

MM

Generated by Doxygen 1.8.6

Wed Apr 16 2014 13:09:49

Contents

1 Namespace Index	1
1.1 Namespace List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	7
3.1 Class List	7
4 File Index	13
4.1 File List	13
5 Namespace Documentation	17
5.1 Package LibMM	17
5.1.1 Enumeration Type Documentation	17
5.1.1.1 MESSAGE	17
5.2 MM Namespace Reference	19
5.2.1 Detailed Description	20
5.2.2 Typedef Documentation	21
5.2.2.1 BOOLEAN	21
5.2.2.2 CALLBACK	21
5.2.2.3 CHAR	21
5.2.2.4 INT32	22
5.2.2.5 TID	22
5.2.2.6 UINT32	22
5.2.2.7 UINT8	22
5.2.2.8 VOID	22
5.2.3 Enumeration Type Documentation	22
5.2.3.1 TID	22
6 Class Documentation	25
6.1 MM::Activation Class Reference	25
6.1.1 Constructor & Destructor Documentation	26

6.1.1.1	Activation	26
6.1.1.2	Activation	27
6.1.1.3	Activation	27
6.1.1.4	~Activation	27
6.1.2	Member Function Documentation	27
6.1.2.1	getLocation	27
6.1.2.2	getMessage	27
6.1.2.3	getTypeld	27
6.1.2.4	instanceof	27
6.1.2.5	recycle	27
6.1.2.6	toString	27
6.1.2.7	toString	27
6.2	Activation Class Reference	28
6.2.1	Detailed Description	28
6.3	MM::ActiveExp Class Reference	28
6.3.1	Constructor & Destructor Documentation	30
6.3.1.1	ActiveExp	30
6.3.1.2	ActiveExp	30
6.3.1.3	~ActiveExp	30
6.3.2	Member Function Documentation	30
6.3.2.1	getName	30
6.3.2.2	getTypeld	30
6.3.2.3	instanceof	31
6.3.2.4	recycle	31
6.3.2.5	toString	31
6.4	ActiveExp Class Reference	32
6.4.1	Detailed Description	32
6.5	AliasExp Class Reference	32
6.5.1	Detailed Description	32
6.6	MM::AliasExp Class Reference	32
6.6.1	Constructor & Destructor Documentation	34
6.6.1.1	AliasExp	34
6.6.1.2	AliasExp	34
6.6.1.3	~AliasExp	34
6.6.2	Member Function Documentation	34
6.6.2.1	getTypeld	34
6.6.2.2	instanceof	34
6.6.2.3	recycle	35
6.6.2.4	toString	35
6.7	MM::AllExp Class Reference	36

6.7.1	Constructor & Destructor Documentation	37
6.7.1.1	AllExp	37
6.7.1.2	AllExp	37
6.7.1.3	~AllExp	37
6.7.2	Member Function Documentation	37
6.7.2.1	getTypeld	37
6.7.2.2	instanceof	37
6.7.2.3	recycle	38
6.7.2.4	toString	38
6.8	AllExp Class Reference	39
6.8.1	Detailed Description	39
6.9	Assertion Class Reference	39
6.9.1	Detailed Description	39
6.10	MM::Assertion Class Reference	39
6.10.1	Constructor & Destructor Documentation	41
6.10.1.1	Assertion	41
6.10.1.2	Assertion	41
6.10.1.3	~Assertion	41
6.10.2	Member Function Documentation	41
6.10.2.1	getExp	41
6.10.2.2	getLocation	41
6.10.2.3	getMessage	41
6.10.2.4	getMessageLength	41
6.10.2.5	getTypeld	41
6.10.2.6	instanceof	41
6.10.2.7	recycle	41
6.10.2.8	toString	42
6.10.2.9	toString	42
6.11	BinExp Class Reference	42
6.11.1	Detailed Description	42
6.12	MM::BinExp Class Reference	42
6.12.1	Constructor & Destructor Documentation	44
6.12.1.1	BinExp	44
6.12.1.2	BinExp	44
6.12.1.3	~BinExp	44
6.12.2	Member Function Documentation	44
6.12.2.1	getLhsExp	44
6.12.2.2	getOperator	44
6.12.2.3	getRhsExp	45
6.12.2.4	getTypeld	45

6.12.2.5 <code>instanceof</code>	45
6.12.2.6 <code>isAlaiasExp</code>	45
6.12.2.7 <code>isTriggerExp</code>	46
6.12.2.8 <code>recycle</code>	46
6.12.2.9 <code>toString</code>	46
6.13 BooleanValExp Class Reference	47
6.13.1 Detailed Description	47
6.14 MM::BooleanValExp Class Reference	47
6.14.1 Constructor & Destructor Documentation	48
6.14.1.1 <code>BooleanValExp</code>	48
6.14.1.2 <code>BooleanValExp</code>	48
6.14.1.3 <code>~BooleanValExp</code>	48
6.14.2 Member Function Documentation	49
6.14.2.1 <code>getLocation</code>	49
6.14.2.2 <code>getTypeld</code>	49
6.14.2.3 <code>getValue</code>	49
6.14.2.4 <code>greaterEquals</code>	49
6.14.2.5 <code>instanceof</code>	49
6.14.2.6 <code>recycle</code>	50
6.14.2.7 <code>toString</code>	50
6.15 MM::Name::Compare Class Reference	50
6.15.1 Detailed Description	50
6.15.2 Member Function Documentation	51
6.15.2.1 <code>operator()</code>	51
6.16 MM::Node::Compare Class Reference	51
6.16.1 Member Function Documentation	51
6.16.1.1 <code>operator()</code>	51
6.17 MM::ConverterNodeBehavior Class Reference	51
6.17.1 Constructor & Destructor Documentation	53
6.17.1.1 <code>ConverterNodeBehavior</code>	53
6.17.1.2 <code>~ConