedString::~RTRProxyManagedString () [virtual]
Destructor

RTRProxyManagedStringConfig Class Reference

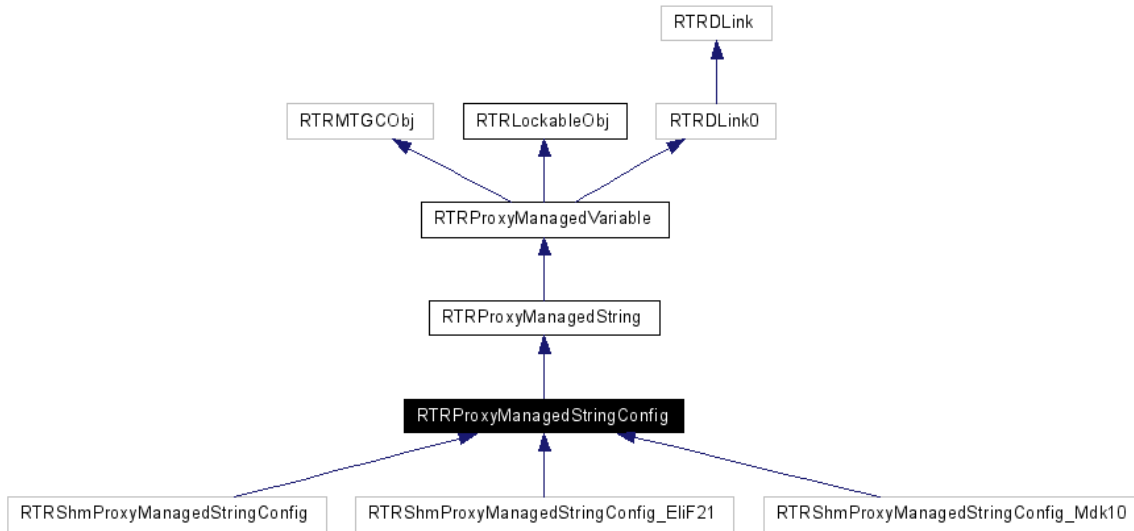
A cloned (proxy) representation of a StringConfig variable. The base class for proxy string managed configuration variables. Inherits from [RTRProxyManagedString](#) and provides additional services for accessing the stored and default values.

```
#include <prxymsc.h>
```

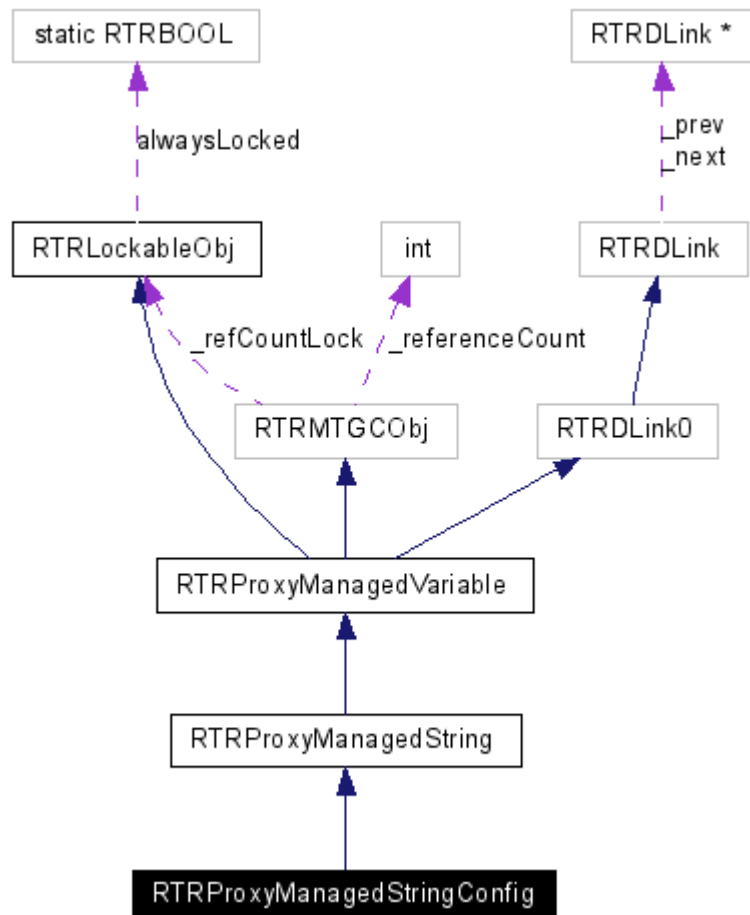
Inherits [RTRProxyManagedString](#).

Inherited by [RTRShmProxyManagedStringConfig](#), [RTRShmProxyManagedStringConfig_EliF21](#), and [RTRShmProxyManagedStringConfig_Mdk10](#).

Inheritance diagram for RTRProxyManagedStringConfig:



Collaboration diagram for RTRProxyManagedStringConfig:



Public Member Functions

- [RTRProxyManagedStringConfig](#) ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
- virtual [~RTRProxyManagedStringConfig](#) ()
- const [RTRString](#) & [activeValue](#) () const
- const [RTRString](#) & [storeValue](#) () const
- const [RTRString](#) & [factoryDefault](#) () const
- RTRBOOL [hasStore](#) () const
- RTRBOOL [isStoreActive](#) () const
- RTRBOOL [isStoreClassConfig](#) () const
- RTRBOOL [isStoreInstanceConfig](#) () const
- [RTRProxyManagedStringConfig](#) & [operator=](#) (const [RTRString](#) &rhs)
- [RTRProxyManagedStringConfig](#) & [operator=](#) (const char *rhs)

Detailed Description

A cloned (proxy) representation of a StringConfig variable. The base class for proxy string managed configuration variables. Inherits from [RTRProxyManagedString](#) and provides additional services for accessing the stored and default values.

See Also:

[RTRProxyManagedVariableClient](#), [RTRProxyManagedCounter](#), [RTRProxyManagedGauge](#), [RTRProxyManagedNumeric](#), [RTRProxyManagedBoolean](#), [RTRProxyManagedString](#), [RTRProxyManagedNumericRange](#), [RTRProxyManagedNumericConfig](#), [RTRProxyManagedBooleanConfig](#), [RTRProxyManagedGaugeConfig](#)

Constructor & Destructor Documentation

RTRProxyManagedStringConfig::RTRProxyManagedStringConfig ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
 Constructor

virtual **RTRProxyManagedStringConfig::~~RTRProxyManagedStringConfig** () [[virtual](#)]
 Destructor

RTRProxyManagedVarHandleIterator Class Reference

A RTRProxyManagedVarHandleIterator is used to sequentially traverse a set of Proxy Managed Variable Handles.

```
#include <prxy.h>
```

Public Member Functions

- [RTRProxyManagedVarHandleIterator](#) (const RTRDLinkedList< [RTRProxyManagedVariableHandle](#), RTRDLink0 > &)
- int [count](#) () const
- RTRBOOL [off](#) () const
- RTRBOOL [empty](#) () const
- [RTRProxyManagedVariableHandle](#) & [item](#) () const
- void [start](#) ()

- void [finish](#) ()
- void [forth](#) ()
- void [back](#) ()

Detailed Description

A RTRProxyManagedVarHandleIterator is used to sequentially traverse a set of Proxy Managed Variable Handles.

See Also:

[RTRProxyManagedObjectHandle](#), [RTRProxyManagedVariableHandle](#), [RTRProxyManagedObjectHandleIterator](#)

Constructor & Destructor Documentation

**RTRProxyManagedVarHandleIterator::RTRProxyManagedVarHandleIterator (const RTRDLinkList<
[RTRProxyManagedVariableHandle](#), RTRDLink0 > &)**
Constructor

RTRProxyManagedVariable Class Reference

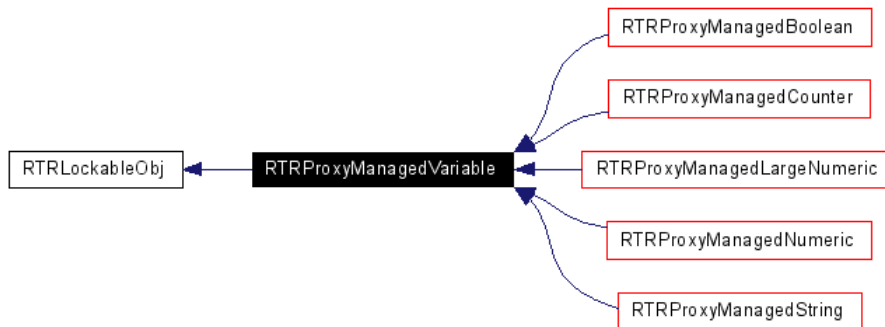
A cloned (proxy) representation of an [RTRManagedVariable](#). The base class for all of the proxy managed variable types. The cloning process could be an asynchronous process and so the state of the proxy variable must be checked before using many of the available operations (methods).

```
#include <prxymv.h>
```

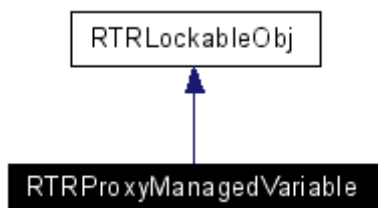
Inherits RTRMTGCObj, [RTRLockableObj](#), and RTRDLink0.

Inherited by [RTRProxyManagedBoolean](#), [RTRProxyManagedCounter](#), [RTRProxyManagedNumeric](#), and [RTRProxyManagedString](#).

Inheritance diagram for RTRProxyManagedVariable:



Collaboration diagram for RTRProxyManagedVariable:



Public Member Functions

- [RTRProxyManagedVariable](#) ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)

- virtual [~RTRProxyManagedVariable](#) ()
- const [RTRString](#) & [name](#) () const
- [RTRProxyManagedObject](#) & [context](#) () const
- const [RTRString](#) & [text](#) () const
- [RTRProxyManagedVariableHandle::MVType](#) [type](#) () const
- const [RTRString](#) & [description](#) () const
- RTRBOOL [error](#) () const
- RTRBOOL [inSync](#) () const
- virtual [RTRString](#) [toString](#) () const =0
- [RTRString](#) [typeString](#) () const
- [operator RTRProxyManagedBoolean &](#) ()
- [operator const RTRProxyManagedBoolean &](#) () const
- operator RTRProxyManagedBooleanConfig & ()
- operator const RTRProxyManagedBooleanConfig & () const
- [operator RTRProxyManagedCounter &](#) ()
- [operator const RTRProxyManagedCounter &](#) () const
- [operator RTRProxyManagedGauge &](#) ()
- [operator const RTRProxyManagedGauge &](#) () const
- operator RTRProxyManagedGaugeConfig & ()
- [operator const RTRProxyManagedGaugeConfig &](#) () const
- [operator RTRProxyManagedNumeric &](#) ()
- [operator const RTRProxyManagedNumeric &](#) () const
- operator RTRProxyManagedNumericConfig & ()
- operator const RTRProxyManagedNumericConfig & () const
- operator RTRProxyManagedNumericRange & ()
- operator const RTRProxyManagedNumericRange & () const
- [operator RTRProxyManagedString &](#) ()
- [operator const RTRProxyManagedString &](#) () const
- operator RTRProxyManagedStringConfig & ()
- operator const RTRProxyManagedStringConfig & () const
- RTRBOOL [hasClient](#) ([RTRProxyManagedVariableClient](#) &) const
- void [addClient](#) ([RTRProxyManagedVariableClient](#) &client)
- void [dropClient](#) ([RTRProxyManagedVariableClient](#) &client)
- virtual void [lock](#) ()

- virtual void **unlock** ()
- virtual RTRBOOL [locked](#) () const

Friends

- class [RTRProxyManagedObject](#)

Detailed Description

A cloned (proxy) representation of an [RTRManagedVariable](#). The base class for all of the proxy managed variable types. The cloning process could be an asynchronous process and so the state of the proxy variable must be checked before using many of the available operations (methods).

Clients may register with instances of this class to receive notifications of variable state changes, attribute changes and deletion of the variable.

See Also:

[RTRProxyManagedVariableClient](#), [RTRProxyManagedCounter](#), [RTRProxyManagedGauge](#), [RTRProxyManagedNumeric](#), [RTRProxyManagedString](#), [RTRProxyManagedBoolean](#), [RTRProxyManagedNumericRange](#), [RTRProxyManagedNumericConfig](#), [RTRProxyManagedStringConfig](#), [RTRProxyManagedBooleanConfig](#), [RTRProxyManagedGaugeConfig](#)

Constructor & Destructor Documentation

RTRProxyManagedVariable::RTRProxyManagedVariable ([RTRProxyManagedObject](#) &, [RTRProxyManagedVariableHandle](#) &)
Constructor

virtual RTRProxyManagedVariable::~RTRProxyManagedVariable () [virtual]
Destructor

RTRProxyManagedVariableClient Class Reference

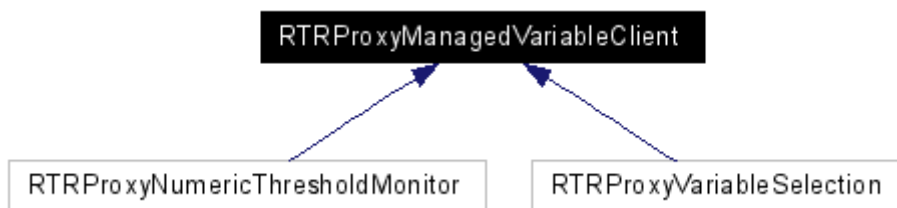
The base class for components which can register with a variable to receive change events from that variable. The notifications are grouped into three categories:

1. proxy variable state changes,
2. the variable has been updated, and
3. the variable has been deleted by the managed application.

```
#include <prxymvc.h>
```

Inherited by [RTRProxyNumericThresholdMonitor](#) and [RTRProxyVariableSelection](#).

Inheritance diagram for [RTRProxyManagedVariableClient](#):



Public Member Functions

- virtual [~RTRProxyManagedVariableClient](#) ()
- virtual void [processProxyManagedVariableError](#) ([RTRProxyManagedVariable](#) &)=0

- virtual void [processProxyManagedVariableSync](#) ([RTRProxyManagedVariable](#) &)=0
- virtual void [processProxyManagedVariableUpdate](#) ([RTRProxyManagedVariable](#) &)=0
- virtual void [processProxyManagedVariableDeleted](#) ([RTRProxyManagedVariable](#) &)=0

Detailed Description

The base class for components which can register with a variable to receive change events from that variable. The notifications are grouped into three categories:

1. proxy variable state changes,
2. the variable has been updated, and
3. the variable has been deleted by the managed application.

See Also:

[RTRProxyManagedVariable](#), [RTRProxyManagedCounter](#), [RTRProxyManagedGauge](#), [RTRProxyManagedNumeric](#), [RTRProxyManagedString](#), [RTRProxyManagedBoolean](#), [RTRProxyManagedNumericRange](#), [RTRProxyManagedNumericConfig](#), [RTRProxyManagedStringConfig](#), [RTRProxyManagedBooleanConfig](#), [RTRProxyManagedGaugeConfig](#)

Constructor & Destructor Documentation

virtual RTRProxyManagedVariableClient::~~RTRProxyManagedVariableClient () [virtual]
Destructor

RTRProxyManagedVariableHandle Class Reference

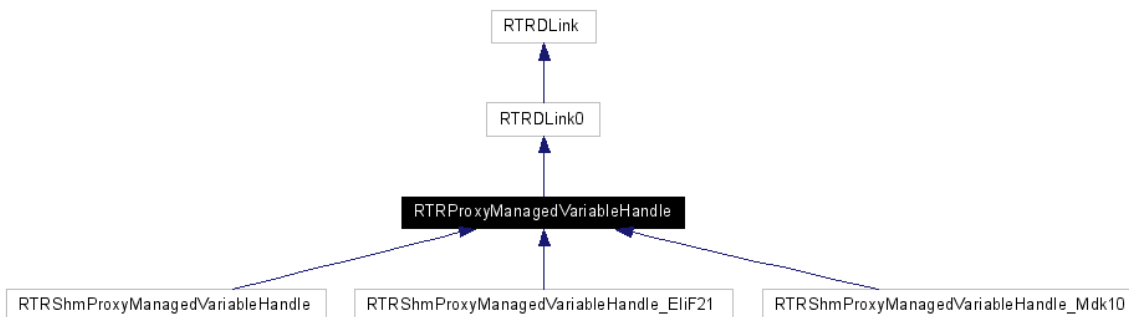
A RTRProxyManagedVariableHandle uniquely identifies a Proxy Managed Variable. The handle is used to request a clone (proxy) of a particular managed variable.

```
#include <prxyh.h>
```

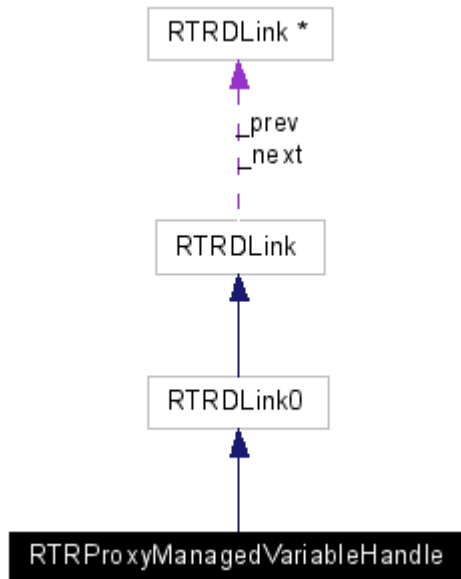
Inherits RTRDLink0.

Inherited by RTRShmProxyManagedVariableHandle, RTRShmProxyManagedVariableHandle_EliF21, and RTRShmProxyManagedVariableHandle_Mdk10.

Inheritance diagram for RTRProxyManagedVariableHandle:



Collaboration diagram for RTRProxyManagedVariableHandle:



Public Types

- enum [MVType](#) { Counter, Numeric, NumericRange, Gauge, String, GaugeConfig, NumericConfig, StringConfig, Boolean, BooleanConfig, Invalid, LastValueDummy }

Public Member Functions

- [RTRProxyManagedVariableHandle](#) (const [RTRString](#) &name, int type)
- virtual [~RTRProxyManagedVariableHandle](#) ()
- const [RTRString](#) & [name](#) () const
- [MVType](#) [type](#) () const
- [RTRString](#) [typeString](#) () const

Static Public Member Functions

- static const char * [typeToString](#) ([RTRProxyManagedVariableHandle::MVType](#))

Detailed Description

A [RTRProxyManagedVariableHandle](#) uniquely identifies a Proxy Managed Variable. The handle is used to request a clone (proxy) of a particular managed variable.

See Also:

[RTRProxyManagedObjectHandle](#), [RTRProxyManagedObjectHandleIterator](#), [RTRProxyManagedVarHandleIterator](#)

Member Enumeration Documentation

enum [RTRProxyManagedVariableHandle::MVType](#)

The proxy managed variable type

Constructor & Destructor Documentation

[RTRProxyManagedVariableHandle::RTRProxyManagedVariableHandle](#) (const [RTRString](#) & *name*, int *type*)

Constructor

```
virtual RTRProxyManagedVariableHandle::~~RTRProxyManagedVariableHandle () [inline, virtual]
```

Destructor

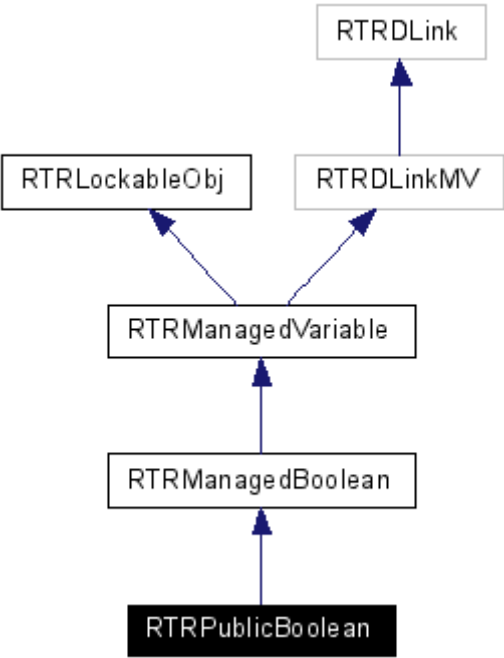
RTRPublicBoolean Class Reference

An implementation of the [RTRManagedBoolean](#) base class which provides modification operations and uses the global instance of RTRMOServerMemPool for storage allocation.

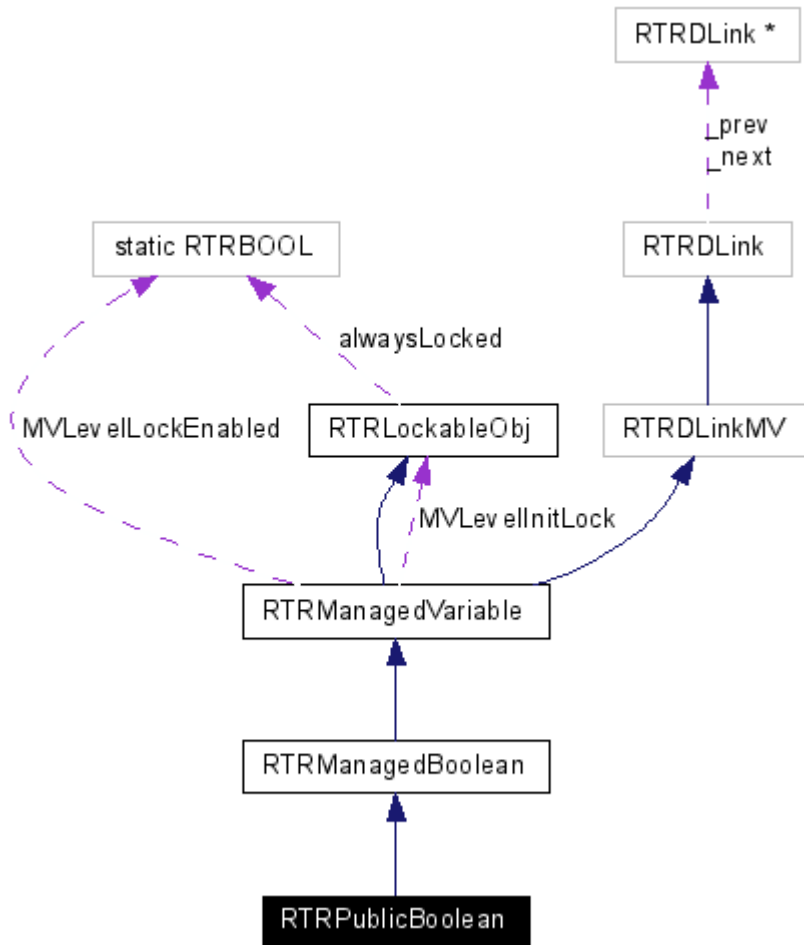
```
#include <pmbv.h>
```

Inherits [RTRManagedBoolean](#).

Inheritance diagram for RTRPublicBoolean:



Collaboration diagram for RTRPublicBoolean:



Public Member Functions

- [RTRPublicBoolean](#) ([RTRPublicObject](#) &context, const char *name, const char *description, RTRBOOL initValue, RTRBOOL modifyEnabled=RTRFALSE)
- virtual [~RTRPublicBoolean](#) ()
- [RTRPublicBoolean](#) & [operator=](#) (RTRBOOL rhs)
- void [internalSet](#) ()
- void [internalClear](#) ()

Detailed Description

An implementation of the [RTRManagedBoolean](#) base class which provides modification operations and uses the global instance of RTRMOServerMemPool for storage allocation.

If modifyEnabled is true (default value is false), then consumers will be permitted to modify the active value.

See Also:

[RTRPublicObject](#), [RTRPublicString](#), [RTRPublicNumeric](#), [RTRPublicCounter](#), [RTRPublicGauge](#), [RTRPublicNumericRange](#), [RTRPublicStringConfig](#), [RTRPublicNumericConfig](#), [RTRPublicBooleanConfig](#), [RTRPublicGaugeConfig](#)

Constructor & Destructor Documentation

RTRPublicBoolean::RTRPublicBoolean ([RTRPublicObject](#) & *context*, const char * *name*, const char * *description*, RTRBOOL *initValue*, RTRBOOL *modifyEnabled* = RTRFALSE)

Constructs an RTRPublicBoolean

ENSURE: [value\(\)](#) == *initValue*

virtual RTRPublicBoolean::~~RTRPublicBoolean () [virtual]

Destructor

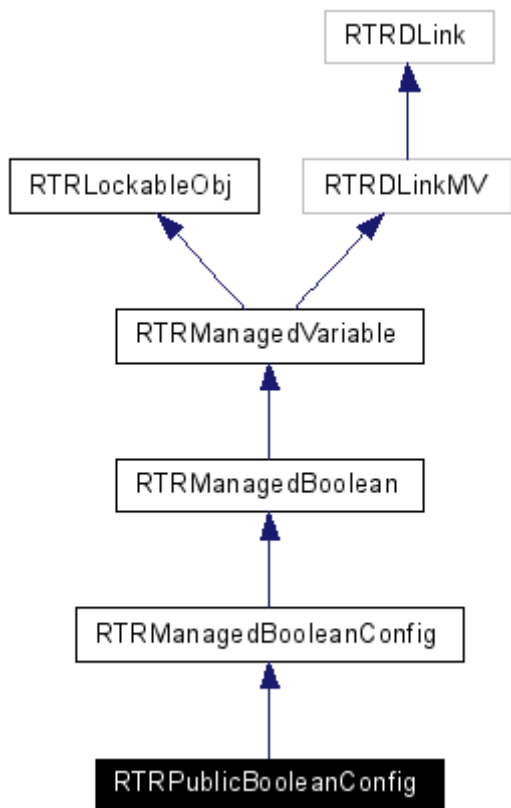
RTRPublicBooleanConfig Class Reference

An implementation of the [RTRManagedBooleanConfig](#) base class which provides modification operations and uses the global instance of RTRMOServerMemPool for storage allocation.

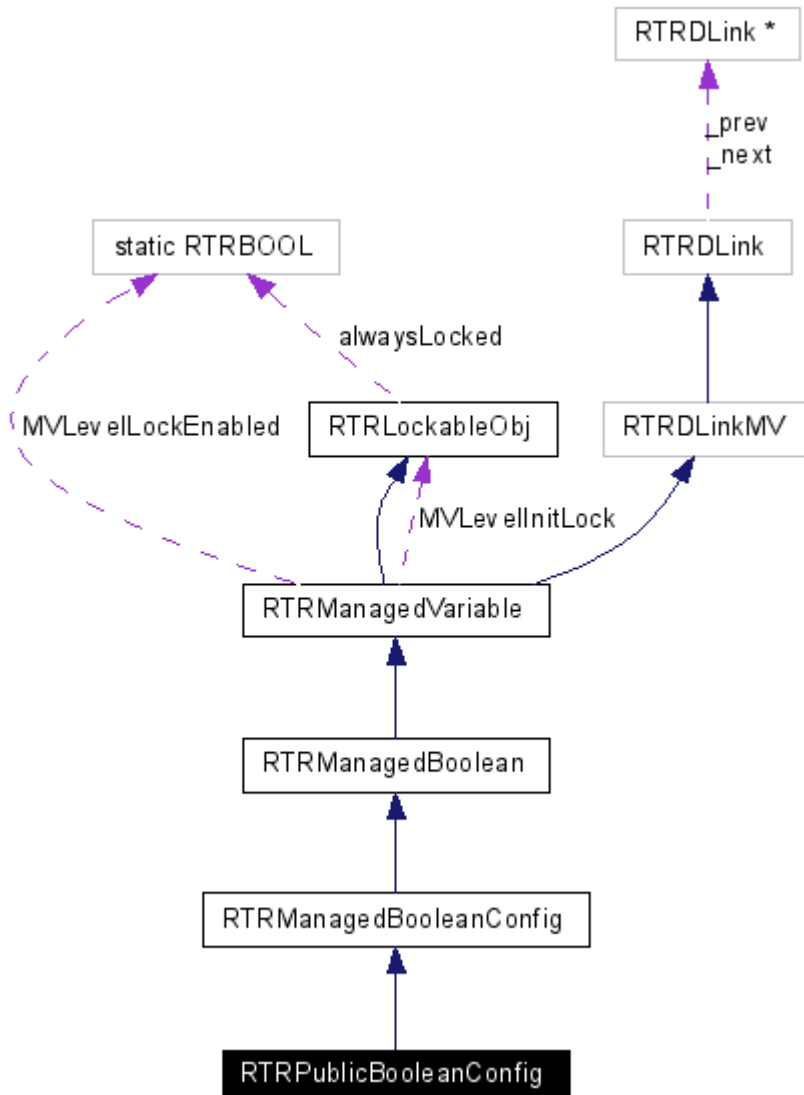
```
#include <pmbcv.h>
```

Inherits [RTRManagedBooleanConfig](#).

Inheritance diagram for RTRPublicBooleanConfig:



Collaboration diagram for RTRPublicBooleanConfig:



Public Member Functions

- [RTRPublicBooleanConfig](#) ([RTRPublicObject](#) &context, const char *name, const char *description, RTRBOOL defValue, RTRBOOL modifyEnabled=RTRFALSE)
- virtual [~RTRPublicBooleanConfig](#) ()
- [RTRPublicBooleanConfig](#) & [operator=](#) (RTRBOOL rhs)
- void [internalSet](#) ()
- void [internalClear](#) ()
- void [setStore](#) ()
- void [clearStore](#) ()

Detailed Description

An implementation of the [RTRManagedBooleanConfig](#) base class which provides modification operations and uses the global instance of RTRMOServerMemPool for storage allocation.

If modifyEnabled is true (default value is false), then consumers will be permitted to modify the active value.

See Also:

[RTRPublicObject](#), [RTRPublicString](#), [RTRPublicNumeric](#), [RTRPublicCounter](#), [RTRPublicGauge](#), [RTRPublicBoolean](#), [RTRPublicNumericRange](#), [RTRPublicStringConfig](#), [RTRPublicNumericConfig](#), [RTRPublicGaugeConfig](#)

Constructor & Destructor Documentation

RTRPublicBooleanConfig::RTRPublicBooleanConfig ([RTRPublicObject](#) & context, const char * name, const char * description, RTRBOOL dfltValue, RTRBOOL modifyEnabled = RTRFALSE)

Constructs an RTRPublicBooleanConfig

ENSURE: [activeValue\(\)](#) == dfltValue

ENSURE: [storeValue\(\)](#) == dfltValue

ENSURE: [factoryDefault\(\)](#) == dfltValue

virtual RTRPublicBooleanConfig::~RTRPublicBooleanConfig () [virtual]

Destructor

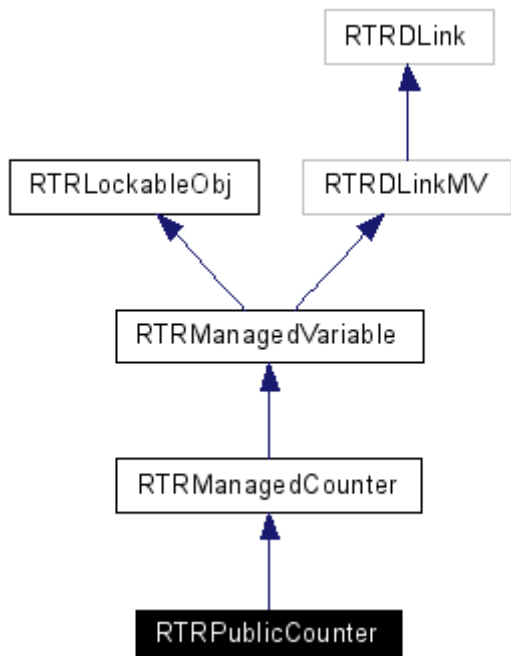
RTRPublicCounter Class Reference

An implementation of the ManagedCounter base class which provides increment capability and uses the class RTRMNumericImpl for storage allocation. Note: counters can be reset to 0 and incremented, but not decremented.

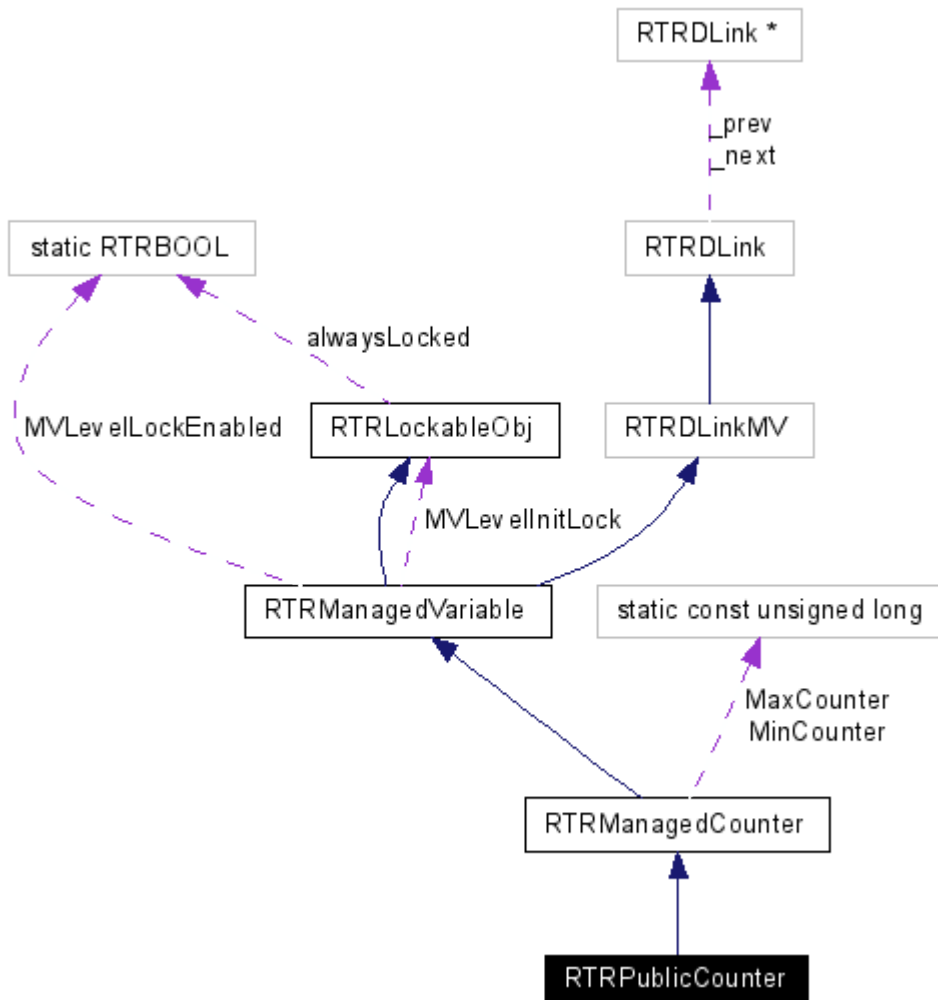
```
#include <pmcntrv.h>
```

Inherits [RTRManagedCounter](#).

Inheritance diagram for RTRPublicCounter:



Collaboration diagram for RTRPublicCounter:



Public Member Functions

- [RTRPublicCounter](#) ([RTRPublicObject](#) &context, const char *name, const char *description, unsigned long initValue=0)
- virtual [~RTRPublicCounter](#) ()
- virtual void [reset](#) ()
- void **operator+=** (unsigned long)
- [RTRPublicCounter](#) & **operator++** ()
- [RTRPublicCounter](#) & **operator++** (int)

Detailed Description

An implementation of the ManagedCounter base class which provides increment capability and uses the class RTRMNumericImpl for storage allocation. Note: counters can be reset to 0 and incremented, but not decremented.

See Also:

[RTRPublicObject](#), [RTRPublicString](#), [RTRPublicNumeric](#), [RTRPublicGauge](#), [RTRPublicBoolean](#), [RTRPublicNumericRange](#), [RTRPublicStringConfig](#), [RTRPublicNumericConfig](#), [RTRPublicBooleanConfig](#), [RTRPublicGaugeConfig](#),

Constructor & Destructor Documentation

RTRPublicCounter::RTRPublicCounter ([RTRPublicObject](#) & *context*, const char * *name*, const char * *description*, unsigned long *initValue* = 0)

Constructs an RTRPublicCounter variable

ENSURE: [value\(\)](#) == *initValue*

virtual RTRPublicCounter::~~RTRPublicCounter () [virtual]

Destructor

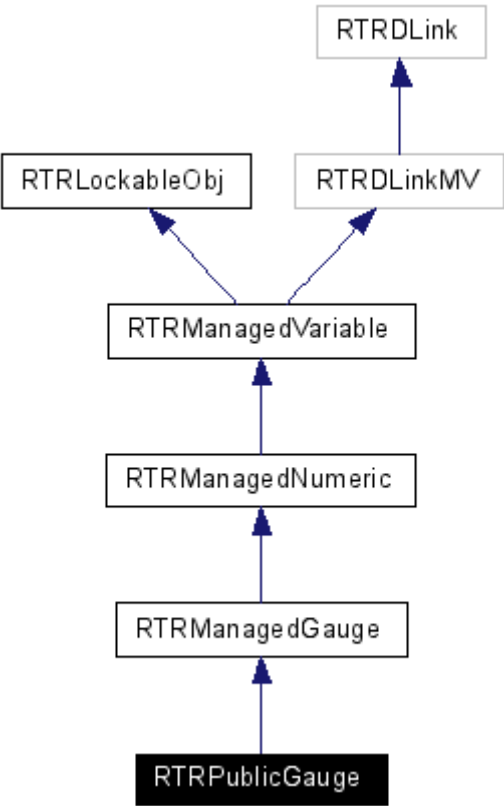
RTRPublicGauge Class Reference

An implementation of the [RTRManagedGauge](#) base class which provides modification operations and uses the global instance of RTRMOServerMemPool for storage allocation.

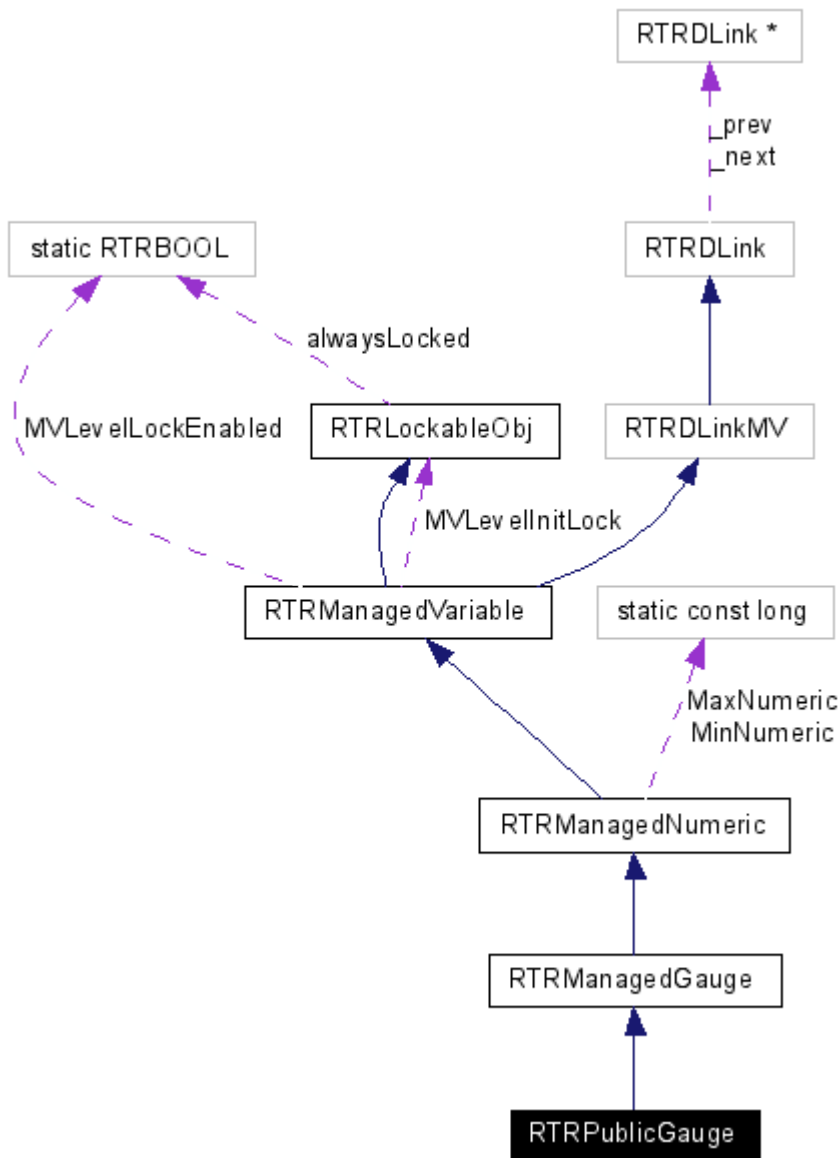
```
#include <pmgv.h>
```

Inherits [RTRManagedGauge](#).

Inheritance diagram for RTRPublicGauge:



Collaboration diagram for RTRPublicGauge:



Public Member Functions

- [RTRPublicGauge](#) ([RTRPublicObject](#) &context, const char *name, const char *description, long initValue, long min=0, long max=RTRManagedNumeric::MaxNumeric, RTRBOOL modifyEnabled=RTRFALSE)
- virtual [~RTRPublicGauge](#) ()
- [RTRPublicGauge](#) & [operator=](#) (long rhs)
- void [operator+=](#) (long)
- void [operator-=](#) (long)
- [RTRPublicGauge](#) & [operator++](#) ()
- [RTRPublicGauge](#) & [operator++](#) (int)
- [RTRPublicGauge](#) & [operator--](#) ()

- [RTRPublicGauge](#) & [operator--](#) (int)
- void [set](#) (long newValue)
- void [set](#) (long newMin, long newMax, long newValue)
- virtual void [internalSetRange](#) (long newMin, long newMax)

Detailed Description

An implementation of the [RTRManagedGauge](#) base class which provides modification operations and uses the global instance of RTRMOServerMemPool for storage allocation.

If modifyEnabled is true (default value is false), then consumers will be permitted to modify the active min/max values. The context (ManagedObject) will be notified of changes.

if modifyEnabled is false, then the active value must always be between [minValue\(\)](#) and [maxValue\(\)](#) (ie. minValue <= activeValue <= maxValue). This restriction does not exist if the consumer is permitted to modify the min/max values. The active value could be outside the min/max range.

See Also:

[RTRPublicObject](#), [RTRPublicString](#), [RTRPublicNumeric](#), [RTRPublicCounter](#), [RTRPublicBoolean](#), [RTRPublicNumericRange](#), [RTRPublicStringConfig](#), [RTRPublicNumericConfig](#), [RTRPublicGaugeConfig](#), [RTRPublicBooleanConfig](#),

Constructor & Destructor Documentation

RTRPublicGauge::RTRPublicGauge ([RTRPublicObject](#) & context, const char * name, const char * description, long initValue, long min = 0, long max = RTRManagedNumeric::MaxNumeric, RTRBOOL modifyEnabled = RTRFALSE)

Constructs an RTRPublicGauge variable

REQUIRE: initValue >= min

REQUIRE: initValue <= max

REQUIRE: min <= max

ENSURE: [lowWaterMark\(\)](#) == [highWaterMark\(\)](#) == [value\(\)](#)

ENSURE: [value\(\)](#) == initValue

ENSURE: [minValue\(\)](#) == min

ENSURE: [maxValue\(\)](#) == max

virtual RTRPublicGauge::~~RTRPublicGauge () [virtual]

Destructor

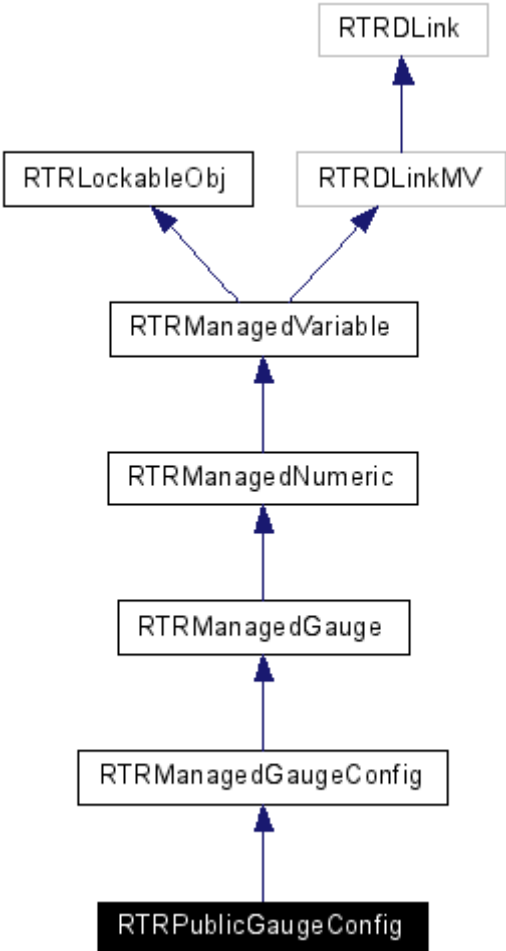
RTRPublicGaugeConfig Class Reference

An implementation of the [RTRManagedGaugeConfig](#) base class which provides modification operations and uses the global instance of RTRMOServerMemPool for storage allocation.

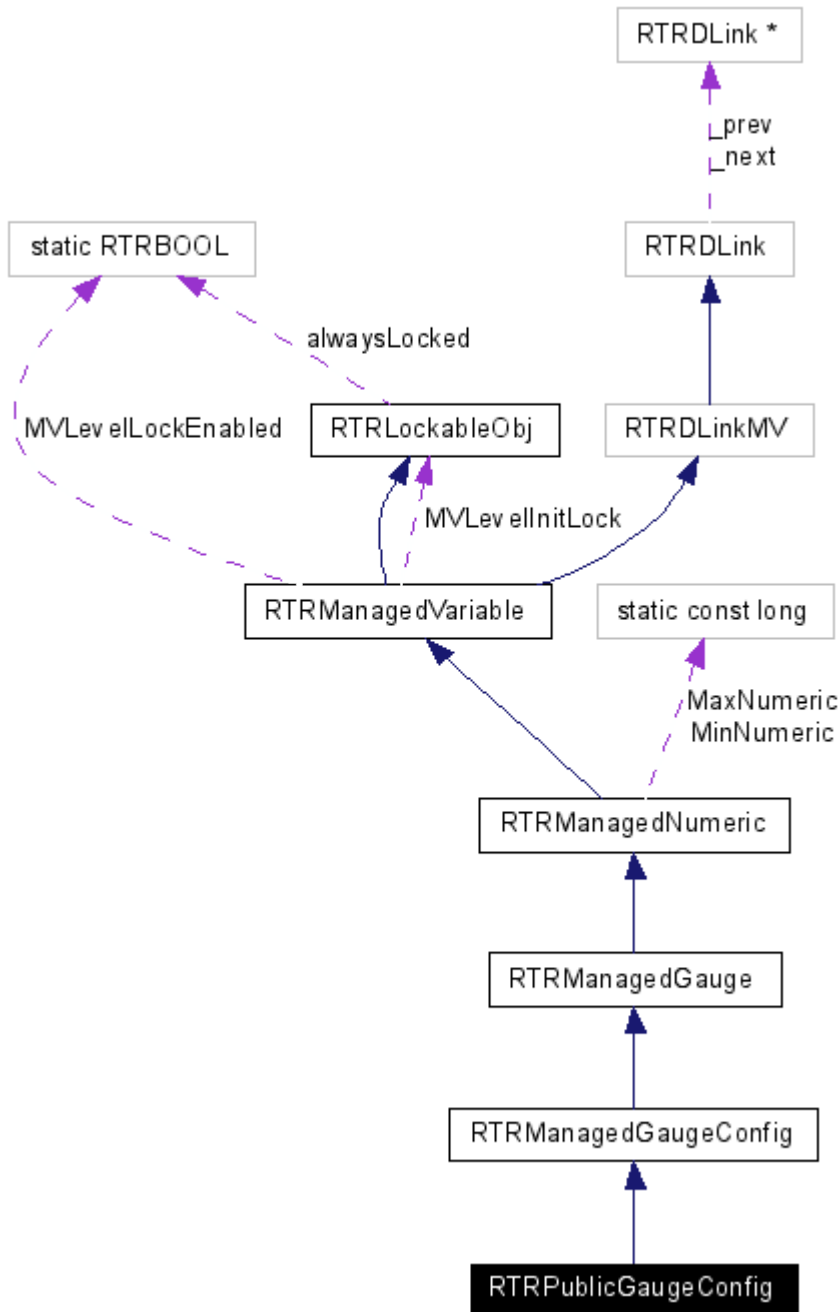
```
#include <pmgcv.h>
```

Inherits [RTRManagedGaugeConfig](#).

Inheritance diagram for RTRPublicGaugeConfig:



Collaboration diagram for RTRPublicGaugeConfig:



Public Member Functions

- [RTRPublicGaugeConfig](#) ([RTRPublicObject](#) &context, const char *name, const char *description, long initValue, long minDflt=0, long maxDflt=RTRManagedNumeric::MaxNumeric, RTRBOOL modifyEnabled=RTRFALSE)
- virtual [~RTRPublicGaugeConfig](#) ()
- [RTRPublicGaugeConfig](#) & [operator=](#) (long rhs)
- void [operator+=](#) (long)

- void [operator-=](#) (long)
- [RTRPublicGaugeConfig](#) & [operator++](#) ()
- [RTRPublicGaugeConfig](#) & [operator++](#) (int)
- [RTRPublicGaugeConfig](#) & [operator--](#) ()
- [RTRPublicGaugeConfig](#) & [operator--](#) (int)
- void [set](#) (long newValue)
- void [set](#) (long newMin, long newMax, long newValue)
- void [internalSetRange](#) (long newMin, long newMax)
- void [setStore](#) (long newMin, long newMax)

Detailed Description

An implementation of the [RTRManagedGaugeConfig](#) base class which provides modification operations and uses the global instance of RTRMOServerMemPool for storage allocation.

If modifyEnabled is true (default value is false), then consumers will be permitted to modify the active min/max values.

if modifyEnabled is false, then the active value will be between [minValue\(\)](#) and [maxValue\(\)](#) (i.e. minValue <= activeValue <= maxValue). This restriction does not exist if the consumer is permitted to modify the min/max values. The active value could be outside the min/max range.

See Also:

[RTRPublicObject](#), [RTRPublicString](#), [RTRPublicNumeric](#), [RTRPublicCounter](#), [RTRPublicBoolean](#), [RTRPublicNumericRange](#), [RTRPublicGauge](#), [RTRPublicStringConfig](#), [RTRPublicNumericConfig](#), [RTRPublicBooleanConfig](#).

Constructor & Destructor Documentation

RTRPublicGaugeConfig::RTRPublicGaugeConfig ([RTRPublicObject](#) & *context*, const char * *name*, const char * *description*, long *initValue*, long *minDflt* = 0, long *maxDflt* = [RTRManagedNumeric::MaxNumeric](#), [RTRBOOL](#) *modifyEnabled* = [RTRFALSE](#))

Constructs an RTRPublicGaugeConfig

REQUIRE: *initValue* >= *minDflt*

REQUIRE: *initValue* <= *maxDflt*

REQUIRE: *minDflt* <= *maxDflt*

ENSURE: [lowWaterMark\(\)](#) == [highWaterMark\(\)](#) == [value\(\)](#)

ENSURE: [value\(\)](#) == *initValue*

ENSURE: [minValue\(\)](#) == [minStoreValue\(\)](#) == [minFactoryDefault\(\)](#) == *minDflt*

ENSURE: [maxValue\(\)](#) == [maxStoreValue\(\)](#) == [maxFactoryDefault\(\)](#) == *maxDflt*

virtual RTRPublicGaugeConfig::~~RTRPublicGaugeConfig () [[virtual](#)]

Destructor

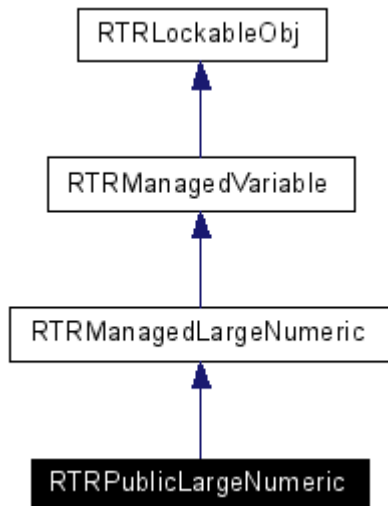
RTRPublicLargeNumeric Class Reference

An implementation of the [RTRManagedLargeNumeric](#) base class which provides modification operations and uses the class RTRMNumericImpl for storage allocation.

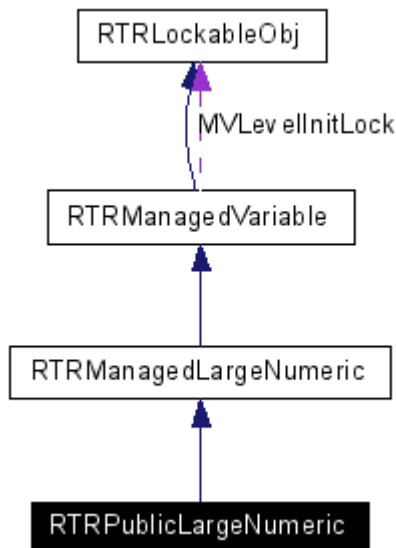
```
#include <pm1numv.h>
```

Inherits [RTRManagedLargeNumeric](#).

Inheritance diagram for RTRPublicLargeNumeric:



Collaboration diagram for RTRPublicLargeNumeric:



Public Member Functions

- [RTRPublicLargeNumeric](#) ([RTRPublicObject](#) &context, const char *name, const char *description, RTR_I64 initValue)
- virtual [~RTRPublicLargeNumeric](#) ()
- [RTRPublicLargeNumeric](#) & **operator=** (RTR_I64)
- void **operator+=** (RTR_I64)
- void **operator-=** (RTR_I64)
- [RTRPublicLargeNumeric](#) & **operator++** ()
- [RTRPublicLargeNumeric](#) & **operator++** (int)
- [RTRPublicLargeNumeric](#) & **operator--** ()

- [RTRPublicLargeNumeric](#) & **operator--** (int)
- void **set** (RTR_I64)

Detailed Description

An implementation of the [RTRManagedLargeNumeric](#) base class which provides modification operations and uses the class RTRMNumericImpl for storage allocation.

See Also:

[RTRPublicObject](#), [RTRPublicString](#), [RTRPublicBoolean](#), [RTRPublicCounter](#), [RTRPublicGauge](#), [RTRPublicNumericRange](#), [RTRPublicStringConfig](#), [RTRPublicNumeric](#), [RTRPublicNumericConfig](#), [RTRPublicGaugeConfig](#), [RTRPublicBooleanConfig](#)

Constructor & Destructor Documentation

RTRPublicLargeNumeric::RTRPublicLargeNumeric ([RTRPublicObject](#) & *context*, const char * *name*, const char * *description*, RTR_I64 *initValue*)

Constructs an RTRPublicLargeNumeric

ENSURE: [value\(\)](#) == initValue

virtual RTRPublicLargeNumeric::~~RTRPublicLargeNumeric () [virtual]

Destructor

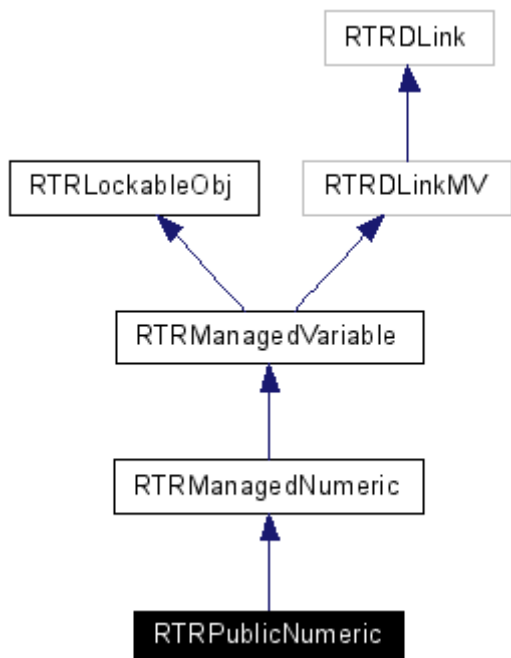
RTRPublicNumeric Class Reference

An implementation of the [RTRManagedNumeric](#) base class which provides modification operations and uses the class RTRMNumericImpl for storage allocation.

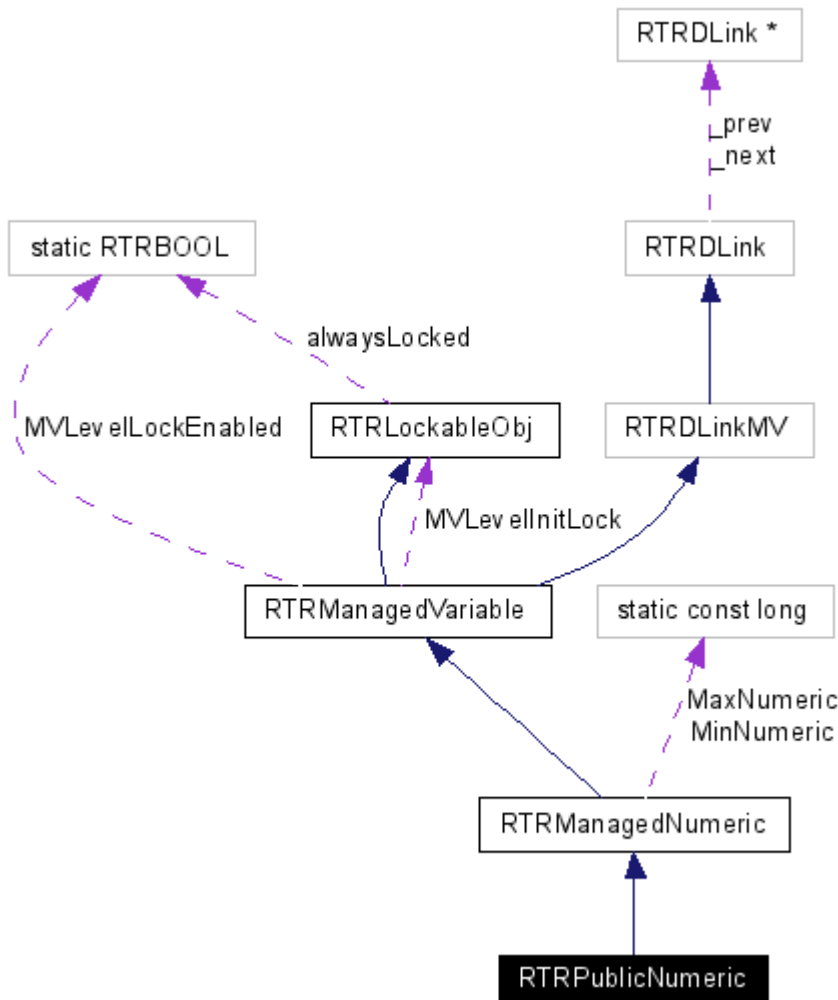
```
#include <pmnumv.h>
```

Inherits [RTRManagedNumeric](#).

Inheritance diagram for RTRPublicNumeric:



Collaboration diagram for RTRPublicNumeric:



Public Member Functions

- [RTRPublicNumeric](#) ([RTRPublicObject](#) &context, const char *name, const char *description, long initValue)
- virtual [~RTRPublicNumeric](#) ()
- [RTRPublicNumeric](#) & **operator=** (long)
- void **operator+=** (long)
- void **operator-=** (long)
- [RTRPublicNumeric](#) & **operator++** ()
- [RTRPublicNumeric](#) & **operator++** (int)
- [RTRPublicNumeric](#) & **operator--** ()
- [RTRPublicNumeric](#) & **operator--** (int)
- void **set** (long)

Detailed Description

An implementation of the [RTRManagedNumeric](#) base class which provides modification operations and uses the class RTRMNumericImpl for storage allocation.

See Also:

[RTRPublicObject](#), [RTRPublicString](#), [RTRPublicBoolean](#), [RTRPublicCounter](#), [RTRPublicGauge](#), [RTRPublicNumericRange](#), [RTRPublicStringConfig](#), [RTRPublicNumericConfig](#), [RTRPublicGaugeConfig](#), [RTRPublicBooleanConfig](#)

Constructor & Destructor Documentation

RTRPublicNumeric::RTRPublicNumeric ([RTRPublicObject](#) & *context*, const char * *name*, const char * *description*, long *initValue*)

Constructs an RTRPublicNumeric

ENSURE: [value\(\)](#) == initValue

virtual RTRPublicNumeric::~~RTRPublicNumeric () [virtual]

Destructor

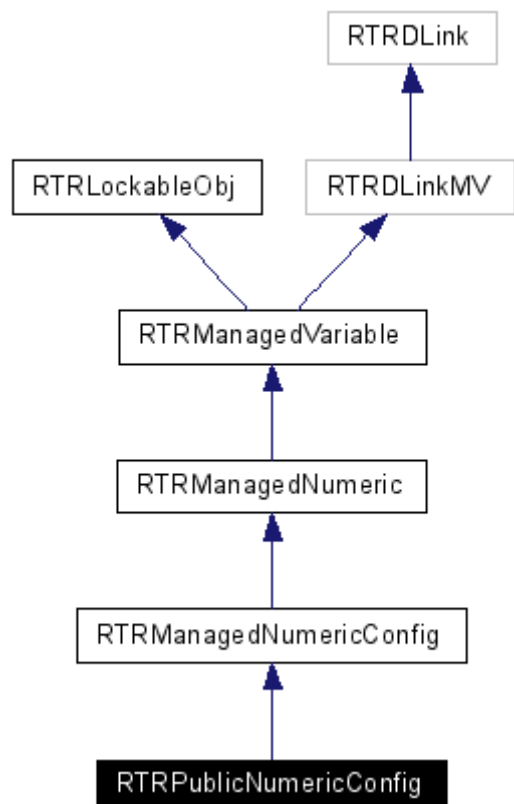
RTRPublicNumericConfig Class Reference

An implementation of the [RTRManagedNumericConfig](#) base class which uses the class RTRMNumConfigImpl for storage allocation.

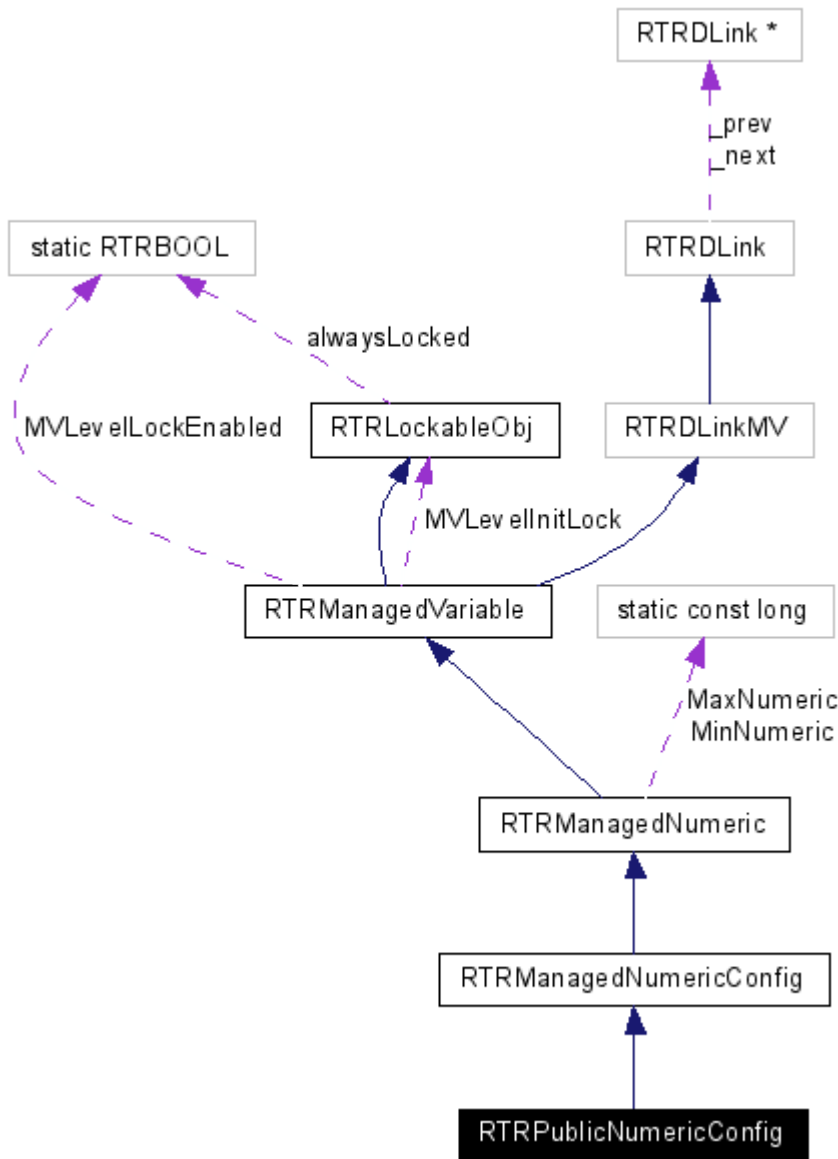
```
#include <pmnumcv.h>
```

Inherits [RTRManagedNumericConfig](#).

Inheritance diagram for RTRPublicNumericConfig:



Collaboration diagram for RTRPublicNumericConfig:



Public Member Functions

- [RTRPublicNumericConfig](#) ([RTRPublicObject](#) &context, const char *name, const char *description, long dfltValue, long min=0, long max=RTRManagedNumeric::MaxNumeric, RTRBOOL modifyEnabled=RTRFALSE)
- virtual [~RTRPublicNumericConfig](#) ()
- [RTRPublicNumericConfig](#) & [operator=](#) (long rhs)
- void [internalSet](#) (long newValue)
- void [setStore](#) (long newStore)

Detailed Description

An implementation of the [RTRManagedNumericConfig](#) base class which uses the class RTRMNumConfigImpl for storage allocation.

If modifyEnabled is true (default value is false), then consumers will be permitted to modify the active value.

The value will always be within the min/max range. (ie. [minValue\(\)](#) <= [value\(\)](#) <= [maxValue\(\)](#))

See Also:

[RTRPublicObject](#), [RTRPublicString](#), [RTRPublicNumeric](#), [RTRPublicCounter](#), [RTRPublicGauge](#), [RTRPublicBoolean](#), [RTRPublicNumericRange](#), [RTRPublicStringConfig](#), [RTRPublicBooleanConfig](#), [RTRPublicGaugeConfig](#)

Constructor & Destructor Documentation

RTRPublicNumericConfig::RTRPublicNumericConfig ([RTRPublicObject](#) & *context*, const char * *name*, const char * *description*, long *dfltValue*, long *min* = 0, long *max* = RTRManagedNumeric::MaxNumeric, RTRBOOL *modifyEnabled* = RTRFALSE)

Constructs an RTRPublicNumericConfig variable

REQUIRE: *dfltValue* >= *min*

REQUIRE: *dfltValue* <= *max*

REQUIRE: *min* <= *max*

ENSURE: [activeValue\(\)](#) == [storeValue\(\)](#) == [factoryDefault\(\)](#) == *dfltValue*

ENSURE: [minValue\(\)](#) == *min*

ENSURE: [maxValue\(\)](#) == *max*

virtual RTRPublicNumericConfig::~~RTRPublicNumericConfig () [virtual]

Destructor

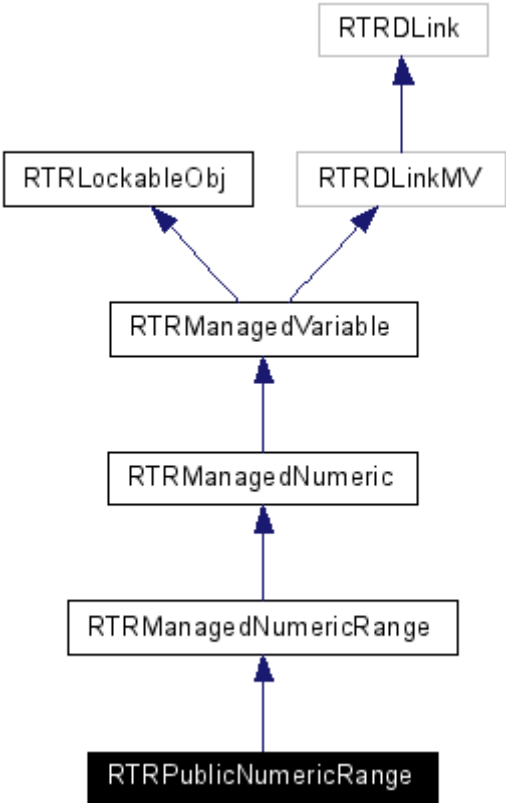
RTRPublicNumericRange Class Reference

An implementation of the [RTRManagedNumericRange](#) base class which uses the class RTRMNumRangeImpl for storage allocation.

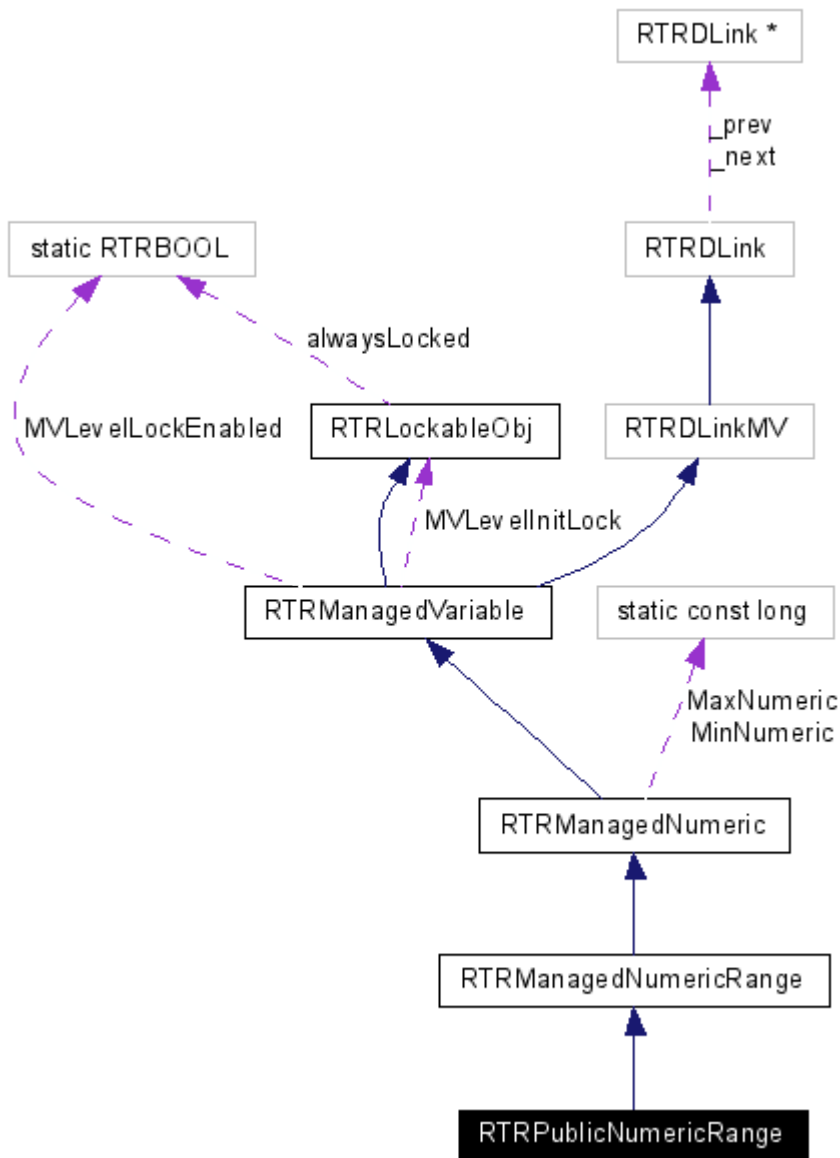
```
#include <pmnumrv.h>
```

Inherits [RTRManagedNumericRange](#).

Inheritance diagram for RTRPublicNumericRange:



Collaboration diagram for `RTRPublicNumericRange`:



Public Member Functions

- [RTRPublicNumericRange](#) ([RTRPublicObject](#) &context, const char *name, const char *description, long initValue, long min=0, long max=RTRManagedNumeric::MaxNumeric)
- virtual [~RTRPublicNumericRange](#) ()
- [RTRPublicNumericRange](#) & [operator=](#) (long rhs)
- void [set](#) (long newValue)
- void [internalSet](#) (long newValue)
- void [set](#) (long newMin, long newMax, long newValue)

Detailed Description

An implementation of the [RTRManagedNumericRange](#) base class which uses the class RTRMNumRangeImpl for storage allocation. The value will always be within the min/max range. (ie. [minValue\(\)](#) <= [value\(\)](#) <= [maxValue\(\)](#))

See Also:

[RTRPublicObject](#), [RTRPublicString](#), [RTRPublicNumeric](#), [RTRPublicCounter](#), [RTRPublicGauge](#), [RTRPublicBoolean](#), [RTRPublicNumericConfig](#), [RTRPublicStringConfig](#), [RTRPublicBooleanConfig](#), [RTRPublicGaugeConfig](#)

Constructor & Destructor Documentation

RTRPublicNumericRange::RTRPublicNumericRange ([RTRPublicObject](#) & *context*, const char * *name*, const char * *description*, long *initValue*, long *min* = 0, long *max* = RTRManagedNumeric::MaxNumeric)

Constructs an RTRPublicNumericRange variable

REQUIRE: *initValue* >= *min*

REQUIRE: *initValue* <= *max*

REQUIRE: *min* <= *max*

ENSURE: [value\(\)](#) == *initValue*

ENSURE: [minValue\(\)](#) == *min*

ENSURE: [maxValue\(\)](#) == *max*

virtual RTRPublicNumericRange::~RTRPublicNumericRange () [virtual]

Destructor

RTRPublicObject Class Reference

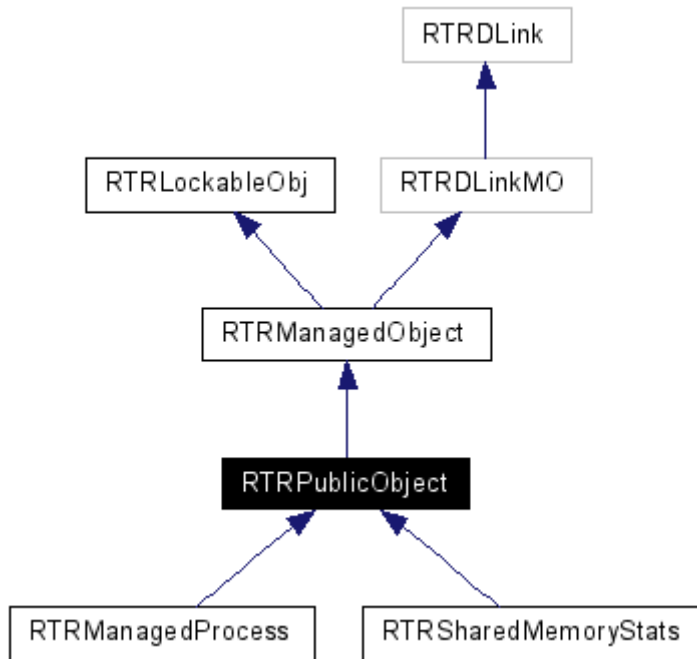
Typically, application components which wish to be managed or become "public" are descendants of RTRPublicObject. They in turn may instantiate other public objects which will be their children in the managed object tree.

```
#include <pmo.h>
```

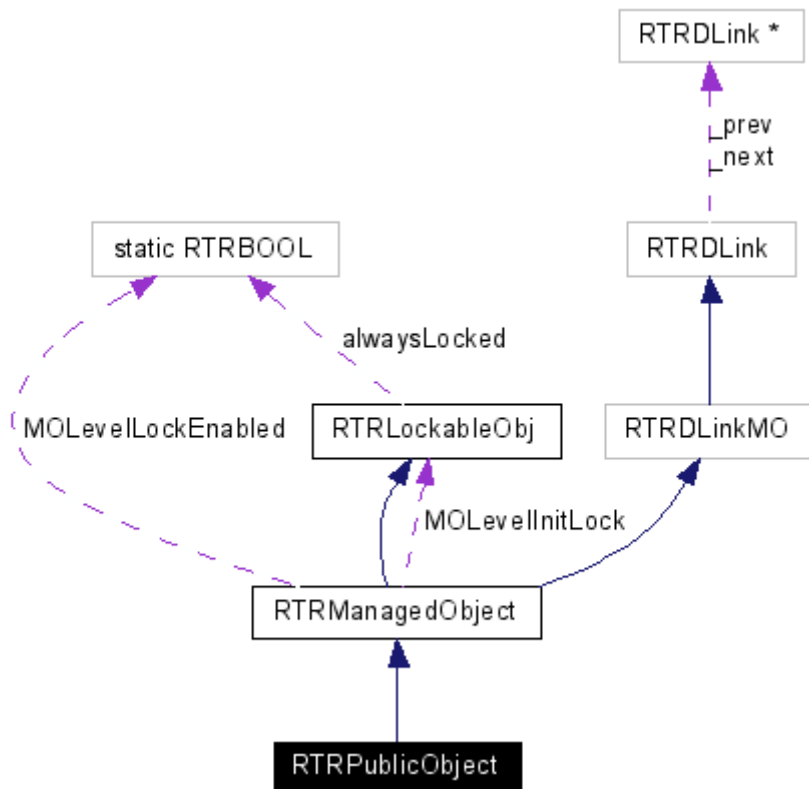
Inherits [RTRManagedObject](#).

Inherited by [RTRManagedProcess](#), and [RTRSharedMemoryStats](#).

Inheritance diagram for RTRPublicObject:



Collaboration diagram for RTRPublicObject:



Public Member Functions

- [RTRPublicObject](#) (const [RTRObjectId](#) &context, const char *name, const char *description, const [RTRObjectId](#) &classId)
- [RTRPublicObject](#) ([RTRPublicObject](#) &parentObject, const char *name, const char *description, const [RTRObjectId](#) &classId)
- [RTRPublicObject](#) (const [RTRObjectId](#) &context, const char *name, const char *description, const [RTRObjectId](#) &classId, [MOState](#) startState)
- [RTRPublicObject](#) ([RTRPublicObject](#) &parentObject, const char *name, const char *description, const [RTRObjectId](#) &classId, [MOState](#) startState)
- [~RTRPublicObject](#) ()
- void [markNormal](#) (const char *)
- void [markRecovering](#) (const char *)
- void [markWaiting](#) (const char *)
- void [markDead](#) (const char *)
- void [indicateInfo](#) (const char *)

Detailed Description

Typically, application components which wish to be managed or become "public" are descendants of RTRPublicObject. They in turn may instantiate other public objects which will be their children in the managed object tree.

Public objects may create member variables as appropriate, using the appropriate Public variable class. For example, to instantiate a numeric counter, the [RTRPublicCounter](#) class would be used.

See Also:

[RTRPublicCounter](#), [RTRPublicGauge](#), [RTRPublicNumeric](#), [RTRPublicString](#), [RTRPublicNumericRange](#), [RTRPublicStringConfig](#), [RTRPublicNumericConfig](#), [RTRPublicBooleanConfig](#), [RTRPublicGaugeConfig](#), [RTRPublicBoolean](#)

Constructor & Destructor Documentation

RTRPublicObject::RTRPublicObject (const [RTRObjectId](#) & context, const char * name, const char * description, const [RTRObjectId](#) & classId)

A new root object in the Normal state

RTRPublicObject::RTRPublicObject ([RTRPublicObject](#) & parentObject, const char * name, const char * description, const [RTRObjectId](#) & classId)

A new sub-object in the Normal state

RTRPublicObject::RTRPublicObject (const [RTRObjectId](#) & context, const char * name, const char * description, const [RTRObjectId](#) & classId, [MOState](#) startState)

A new root object in the given state

RTRPublicObject::RTRPublicObject ([RTRPublicObject](#) & parentObject, const char * name, const char * description, const [RTRObjectId](#) & classId, [MOState](#) startState)

A new sub-object in the given state

RTRPublicObject::~~RTRPublicObject ()

Destructor

RTRPublicObjectLock Class Reference

A construct that is convenient in a multi-thread application where synchronization is needed for accessing managed object directory (MOD) and parent managed object. For example, when constructing/desconstructing instances of [RTRPublicObject](#) in multiple threads, instance of this can be constructed on stack to lock the global object tree, and when this instance is out of scope, its destructor is called to unlock the global object tree.

```
#include <pmo.h>
```

Public Member Functions

- [RTRPublicObjectLock](#) ([RTRManagedObject](#) *parent=0)
- [~RTRPublicObjectLock](#) ()

Detailed Description

A construct that is convenient in a multi-thread application where synchronization is needed for accessing managed object directory (MOD) and parent managed object. For example, when constructing/desconstructing instances of [RTRPublicObject](#) in multiple threads, instance of this can be constructed on stack to lock the global object tree, and when this instance is out of scope, its destructor is called to unlock the global object tree.

```
Example code:
// A block of code in one thread
{
    RTRPublicObjectLock lock1;
    //Now the global object directory is locked
    root = new RTRPublicObject(...);
}
// lock1 now is out of scope
```

See Also:

[RTRPublicObject](#), [RTRGlobalManagedObjectDirectory](#), [RTRLockableObj](#)

Constructor & Destructor Documentation

RTRPublicObjectLock::RTRPublicObjectLock ([RTRManagedObject](#) *parent = 0) [inline]

Used to synchronize the access of global managed object directory. Parent object is present when constructing a non-root object where its parent also needs to be locked in a multi-thread context

RTRPublicObjectLock::~RTRPublicObjectLock () [inline]

Destructor

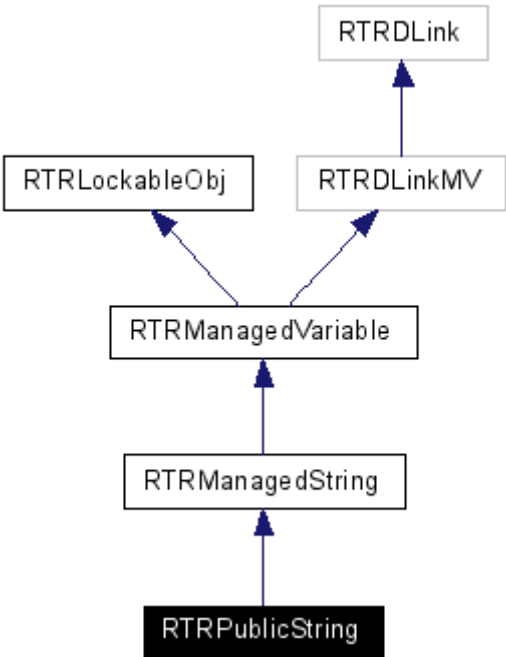
RTRPublicString Class Reference

An implementation of the [RTRManagedString](#) base class which provides set operations and uses the class [RTRMStringImpl](#) for storage allocation.

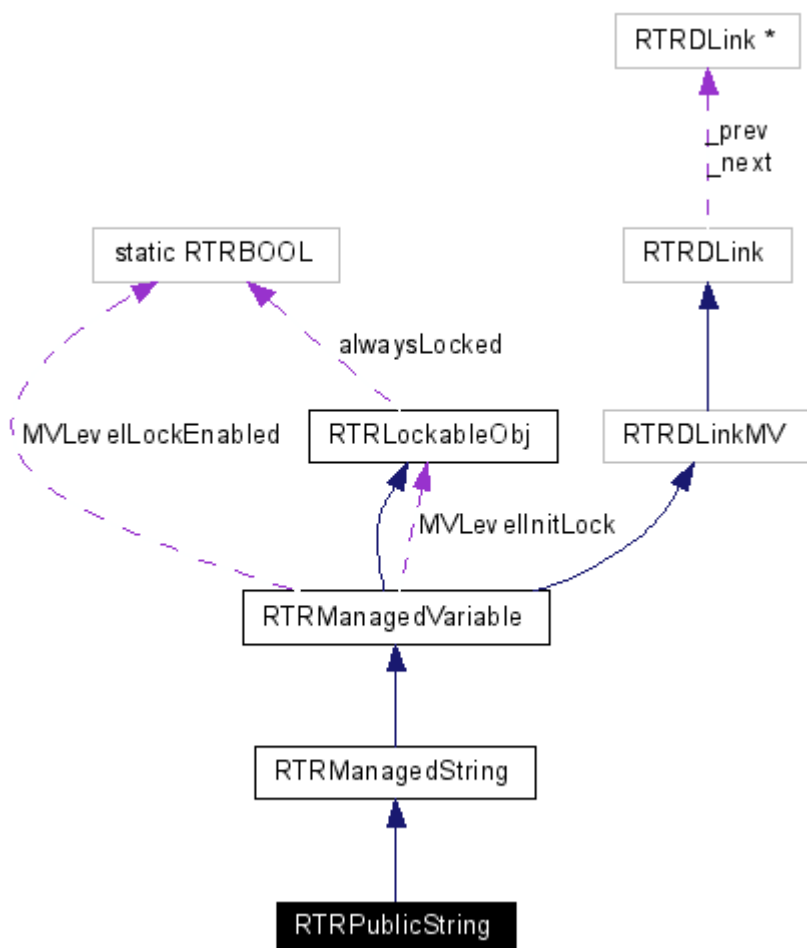
```
#include <pmstrv.h>
```

Inherits [RTRManagedString](#).

Inheritance diagram for RTRPublicString:



Collaboration diagram for `RTRPublicString`:



Public Member Functions

- [RTRPublicString](#) ([RTRPublicObject](#) &context, const char *name, const char *description, const char *initValue, RTRBOOL modifyEnabled=RTRFALSE)
- virtual [~RTRPublicString](#) ()
- [RTRPublicString](#) & [operator=](#) (const char *rhs)
- void [set](#) (const char *newValue)
- void [internalSet](#) (const char *newValue)

Detailed Description

An implementation of the [RTRManagedString](#) base class which provides set operations and uses the class [RTRMStringImpl](#) for storage allocation.

If modifyEnabled is true (default value is false), then consumers will be permitted to modify the active value.

See Also:

[RTRPublicObject](#), [RTRPublicNumeric](#), [RTRPublicBoolean](#), [RTRPublicCounter](#), [RTRPublicGauge](#), [RTRPublicNumericRange](#), [RTRPublicStringConfig](#), [RTRPublicNumericConfig](#), [RTRPublicGaugeConfig](#), [RTRPublicBooleanConfig](#),

Constructor & Destructor Documentation

RTRPublicString::RTRPublicString ([RTRPublicObject](#) & *context*, const char * *name*, const char * *description*, const char * *initValue*, RTRBOOL *modifyEnabled* = RTRFALSE)

Constructs an RTRPublicString variable ENSURE: [value\(\)](#) == *initValue*

virtual RTRPublicString::~~RTRPublicString () [virtual]

Destructor

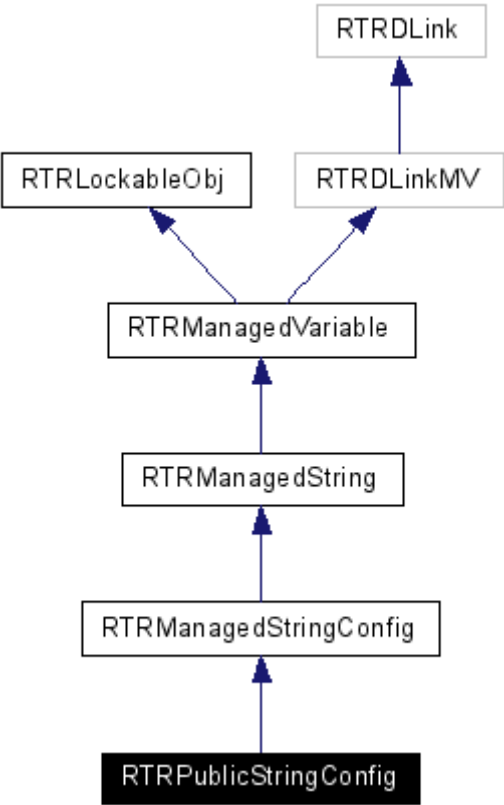
RTRPublicStringConfig Class Reference

An implementation of the [RTRManagedStringConfig](#) base class which uses the class RTRMStrConfigImpl for storage allocation.

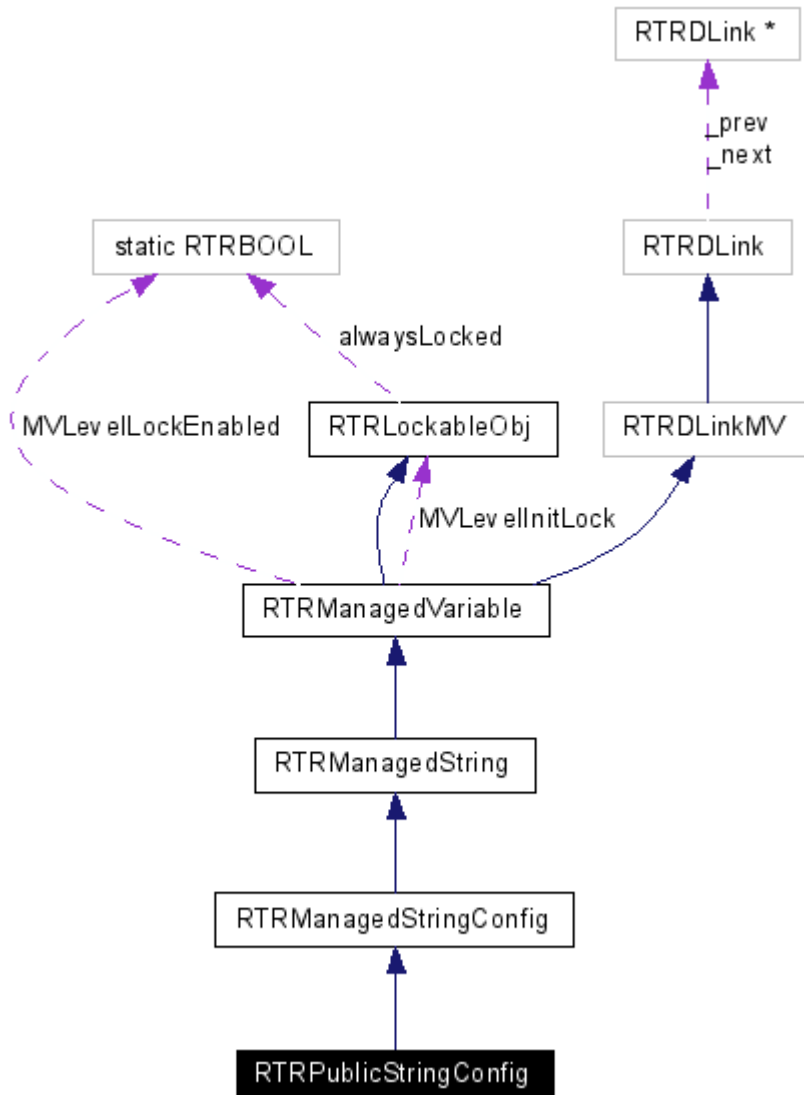
```
#include <pmstrcv.h>
```

Inherits [RTRManagedStringConfig](#).

Inheritance diagram for RTRPublicStringConfig:



Collaboration diagram for RTRPublicStringConfig:



Public Member Functions

- [RTRPublicStringConfig](#) ([RTRPublicObject](#) &context, const char *name, const char *description, const char *dfltValue, RTRBOOL modifyEnabled=RTRFALSE)
- virtual [~RTRPublicStringConfig](#) ()
- [RTRPublicStringConfig](#) & [operator=](#) (const char *rhs)
- virtual void [set](#) (const char *newValue)
- virtual void [internalSet](#) (const char *newValue)
- void [setStore](#) (const char *newStore)

Detailed Description

An implementation of the [RTRManagedStringConfig](#) base class which uses the class [RTRMStrConfigImpl](#) for storage allocation.

If modifyEnabled is true (default value is false), then consumers will be permitted to modify the active value.

See Also:

[RTRPublicObject](#), [RTRPublicNumeric](#), [RTRPublicString](#), [RTRPublicGauge](#), [RTRPublicCounter](#) [RTRPublicBoolean](#), [RTRPublicNumericRange](#), [RTRPublicNumericConfig](#), [RTRPublicGaugeConfig](#), [RTRPublicBooleanConfig](#)

Constructor & Destructor Documentation

RTRPublicStringConfig::RTRPublicStringConfig ([RTRPublicObject](#) & *context*, const char * *name*, const char * *description*, const char * *dfltValue*, RTRBOOL *modifyEnabled* = RTRFALSE)

Constructs an RTRPublicStringConfig variable.

ENSURE: [activeValue\(\)](#) == [storeValue\(\)](#) == [factoryDefault\(\)](#) == dfltValue

virtual RTRPublicStringConfig::~RTRPublicStringConfig () [virtual]

Destructor

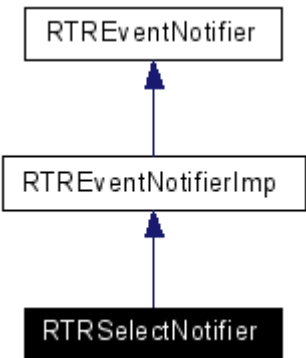
RTRSelectNotifier Class Reference

This implementation of [RTREventNotifierImp](#) implements a main loop based on the select() system call.

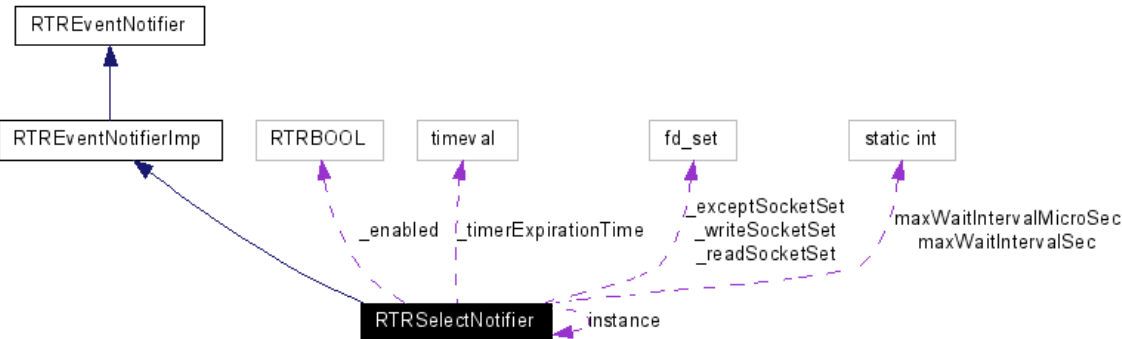
```
#include <selectni.h>
```

Inherits [RTREventNotifierImp](#).

Inheritance diagram for RTRSelectNotifier:



Collaboration diagram for RTRSelectNotifier:



Public Member Functions

- [RTRSelectNotifier](#) ()

- [~RTRSelectNotifier \(\)](#)
- void [enable \(\)](#)
- void [disable \(\)](#)
- void [enableTimer](#) (long seconds, int milliseconds)
- void [disableTimer \(\)](#)
- void enableReadNotification (int fd)
- void disableReadNotification (int fd)
- void enableWriteNotification (int fd)
- void disableWriteNotification (int fd)
- void enableExceptNotification (int fd)
- void disableExceptNotification (int fd)

Static Public Member Functions

- static void [run \(\)](#)

Static Public Attributes

- static int [maxWaitIntervalSec](#)
- static int [maxWaitIntervalMicroSec](#)

Friends

- class RTREventNotifierInit

Detailed Description

This implementation of [RTREventNotifierImp](#) implements a main loop based on the select() system call.

[RTREventNotifierImp](#) maintains a record of all requested timers, only one of which is outstanding at a given time. When a timer event is requested, [RTREventNotifierImp](#) will determine what the next required timer interval is and invoke [enableTimer\(\)](#). This method is implemented here to utilize the select() system call to call-back at the requested interval, unless some I/O event occurs first.

[RTREventNotifierImp](#) requests I/O notification via calls to enable/disableReadNotification(), etc. These routines are implemented here to use the system select() call to register for I/O events.

```
RTRSelectNotifier::run();
```

See Also:

[RTREventNotifier](#), [RTREventNotifierImp](#), [RTRWindowsNotifier](#), [RTRXtNotifier](#), [RTRXViewNotifier](#)

Constructor & Destructor Documentation

RTRSelectNotifier::RTRSelectNotifier ()
 Construtor

RTRSelectNotifier::~~RTRSelectNotifier ()
 Destrutor

Member Function Documentation

static void RTRSelectNotifier::run () [static]

Main Loop

Member Data Documentation

- int

[RTRSelectNotifier::maxWaitIntervalSec](#)

[static]

The maximum blocking seconds on select when there is no interested IO happening. This is to ensure that in case other threads register IO/timer interests, they should get response in a limited time.
- int

[RTRSelectNotifier::maxWaitIntervalMicroSec](#)

[static]

The maximum blocking microseconds on select when there is no interested IO happening The actual time should be the sum of the above two.

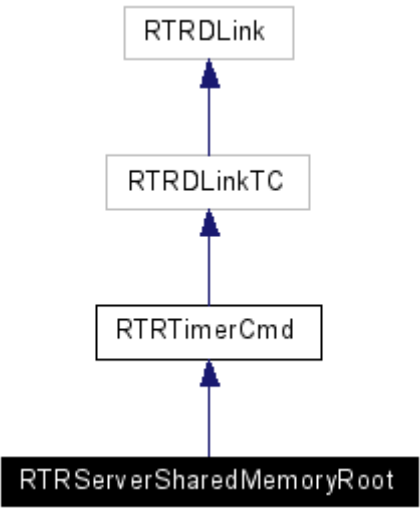
RTRServerSharedMemoryRoot Class Reference

The encapsulation of the server side of a server/client shared memory relationship. An instance of RTRServerSharedMemoryRoot is constructed with a key and will then attempt to allocate the shared memory using with that key. If memory already exists with that key then the memory server will examine that memory to determine whether or not it can safely be reinitialized. If the memory header matches that which the server would create (version, size etc) and the memory appears to no longer be in use, then the server will re-initialize the existing memory. If the memory could be used but has not yet timed-out (based on data extracting from the existing memory) then the server will periodically retry the allocation process.

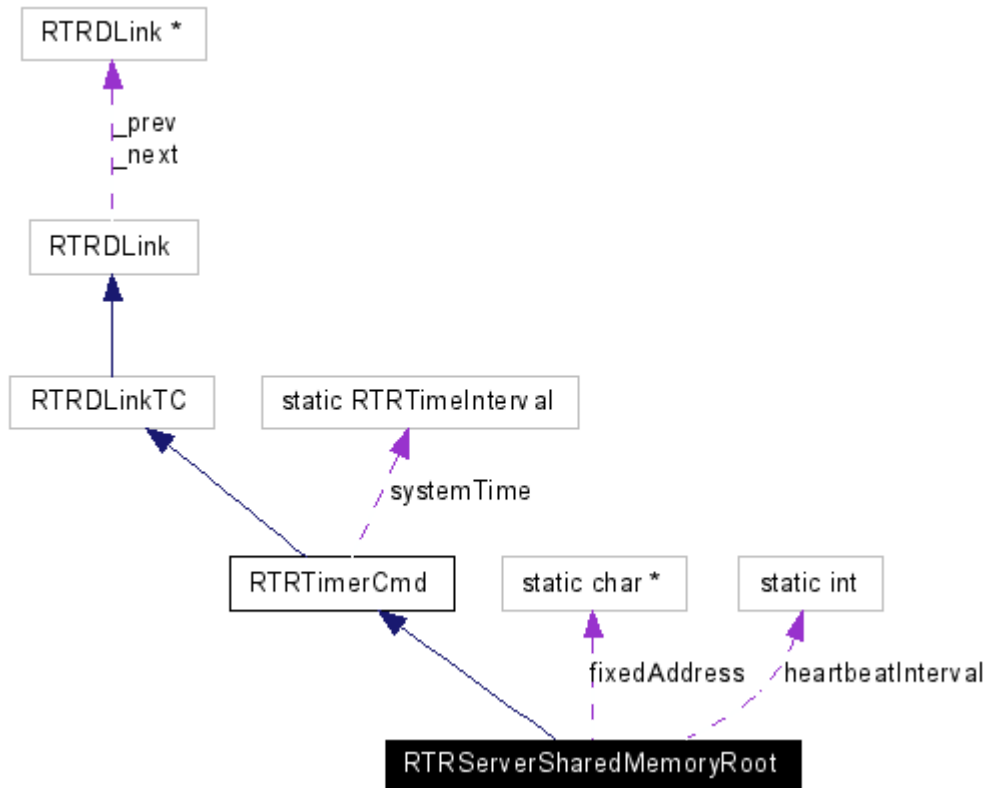
```
#include <shrdmem.h>
```

Inherits [RTRTimerCmd](#).

Inheritance diagram for RTRServerSharedMemoryRoot:



Collaboration diagram for RTRServerSharedMemoryRoot:



Public Member Functions

- [RTRServerSharedMemoryRoot](#) (const [RTRObjectId](#) &context, const char *name, const [RTRString](#) &mk, const [RTRString](#) &sk, int s, unsigned int long, int m)
- [~RTRServerSharedMemoryRoot](#) ()
- const [RTRObjectId](#) & `instanceId` () const
- const [RTRString](#) & `text` () const
- [RTRString](#) & `key` ()
- `HANDLE id` () const
- `RTRSharedMemoryHdr * header` () const
- `RTRServerSemaphoreSet * semaphoreSet` () const
- virtual `RTRBOOL error` () const
- `RTRSharedMemoryPartitionIterator partitionIterator` () const
- void `processTimerEvent` ()

Static Public Attributes

- static int [heartbeatInterval](#)
- static char * [fixedAddress](#)

Friends

- `std::ostream & operator<< (std::ostream &, const RTRServerSharedMemoryRoot &)`

Detailed Description

The encapsulation of the server side of a server/client shared memory relationship. An instance of `RTRServerSharedMemoryRoot` is constructed with a key and will then attempt to allocate the shared memory using with that key. If memory already exists with that key then the memory server will examine that memory to determine whether or not it can safely be reinitialized. If the memory header matches that which the server would create (version, size etc) and the memory appears to no longer be in use, then the server will re-initialize the existing memory. If the memory could be used but has not yet timed-out (based on data extracting from the existing memory) then the server will periodically retry the allocation process.

Once memory has been successfully allocated or re-initialized then the server will implement a handshaking scheme with any clients which attach to the memory segment.

Options:

By setting the public static data member "fixedAddress" prior to construction the mapping of the shared memory into the process address space can be controlled. It is up to the caller to ensure that the given address is appropriate.

See Also:

`RTRSharedMemoryHdr`, `RTRServerSemaphoreSet`

Constructor & Destructor Documentation

`RTRServerSharedMemoryRoot::RTRServerSharedMemoryRoot (const RTRObjectId & context, const char * name, const RTRString & mk, const RTRString & sk, int s, unsigned int long, int m)`

Allocate sh. mem so that s bytes are available for use.

`RTRServerSharedMemoryRoot::~~RTRServerSharedMemoryRoot ()`

Destructor

Member Data Documentation

`int RTRServerSharedMemoryRoot::heartbeatInterval [static]`

The interval at which client handshaking will occur.

`char* RTRServerSharedMemoryRoot::fixedAddress [static]`

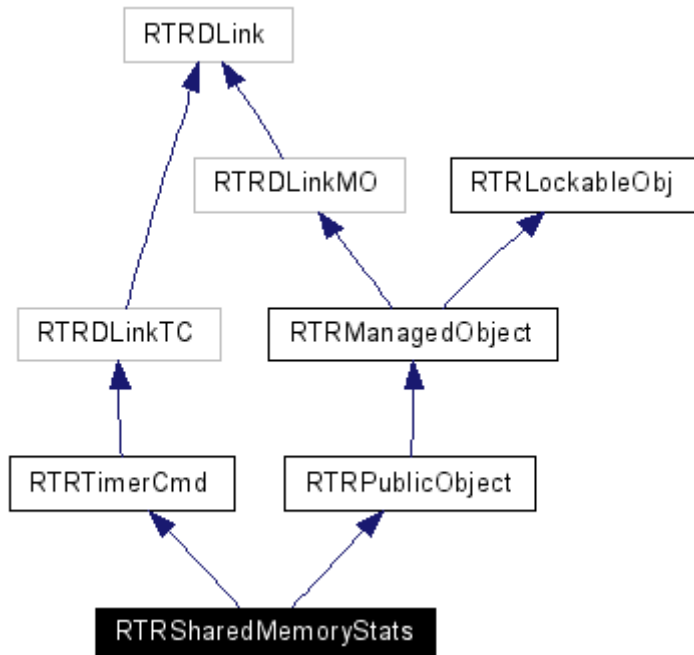
Defaults to 0, meaning that system will assign the mapping address for you. Modify this prior to construction to control the mapping yourself.

RTRSharedMemoryStats Class Reference

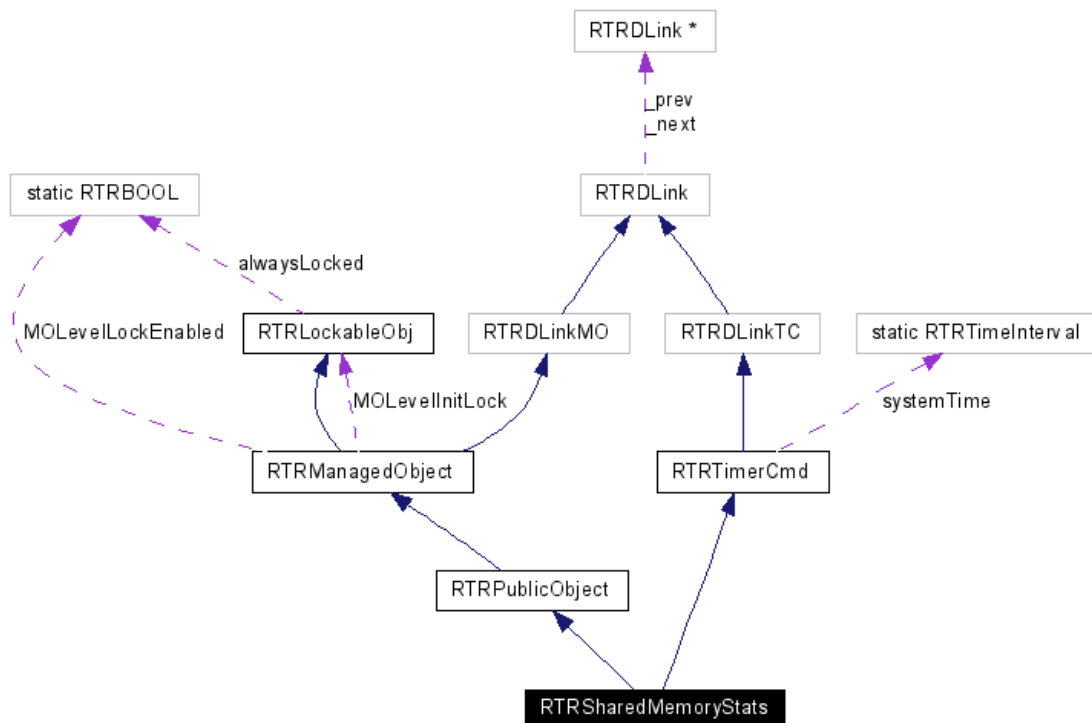
```
#include <shmstats.h>
```

Inherits [RTRTimerCmd](#), and [RTRPublicObject](#).

Inheritance diagram for `RTRSharedMemoryStats`:



Collaboration diagram for RTRSharedMemoryStats:



Public Member Functions

- [RTRSharedMemoryStats](#) (const [RTRObjectId](#) &cntxt, const [RTRString](#) &nm, RTRSharedMemoryHdr &hdr, RTRBOOL publishItself, int upd_intvl=5)
- [RTRSharedMemoryStats](#) ([RTRPublicObject](#) &parent, const [RTRString](#) &nm, RTRSharedMemoryHdr &hdr, RTRBOOL publishItself, int upd_intvl=5)
- [~RTRSharedMemoryStats](#) ()
- void **setServer** ([RTRShmMOServerMemPool](#) *s)
- void **update** ()
- virtual void [processTimerEvent](#) ()

Detailed Description

See Also:

[RTRShmMOServerMemPool](#)

Constructor & Destructor Documentation

RTRSharedMemoryStats::RTRSharedMemoryStats (const [RTRObjectId](#) & cntxt, const [RTRString](#) & nm, RTRSharedMemoryHdr & hdr, RTRBOOL *publishItself*, int *upd_intvl* = 5)
 Constructor

RTRSharedMemoryStats::RTRSharedMemoryStats ([RTRPublicObject](#) & parent, const [RTRString](#) & nm, RTRSharedMemoryHdr & hdr, RTRBOOL *publishItself*, int *upd_intvl* = 5)
 Constructor

RTRSharedMemoryStats::~~RTRSharedMemoryStats ()
 Destructor

Member Function Documentation

virtual void RTRSharedMemoryStats::processTimerEvent () [virtual]
 Redefined by descendants to provide specific behaviour for this timer.

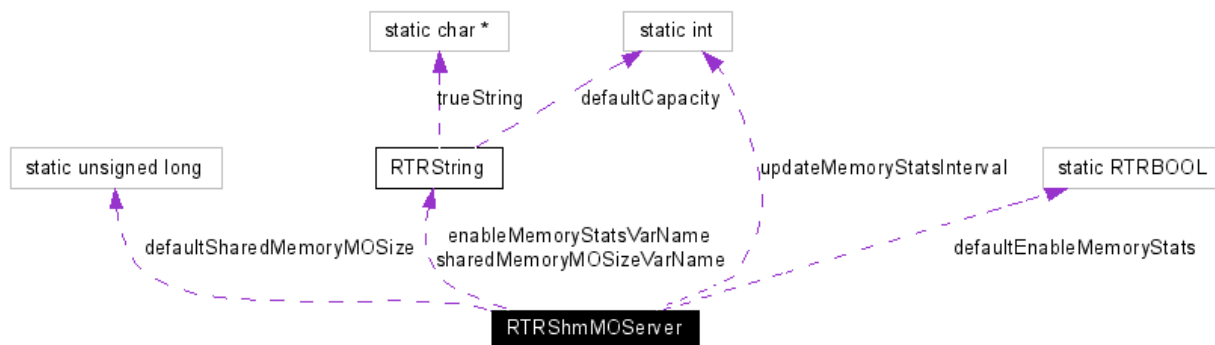
Implements [RTRTimerCmd](#).

RTRShmMOServer Class Reference

RTRShmMOServer is a helper class which instantiates an instance of [RTRServerSharedMemoryRoot](#) (and conditionally [RTRSharedMemoryStats](#)) as indicated by either the config db or information passed in on the constructor.

```
#include <shmosrvr.h>
```

Collaboration diagram for RTRShmMOServer:



Public Member Functions

- [RTRShmMOServer](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &context, const char *name, [RTRShmServer](#) &shm, `RTRBOOL enable=RTRTRUE`)
- [RTRShmMOServer](#) (const [RTRObjectId](#) &context, const char *name, [RTRShmServer](#) &shm, unsigned long sharedMemorySize, `RTRBOOL enable=RTRTRUE`, `RTRBOOL enableStats=RTRFALSE`)
- [~RTRShmMOServer](#) ()
- `RTRBOOL enabled () const`
- `RTRBOOL error () const`
- const [RTRObjectId](#) & `instanceId` () const
- unsigned long [sharedMemorySize](#) () const
- int [maxClients](#) () const
- const [RTRString](#) & `text` () const
- [RTRShmMOServerMemPool](#) * [managedObjectServer](#) () const
- [RTRSharedMemoryStats](#) * `memoryStats` () const
- void [enable](#) ()
- void [disable](#) ()

Static Public Attributes

- static unsigned long `defaultSharedMemoryMOSize`
- static [RTRString](#) `sharedMemoryMOSizeVarName`
- static [RTRString](#) `enableMemoryStatsVarName`
- static int `updateMemoryStatsInterval`
- static `RTRBOOL` `defaultEnableMemoryStats`

Detailed Description

`RTRShmMOServer` is a helper class which instantiates an instance of [RTRServerSharedMemoryRoot](#) (and conditionally [RTRSharedMemoryStats](#)) as indicated by either the config db or information passed in on the constructor.

`RTRShmMOServer` utilizes a portion of the shared memory segment created by [RTRShmServer](#) for storing managed objects.

See Also:

[RTRServerSharedMemoryRoot](#), [RTRSharedMemoryStats](#), [RTRShmServer](#)

Constructor & Destructor Documentation

RTRShmMOServer::RTRShmMOServer (const [RTRObjectld](#) & *classId*, const [RTRObjectld](#) & *context*, const char * *name*, [RTRShmServer](#) & *shm*, RTRBOOL *enable* = RTRTRUE)
 Use the RTRConfigDb::configDb for configuration.

REQUIRE: !shm.[error\(\)](#)

RTRShmMOServer::RTRShmMOServer (const [RTRObjectld](#) & *context*, const char * *name*, [RTRShmServer](#) & *shm*, unsigned long *sharedMemorySize*, RTRBOOL *enable* = RTRTRUE, RTRBOOL *enableStats* = RTRFALSE)
 Use the constructor arguments for configuring the managed object server.

REQUIRE: !shm.[error\(\)](#)

RTRShmMOServer::~~RTRShmMOServer ()
 Destructor

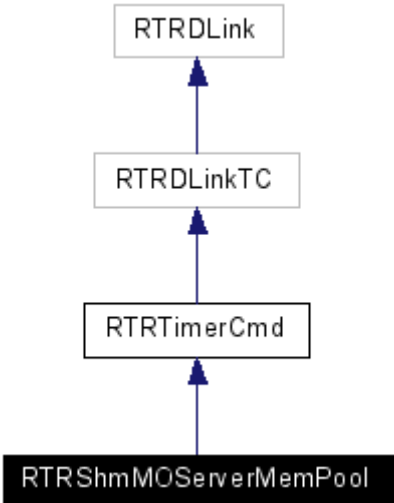
RTRShmMOServerMemPool Class Reference

RTRShmMOServerMemPool is an implementation of the abstract base class RTRMOServerMemPool which uses shared memory to allocate storage for managed objects and variables allocated by the application (server).

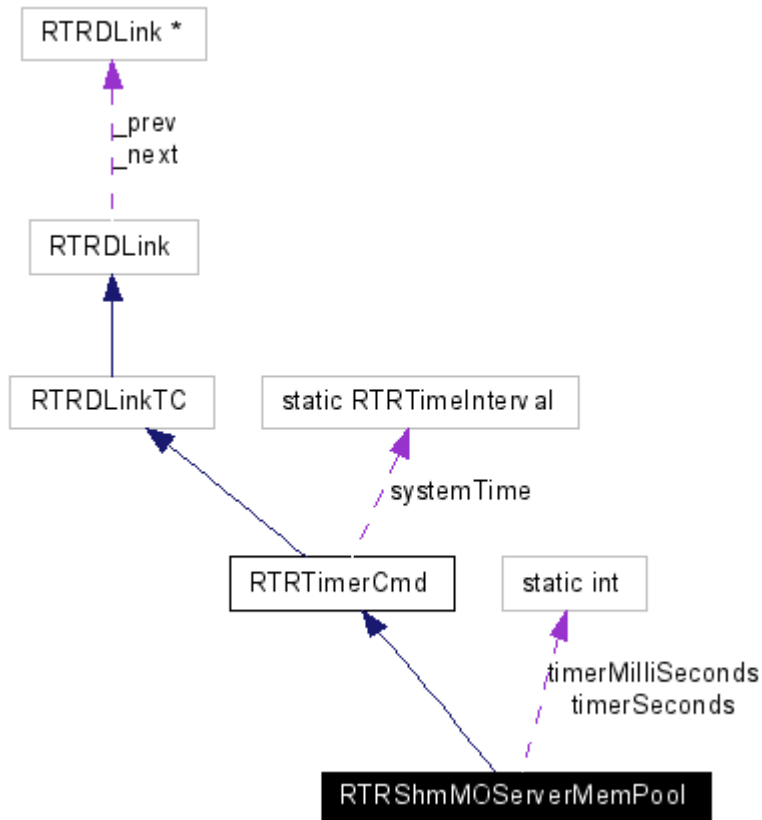
```
#include <shmmosvr.h>
```

Inherits [RTRTimerCmd](#).

Inheritance diagram for RTRShmMOServerMemPool:



Collaboration diagram for RTRShmMOServerMemPool:



Public Member Functions

- [RTRShmMOServerMemPool](#) ([RTRServerSharedMemoryRoot](#) &, [RTRServerSemaphoreSet](#) &, int maxClients, unsigned long size)
- [~RTRShmMOServerMemPool](#) ()
- [RTRBOOL error](#) () const
- const [RTRString](#) & [text](#) () const
- [RTRServerPartition](#) & [partition](#) ()
- void [useStats](#) ([RTRSharedMemoryStats](#) *)
- void [processTimerEvent](#) ()
- void [pollForMessages](#) ()

Static Public Attributes

- static int [timerSeconds](#)
- static int [timerMilliseconds](#)

Friends

- class [RTRSharedMemoryStats](#)
- std::ostream & [operator<<](#) (std::ostream &, const [RTRShmMOServerMemPool](#) &)

Detailed Description

RTRShmMOServerMemPool is an implementation of the abstract base class RTRMOServerMemPool which uses shared memory to allocate storage for managed objects and variables allocated by the application (server).

The object structure defined by the application is reproduced in shared memory and used by clients (instances of RTRShmProxyMgr) to clone managed objects and variables in the client process.

The memory management scheme is pseudo dynamic. Shared memory storage is allocated dynamically (based on demand from the application) as storage for either an object or some type of variable. Once allocated for a particular purpose, it will never be freed for general purpose use. When storage is freed it is placed on a free list with other storage of the same type (same size and layout).

In general, modification of shared memory must be synchronized with access with other processes. Allocated objects and variables cannot be attached to or removed from the existing shared memory layout without acquiring a lock. For efficiency, allocation is done on demand but attach and remove operations are done in batches. Allocation does not need to be synchronized because there is only one reader and writer of the free lists, the server. Lists of objects and variables which need to be attached or removed are maintained by the server and serviced at regular intervals. This means that the server does not have to obtain a semaphore lock every time an object is allocated or deleted.

If no memory (or no memory of the requested type) is available the server returns a null pointer to the caller (usually an instance of an object or variable implementation class). It is assumed that the caller will detect this and allocate memory from the heap as necessary.

The server also detects and processes parameter operations requested by clients.

If a fatal error is encountered during initialization, `error()` will be true and `text()` will contain an explanation of the problem.

See Also:

RTRShmProxyMgr, RTRServerPartition, RTRServerSemaphoreSet

Constructor & Destructor Documentation

RTRShmMOServerMemPool::RTRShmMOServerMemPool ([RTRServerSharedMemoryRoot](#) &, RTRServerSemaphoreSet &, int *maxClients*, unsigned long *size*)
 Constructor

RTRShmMOServerMemPool::~~RTRShmMOServerMemPool ()
 Destructor

Member Data Documentation

int [RTRShmMOServerMemPool::timerSeconds](#) [static]
 default 1

int [RTRShmMOServerMemPool::timerMilliseconds](#) [static]
 default 0

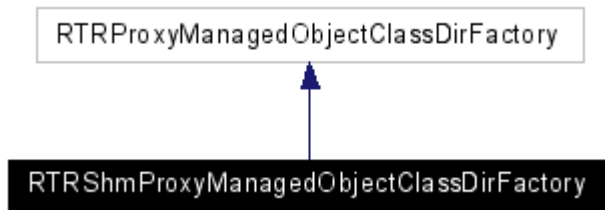
RTRShmProxyManagedObjectClassDirFactory Class Reference

A utility class used to obtain instances of [RTRProxyManagedObjectClassDirectory](#).

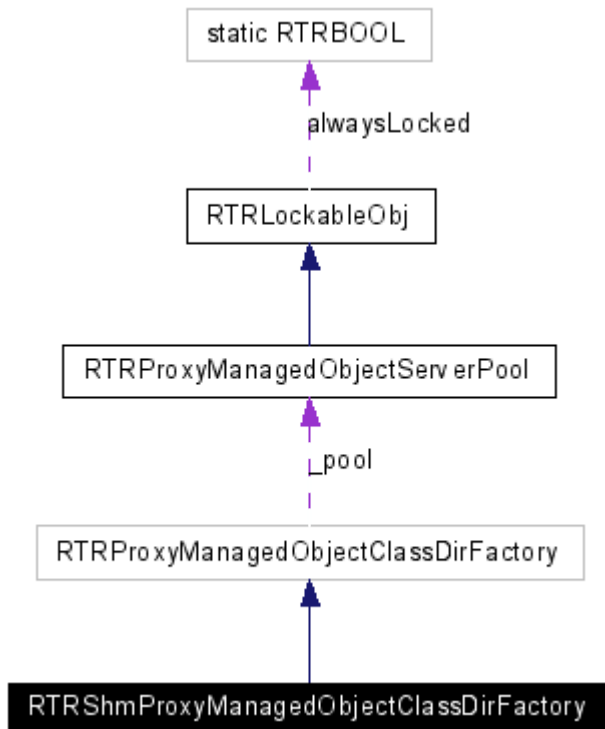
```
#include <shmpmocdf.h>
```

Inherits RTRProxyManagedObjectClassDirFactory.

Inheritance diagram for RTRShmProxyManagedObjectClassDirFactory:



Collaboration diagram for RTRShmProxyManagedObjectClassDirFactory:



Public Member Functions

- **RTRShmProxyManagedObjectClassDirFactory** (**RTRShmProxyManagedObjectServerPool** &)
- virtual [~RTRShmProxyManagedObjectClassDirFactory](#) ()
- **RTRProxyManagedObjectClassDirectoryPtr newClassDirectory** (const [RTRObjectId](#) &classFilter) const

Detailed Description

A utility class used to obtain instances of [RTRProxyManagedObjectClassDirectory](#).

See Also:

[RTRProxyManagedObjectClassDirectory](#)

Constructor & Destructor Documentation

RTRShmProxyManagedObjectClassDirFactory::RTRShmProxyManagedObjectClassDirFactory
([RTRShmProxyManagedObjectServerPool](#) &)

Constructor

```
virtual RTRShmProxyManagedObjectClassDirFactory::~RTRShmProxyManagedObjectClassDirFactory () [virtual]
Destructor
```

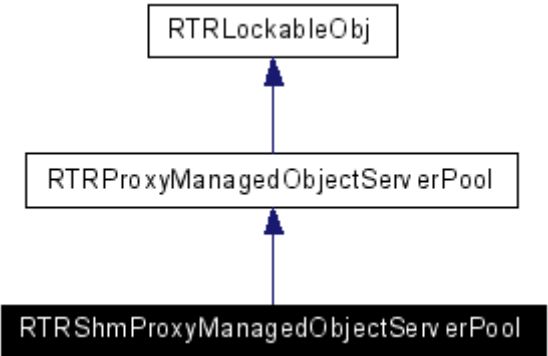
RTRShmProxyManagedObjectServerPool Class Reference

A shared memory based implementation of a [RTRProxyManagedObjectServerPool](#).

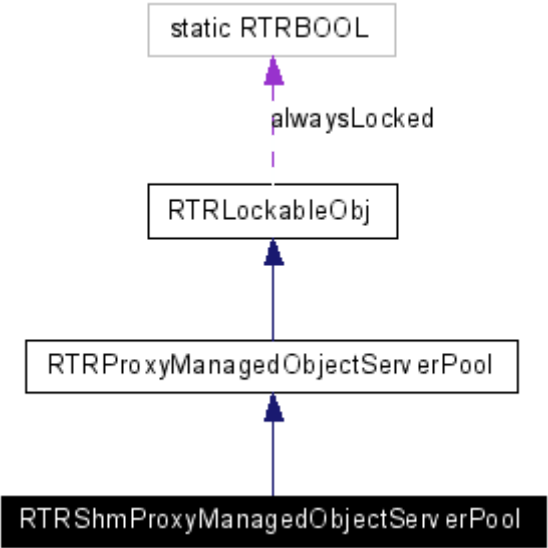
```
#include <shmpmosp.h>
```

Inherits [RTRProxyManagedObjectServerPool](#).

Inheritance diagram for RTRShmProxyManagedObjectServerPool:



Collaboration diagram for RTRShmProxyManagedObjectServerPool:



Public Member Functions

- [RTRShmProxyManagedObjectServerPool](#) (const [RTRObjectId](#) &context, const char *name)
- virtual [~RTRShmProxyManagedObjectServerPool](#) ()
- RTRShmProxyManagedObjectServer * [addServer](#) (const char *key, int pollInterval=1, int handshakeInterval=2)

- void [dropServer](#) (const char *key)

Detailed Description

A shared memory based implementation of a [RTRProxyManagedObjectServerPool](#).

See Also:

RTRShmProxyManagedObjectServer

Constructor & Destructor Documentation

RTRShmProxyManagedObjectServerPool::RTRShmProxyManagedObjectServerPool (const [RTRObjectId](#) & context, const char * name)

Constructor

virtual RTRShmProxyManagedObjectServerPool::~~RTRShmProxyManagedObjectServerPool () [virtual]

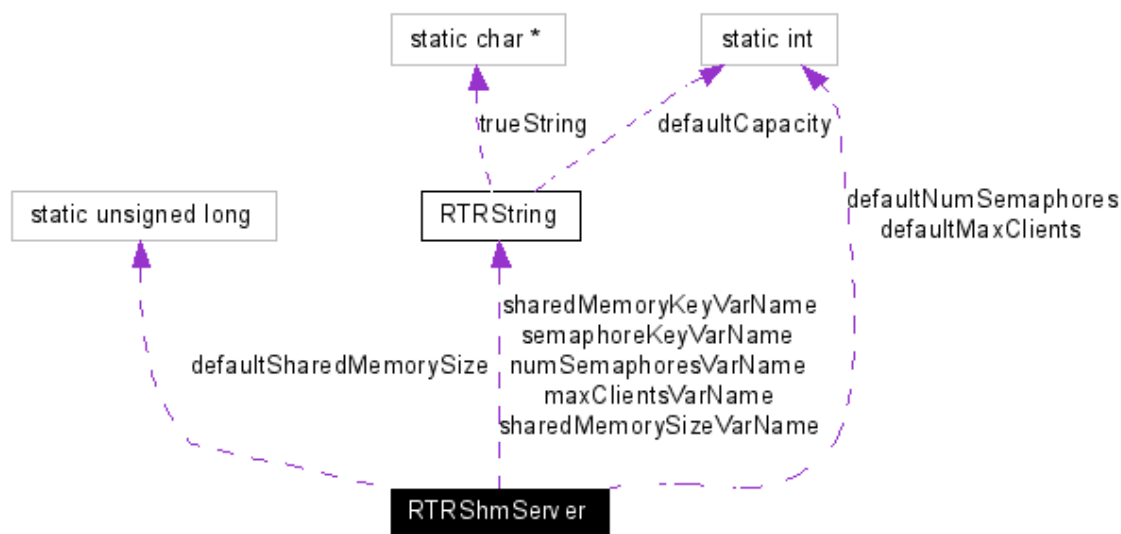
Destructor

RTRShmServer Class Reference

RTRShmServer is a helper class which instantiates an [RTRServerSharedMemoryRoot](#). The configuration of the [RTRServerSharedMemoryRoot](#) is obtained from either a config file or passed in from the constructor (with a minimal number of required arguments).

```
#include <shmsrvr.h>
```

Collaboration diagram for RTRShmServer:



Public Member Functions

- [RTRShmServer](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &context, const char *name, RTRBOOL enable=RTRTRUE)
- [RTRShmServer](#) (const [RTRObjectId](#) &context, const char *name, unsigned long sharedMemorySize, const char *sharedMemoryKey, const char *semaphoreKey, RTRBOOL enable=RTRTRUE, int maxClients=10, int numSemaphores=8)
- [~RTRShmServer](#) ()
- RTRBOOL [enabled](#) () const
- RTRBOOL [error](#) () const

- const [RTRObjectld](#) & [instanceld](#) () const
- const [RTRString](#) & [sharedMemoryKey](#) () const
- const [RTRString](#) & [semaphoreKey](#) () const
- unsigned long [sharedMemorySize](#) () const
- int [maxClients](#) () const
- int [numberOfSemaphores](#) () const
- const [RTRString](#) & [text](#) () const
- [RTRServerSharedMemoryRoot](#) * [sharedMemory](#) () const
- void [enable](#) ()
- void [disable](#) ()

Static Public Attributes

- static unsigned long defaultSharedMemorySize
- static int defaultMaxClients
- static int defaultNumSemaphores
- static [RTRString](#) [sharedMemoryKeyVarName](#)
- static [RTRString](#) [sharedMemorySizeVarName](#)
- static [RTRString](#) [semaphoreKeyVarName](#)
- static [RTRString](#) [maxClientsVarName](#)
- static [RTRString](#) [numSemaphoresVarName](#)

Detailed Description

RTRShmServer is a helper class which instantiates an [RTRServerSharedMemoryRoot](#). The configuration of the [RTRServerSharedMemoryRoot](#) is obtained from either a config file or passed in from the constructor (with a minimal number of required arguments).

See Also:

[RTRServerSharedMemoryRoot](#)

Constructor & Destructor Documentation

RTRShmServer::RTRShmServer (const [RTRObjectld](#) & *classld*, const [RTRObjectld](#) & *context*, const char * *name*, RTRBOOL *enable* = RTRTRUE)

Use [RTRConfig::configDb\(\)](#).

RTRShmServer::RTRShmServer (const [RTRObjectld](#) & *context*, const char * *name*, unsigned long *sharedMemorySize*, const char * *sharedMemoryKey*, const char * *semaphoreKey*, RTRBOOL *enable* = RTRTRUE, int *maxClients* = 10, int *numSemaphores* = 8)

Use arguments provided.([RTRConfigDb::configDb\(\)](#) is not used).

RTRShmServer::~~RTRShmServer ()

Destructor

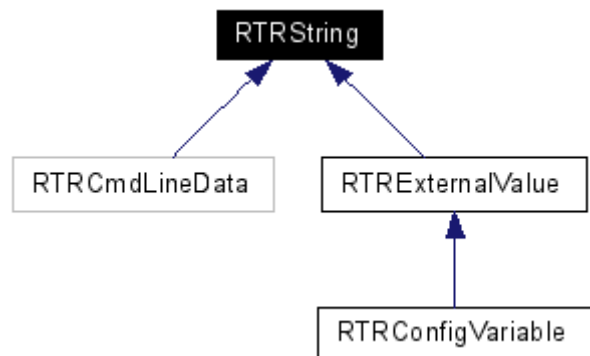
RTRString Class Reference

A representation for a sequence of characters. The sequence may contain embedded null characters.

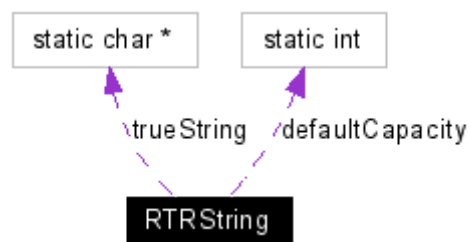
```
#include <rtstring.h>
```

Inherited by `RTRCmdLineData`, and [RTRExternalValue](#).

Inheritance diagram for `RTRString`:



Collaboration diagram for `RTRString`:



Public Member Functions

- [RTRString](#) ()
- [RTRString](#) (unsigned int n)
- [RTRString](#) (char c, unsigned int n)
- [RTRString](#) (const char *str)
- [RTRString](#) (const char *str, unsigned int n)
- [RTRString](#) (const [RTRString](#) &)
- [~RTRString](#) ()
- const unsigned int [capacity](#) () const
- unsigned int [count](#) () const
- RTRBOOL [isEmpty](#) () const
- unsigned long [hash](#) () const
- int [lower](#) () const
- int [upper](#) () const

- [RTRString](#) & [set](#) (const char *str, unsigned int p1, unsigned int p2)
- [RTRString](#) & [set](#) (const char *str, unsigned int n)
- [RTRString](#) & [readLine](#) (std::istream &, RTRBOOL skipWhite=1)
- [RTRString](#) & [clear](#) ()
- [RTRString](#) & [fromNumeric](#) (int i)
- [RTRString](#) & [fromNumeric](#) (unsigned int i)
- [RTRString](#) & [fromNumeric](#) (long i)
- [RTRString](#) & [fromNumeric](#) (unsigned long i)
- [RTRString](#) & [fromNumeric](#) (double i)
- char & [operator\[\]](#) (int i)
- [RTRString](#) & [prepend](#) (const char *)
- [RTRString](#) & [prepend](#) (char)
- [RTRString](#) & [prepend](#) (long)
- [RTRString](#) & [prepend](#) (unsigned long)
- [RTRString](#) & [prepend](#) (double)
- [RTRString](#) & [append](#) (const char *)
- [RTRString](#) & [append](#) (const char *, int)
- [RTRString](#) & [append](#) (const [RTRString](#) &)
- [RTRString](#) & [append](#) (const char)
- [RTRString](#) & [append](#) (const unsigned char)
- [RTRString](#) & [append](#) (const short n)
- [RTRString](#) & [append](#) (const unsigned short n)
- [RTRString](#) & [append](#) (const int n)
- [RTRString](#) & [append](#) (const unsigned int n)
- [RTRString](#) & [append](#) (const long n)
- [RTRString](#) & [append](#) (const unsigned long n)
- [RTRString](#) & [append](#) (const float n)
- [RTRString](#) & [append](#) (const double n)
- [RTRString](#) & [toLower](#) ()
- [RTRString](#) & [toUpper](#) ()
- void [leftAdjust](#) ()
- void [rightAdjust](#) ()
- [RTRString](#) & [head](#) (unsigned int n)
- [RTRString](#) & [tail](#) (unsigned int n)

- int [compare](#) (const char *) const
- char [operator\[\]](#) (int i) const
- [operator const char *](#) () const
- [RTRString subString](#) (int p1, int p2)
- const char * [to_c](#) () const
- RTRBOOL [contains](#) (const char *) const
- RTRBOOL [contains](#) (const char) const
- int [indexOf](#) (char c, int p1)
- int [toInteger](#) () const
- float [toFloat](#) () const
- double [toDouble](#) () const
- RTRBOOL [toBoolean](#) () const
- [RTRString](#) & [operator=](#) (const char *)
- [RTRString](#) & [operator=](#) (const [RTRString](#) &)
- RTRBOOL [operator==](#) (const char *) const
- RTRBOOL [operator==](#) (const [RTRString](#) &) const
- RTRBOOL [operator!=](#) (const char *) const
- RTRBOOL [operator!=](#) (const [RTRString](#) &) const
- RTRBOOL [operator>](#) (const char *) const
- RTRBOOL [operator>](#) (const [RTRString](#) &) const
- RTRBOOL [operator>=](#) (const char *) const
- RTRBOOL [operator>=](#) (const [RTRString](#) &) const
- RTRBOOL [operator<](#) (const char *) const
- RTRBOOL [operator<](#) (const [RTRString](#) &) const
- RTRBOOL [operator<=](#) (const char *) const
- RTRBOOL [operator<=](#) (const [RTRString](#) &) const
- [RTRString](#) & [operator+=](#) (const char *)
- [RTRString](#) & [operator+=](#) (const [RTRString](#) &)
- [RTRString](#) & [operator+=](#) (const char)
- void [grow](#) (unsigned int n)
- void [trim](#) (unsigned int)
- void [setCount](#) (unsigned int i)
- [RTRString](#) (const char *str, int n)
- RTRBOOL [isEqual](#) (const char *) const

- [RTRString](#) & [empty](#) ()
- int [length](#) () const
- int [index](#) (char c, int start)
- [RTRString](#) & [set](#) ([RTRString](#) &, unsigned int p1, unsigned int p2)
- [RTRString](#) & [fromInteger](#) (int i)
- void [appendNumeric](#) (const char)
- void [appendNumeric](#) (const unsigned char)
- void [appendNumeric](#) (const short n)
- void [appendNumeric](#) (const unsigned short n)
- void [appendNumeric](#) (const int n)
- void [appendNumeric](#) (const unsigned int n)
- void [appendNumeric](#) (const long n)
- void [appendNumeric](#) (const unsigned long n)
- void [appendNumeric](#) (const float n)
- void [appendNumeric](#) (const double n)

Static Public Attributes

- static int [defaultCapacity](#)
- static char * [trueString](#)

Friends

- std::ostream & [operator<<](#) (std::ostream &, const [RTRString](#) &)

Detailed Description

A representation for a sequence of characters. The sequence may contain embedded null characters.

See Also:

[RTRExternalValue](#), [RTRLISTOfExternalValue](#)

Constructor & Destructor Documentation

RTRString::RTRString ()

An empty string.

RTRString::RTRString (unsigned int *n*)

A string with capacity *n*.

RTRString::RTRString (char *c*, unsigned int *n*)

A string with capacity *n*, initialized with character *c*.

RTRString::RTRString (const char * *str*)

A string with a copy of the null terminated string *str*.

RTRString::RTRString (const char * *str*, unsigned int *n*)

A string with a copy of the *n* bytes from null terminated string *str*.

RTRString::RTRString (const [RTRString](#) &)

A string with a copy of the *n* bytes from null terminated string *str*.

RTRString::~~RTRString ()

Destructor

RTRString::RTRString (const char * *str*, int *n*)

A string with a copy of the *n* bytes from null terminated string *str*.

Compatibility of using "int" rather than "unsigned int"

Compatitbility - OBSOLETE

Member Data Documentation

int [RTRString::defaultCapacity](#) [static]

The size of area allocated when using the default constructor.

char* [RTRString::trueString](#) [static]

The value to which strings are compared when converting to a boolean.

RTRTimerCmd Class Reference

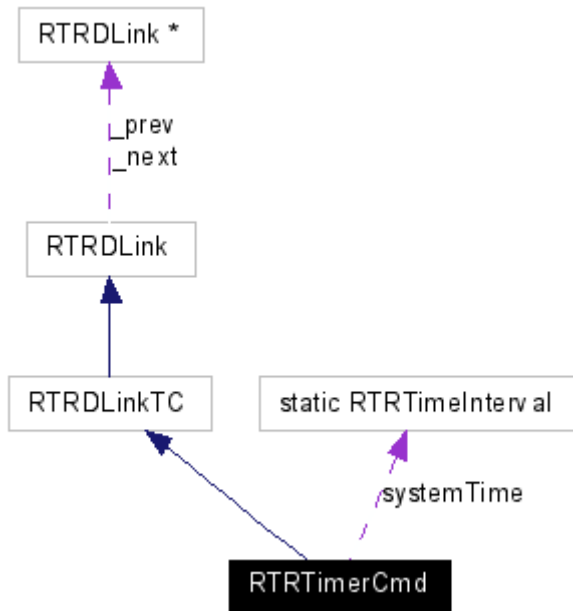
RTRTimerCmd is an abstract base class for components that will receive timer events.

```
#include <timercmd.h>
```

Inherits RTRDLinkTC.

Inherited by RTRBooleanConfigDisplay, RTRBooleanDisplay, RTRClientSharedMemoryRoot, RTRCounterDisplay, RTRGaugeConfigDisplay, RTRGaugeDisplay, RTRNumericConfigDisplay, RTRNumericConfigDisplay, RTRNumericDisplay, RTRNumericRangeDisplay, RTRProxyBooleanConfigDisplay, RTRProxyBooleanDisplay, RTRProxyCounterDisplay, RTRProxyGaugeConfigDisplay, RTRProxyGaugeDisplay, RTRProxyNumericConfigDisplay, RTRProxyNumericDisplay, RTRProxyNumericRangeDisplay, RTRProxyStringConfigDisplay, RTRProxyStringDisplay, [RTRServerSharedMemoryRoot](#), [RTRSharedMemoryStats](#), RTRShmCloneMgr, [RTRShmMOServerMemPool](#), RTRShmProxyManagedObjectDirectory, RTRShmProxyManagedObjectDirectory_EliF21, RTRShmProxyManagedObjectDirectory_Mdk10, RTRShmProxyMgr, RTRShmProxyMOSDirectory, RTRStringConfigDisplay, RTRStringDisplay, rvTimer, rvTimerGroupMgr [private], TIBTimerGroupMgr [private], and TIBTimerGroupMgr [private].

Collaboration diagram for RTRTimerCmd:



Public Member Functions

- [RTRTimerCmd](#) ([RTREventNotifier](#) *notifier=0)
- virtual [~RTRTimerCmd](#) ()
- const [RTRTimeInterval](#) & [timeOfEvent](#) () const
- long [offsetSeconds](#) () const
- short [offsetMilliseconds](#) () const
- [RTRBOOL](#) [operator==](#) ([RTRTimerCmd](#) &) const
- [RTRBOOL](#) [operator<](#) ([RTRTimerCmd](#) &) const
- [RTRBOOL](#) [operator<=](#) ([RTRTimerCmd](#) &) const
- [RTRBOOL](#) [operator>](#) ([RTRTimerCmd](#) &) const
- [RTRBOOL](#) [operator>=](#) ([RTRTimerCmd](#) &) const
- [RTRBOOL](#) [active](#) () const
- void [setTimerOffset](#) (long s, short m)
- void [activate](#) ()
- void [deactivate](#) ()
- virtual void [processTimerEvent](#) ()=0
- [RTRTimeInterval](#) & [eventTime](#) ()

Static Public Attributes

- static [RTRTimeInterval](#) **systemTime**

Friends

- `std::ostream & operator<< (std::ostream &os, RTRTimerCmd &tc)`

Detailed Description

RTRTimerCmd is an abstract base class for components that will receive timer events.

Descendants must define the method [processTimerEvent\(\)](#) to perform a specific timer related task.

The interval to be timed is set using [setTimerOffset\(\)](#). The timer must then be activated using [activate\(\)](#). An active timer may be deactivated. A timer will not automatically repeat. To implement a period timer, invoke [activate\(\)](#) in the implementation of [processTimerEvent\(\)](#). It isn't necessary to reset the offset, the most recently specified value will be used when activated.

See Also:

RTRTimerInterval

Constructor & Destructor Documentation

RTRTimerCmd::RTRTimerCmd ([RTREventNotifier](#) * *notifier* = 0)
 Constructor

virtual RTRTimerCmd::~~RTRTimerCmd () [virtual]
 Destructor

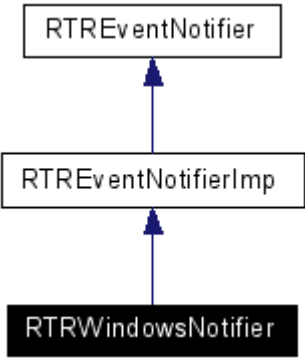
RTRWindowsNotifier Class Reference

This implementation of [RTREventNotifierImp](#) ([RTREventNotifier](#)) is based on the Windows library. The implementation allocates a window (WNDCLASS) which is used to register for I/O and timing events as needed.

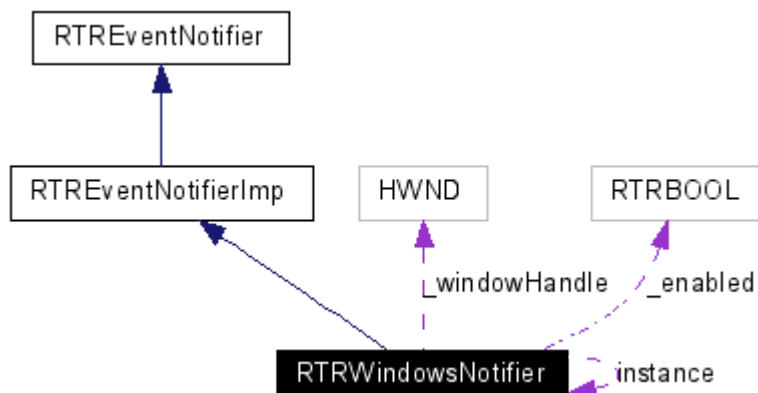
```
#include <winapi.h>
```

Inherits [RTREventNotifierImp](#).

Inheritance diagram for RTRWindowsNotifier:



Collaboration diagram for RTRWindowsNotifier:



Public Member Functions

- [RTRWindowsNotifier](#) ()
- [~RTRWindowsNotifier](#) ()
- void [enable](#) ()
- void [disable](#) ()
- void [enableTimer](#) (long seconds, int milliseconds)
- virtual void [disableTimer](#) ()
- void enableReadNotification (int fd)
- void disableReadNotification (int fd)
- void enableWriteNotification (int fd)
- void disableWriteNotification (int fd)
- void enableExceptNotification (int fd)
- void disableExceptNotification (int fd)
- HWND [windowHandle](#) () const
- RTRBOOL [enabled](#) () const

Static Public Attributes

- static [RTRWindowsNotifier](#) * [instance](#)

Friends

- class RTREventNotifierInit

Detailed Description

This implementation of [RTREventNotifierImp](#) ([RTREventNotifier](#)) is based on the Windows library. The implementation allocates a window (WNDCLASS) which is used to register for I/O and timing events as needed.

[RTREventNotifierImp](#) maintains a record of all requested timers, only one of which is outstanding at a given time. When a timer event is requested, [RTREventNotifierImp](#) will determine what the next required timer interval is and invoke [enableTimer\(\)](#). This method is implemented here to register with Windows for a call-back at the requested interval.

[RTREventNotifierImp](#) requests I/O notification via calls to enable/disableReadNotification(), etc. These routines are implemented here to use the WSAAsyncSelect() to register for Windows I/O events.

NOTE: windows.h (or afx.h if you are using MFC) must be included before this file

The explicit include of windows.h has been removed to make it easier for the user to choose whether windows.h or afx.h should be used.

```
//must first #include <windows.h> (or <afx.h> if using MFC)
#include "rtr/winni.h"
```

See Also:

[RTREventNotifier](#), [RTRSelectNotifier](#), [RTRXtNotifier](#), [RTRXViewNotifier](#)

Constructor & Destructor Documentation

RTRWindowsNotifier::RTRWindowsNotifier ()
Constructor

RTRWindowsNotifier::~~RTRWindowsNotifier ()
Destructor

Member Function Documentation

HWND RTRWindowsNotifier::windowHandle () const [inline]
Attributes

RTRBOOL RTRWindowsNotifier::enabled () const [inline]
State

Member Data Documentation

[RTRWindowsNotifier](#)* [RTRWindowsNotifier::instance](#) [static]
WindowsNotifier members

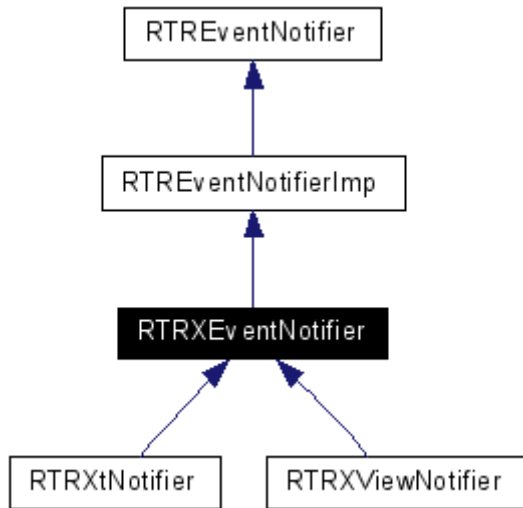
RTRXEventNotifier Class Reference

```
#include <xenimp.h>
```

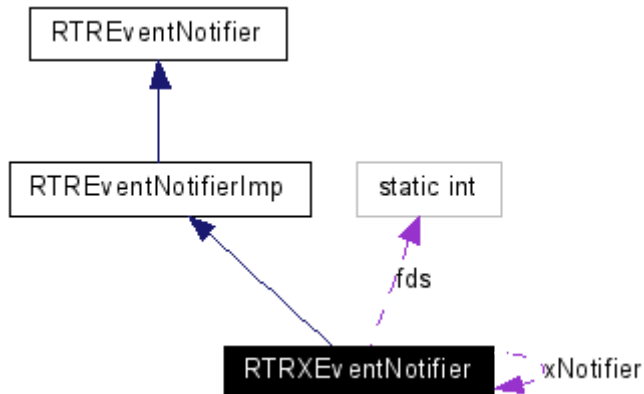
Inherits [RTREventNotifierImp](#).

Inherited by [RTRXtNotifier](#), and [RTRXViewNotifier](#).

Inheritance diagram for RTRXEventNotifier:



Collaboration diagram for RTRXEventNotifier:



Public Member Functions

- [RTRXEventNotifier](#) (int n)

Static Public Member Functions

- static int **initPipe** ()
- static int **pipeReadFd** ()
- static int **pipeWriteFd** ()

Static Public Attributes

- static [RTRXEventNotifier](#) * **xNotifier**
- static int **fds** [2]

Detailed Description

See Also:

[RTRXtNotifier](#), [RTRXViewNotifier](#)

Constructor & Destructor Documentation

RTRXEventNotifier::RTRXEventNotifier (int *n*)
 Constructor

Member Data Documentation

int [RTRXEventNotifier::fds](#)[2] [static]
 Array of 2 file descriptors for pipe.

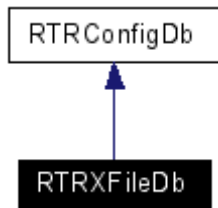
RTRXFileDb Class Reference

This descendant of a RTRFileConfigDb implements an "X" version of a file based configuration database. The X11 library configuration utilities are used to parse and maintain config variables retrieved from a disk file.

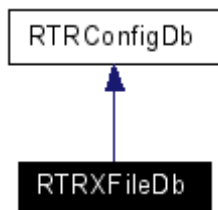
```
#include <xfdb.h>
```

Inherits [RTRConfigDb](#).

Inheritance diagram for RTRXFileDb:



Collaboration diagram for RTRXFileDb:



Public Member Functions

- [RTRXFileDb](#) ()
- [RTRXFileDb](#) (const char *p)
- [~RTRXFileDb](#) ()
- RTRBOOL [error](#) () const
- const char * [errorText](#) () const
- RTRBOOL [has](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &varName) const
- [RTRConfigVariable](#) [variable](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &varName, const [RTRString](#) &dflt) const
- [RTRConfigVariable](#) [variable](#) (const [RTRObjectId](#) &classId, const [RTRObjectId](#) &instanceId, const [RTRString](#) &varName) const

- [RTRConfigVariable value](#) (const [RTRObjectld](#) &classId, const [RTRObjectld](#) &instanceId, const [RTRString](#) &varName, const [RTRString](#) &dflt) const
- [RTRConfigVariable value](#) (const [RTRObjectld](#) &classId, const [RTRObjectld](#) &instanceId, const [RTRString](#) &varName) const
- void [load](#) (const char *fileName)

Detailed Description

This descendant of a RTRFileConfigDb implements an "X" version of a file based configuration database. The X11 library configuration utilities are used to parse and maintain config variables retrieved from a disk file.

Application components which use configuration variables have associated with them both a class identifier and an instance identifier. This allows system components to be configured (by means of variables) on a class basis and on a per instance basis. The precedence of the class identifiers relative to instance identifiers is based on the X11 configuration utility. (See the "Xlib Reference Manual - Volume Two" under "XrmGetResource" for details).

```
RTRXFileDb configDb("path_name");
```

See Also:

[RTRConfig](#)

Constructor & Destructor Documentation

RTRXFileDb::RTRXFileDb ()

Create an empty config database.

RTRXFileDb::RTRXFileDb (const char * p)

Create the config database using the pathname p to retrieve configuration info. Upon completion, the client should call the [error\(\)](#) feature.

RTRXFileDb::~~RTRXFileDb ()

Destructor

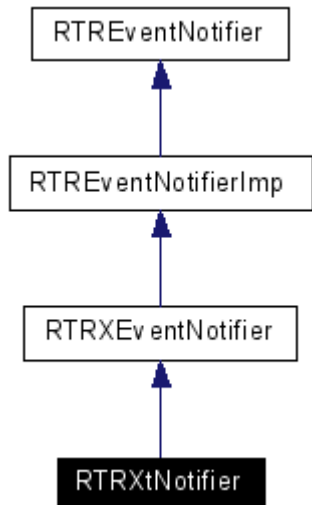
RTRXtNotifier Class Reference

An implementation of an [RTREventNotifierImp](#) ([RTREventNotifier](#)) based on the Xt library. The application must initialize the static class member appContext. It is of type XtAppContext. The initialization must occur before any methods of the notifier are invoked by any part of the system.

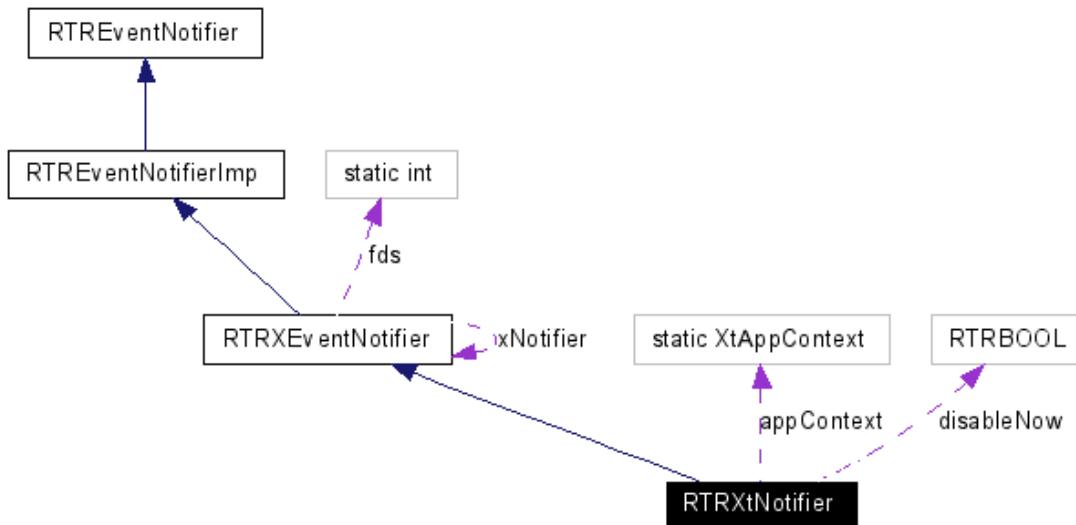
```
#include <xtenimp.h>
```

Inherits [RTRXEventNotifier](#).

Inheritance diagram for RTRXtNotifier:



Collaboration diagram for `RTRXtNotifier`:



Public Member Functions

- [RTRXtNotifier](#) ()
- virtual [~RTRXtNotifier](#) ()
- void [enable](#) ()
- void [disable](#) ()

Public Attributes

- RTRBOOL [disableNow](#)

Static Public Attributes

- static XtAppContext **appContext**

Detailed Description

An implementation of an [RTREventNotifierImp](#) ([RTREventNotifier](#)) based on the Xt library. The application must initialize the static class member appContext. It is of type XtAppContext. The initialization must occur before any methods of the notifier are invoked by any part of the system.

I/O (read, write, exception) events are implemented in a straight forward manner in which this class is just translating from the abstract calls (inherited from [RTREventNotifierImp](#)) to the equivalent underlying Xt calls.

Because Xt uses a signal to implement timer events, this class needs to "synchronize" timer events with I/O events (we don't want to be interrupting ongoing processing of I/O events). The mechanism used to synchronize timer events involves the use of a pipe. The pipe is created by this class (actually by [RTRXEventNotifier](#), an ancestor) and is both written into and read from by this class.

[RTREventNotifierImp](#) maintains a record of all requested timers, only one of which is outstanding at a given time. When a timer event is requested, [RTREventNotifierImp](#) will eventually decide what the next timer interval is and invoke enableTimer(). This method is implemented here to register the method _timer_to_io() with Xt as a timer handling function. When the timer expires Xt will invoke _timer_to_io() which in turn writes a byte into the pipe. This class has registered the method _io_to_events() for read notification on the "other end" of the pipe. Once any current I/O processing is completed, _io_to_events() will be invoked (by X) and will trigger processing of timer events by invoking RTREventNotifier::expireEvents()

```
XtAppContext RTRXtNotifier::appContext = 0;

main()
{
    XtAppContext app_context = ...
    RTRXtNotifier::appContext = app_context;
    XtAppMainLoop(app_context);
}
```

See Also:

[RTRXViewNotifier](#), [RTREventNotifierImp](#), [RTRSelectNotifier](#), [RTRWindowsNotifier](#)

Constructor & Destructor Documentation

RTRXtNotifier::RTRXtNotifier ()

Constructor

virtual RTRXtNotifier::~~RTRXtNotifier () [inline, virtual]

Destructor

Member Function Documentation

void RTRXtNotifier::enable () [inline, virtual]

From [RTREventNotifier](#)

Implements [RTREventNotifier](#). **void RTRXtNotifier::disable () [inline, virtual]**

From [RTREventNotifier](#)

Implements [RTREventNotifier](#).

Member Data Documentation

RTRBOOL [RTRXtNotifier::disableNow](#)

From [RTREventNotifier](#)

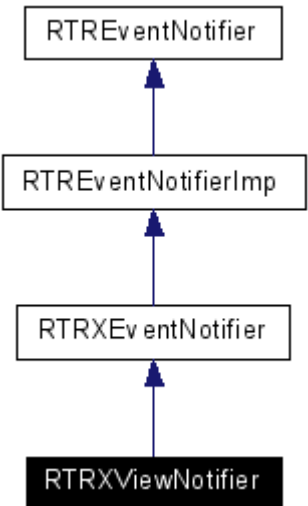
RTRXViewNotifier Class Reference

An implementation of an [RTREventNotifierImp](#) ([RTREventNotifier](#)) based on the XView library. The application must initialize the static class member appContext. It is of type Frame. The initialization must occur before any methods of the notifier are invoked by any part of the system.

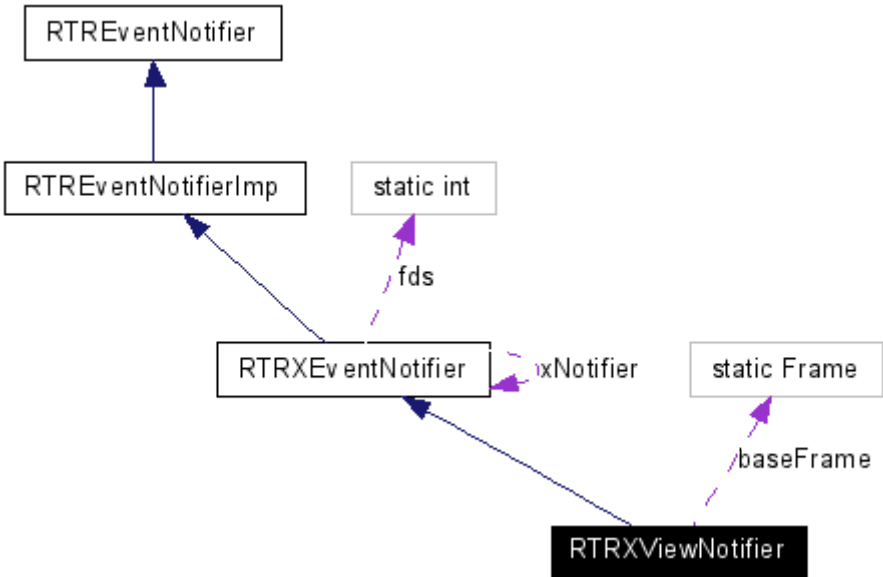
```
#include <xvenimp.h>
```

Inherits [RTRXEventNotifier](#).

Inheritance diagram for RTRXViewNotifier:



Collaboration diagram for RTRXViewNotifier:



Public Member Functions

- [RTRXViewNotifier](#) ()
- virtual [~RTRXViewNotifier](#) ()
- void [enable](#) ()
- void [disable](#) ()

Static Public Attributes

- static Frame **baseFrame**

Detailed Description

An implementation of an [RTREventNotifierImp](#) ([RTREventNotifier](#)) based on the XView library. The application must initialize the static class member appContext. It is of type Frame. The initialization must occur before any methods of the notifier are invoked by any part of the system.

Because XView uses a signal to implement timer events, this classes needs to "synchronize" timer events with I/O events (we don't want to be interrupting ongoing processing of I/O events). The mechanism used to synchronize timer events involves the use of a pipe. The pipe is created by this class (actually by [RTRXEventNotifier](#), an ancestor) and is both written into and read from by this class.

[RTREventNotifierImp](#) maintains a record of all requested timers, only one of which is outstanding at a given time. When a timer event is requested, [RTREventNotifierImp](#) will eventually decide what the next timer interval is and invoke enableTimer(). This method is implemented here to register the method _timer_to_io() with X as a timer handling function. When the timer expires X will invoke _timer_to_io() which in turn writes a byte into the pipe. This class has registered the method _io_to_events() for read notification on the "other end" of the pipe. Once any current I/O processing is completed, _io_to_events() will be invoked (by X) and will trigger processing of timer events by invoking RTREventNotifier::expireEvents()

```
Frame RTRXViewNotifier::baseFrame = 0;

main()
{
    Frame base_frame = ...
    RTRXViewNotifier::baseFrame = base_frame;
    xv_main_loop(base_frame);
}
```

See Also:

[RTRXtNotifier](#), [RTREventNotifierImp](#), [RTRSelectNotifier](#), [RTRWindowsNotifier](#)

Constructor & Destructor Documentation

RTRXViewNotifier::RTRXViewNotifier ()

Constructor

virtual RTRXViewNotifier::~~RTRXViewNotifier () [inline, virtual]

Destructor

Member Function Documentation

void RTRXViewNotifier::enable () [inline, virtual]

From [RTREventNotifier](#)

Implements [RTREventNotifier](#). **void RTRXViewNotifier::disable ()** [inline, virtual]

From [RTREventNotifier](#)

Implements [RTREventNotifier](#).

© 2008, 2012, 2020 Refinitiv. All rights reserved.

Republication or redistribution of Refinitiv content, including by framing or similar means, is prohibited without the prior written consent of Refinitiv. 'Refinitiv' and the Refinitiv logo are registered trademarks and trademarks of Refinitiv.

Any third-party names or marks are the trademarks or registered trademarks of the relevant third party.

Document ID: RMC220RE.200

Date of issue: December 2020

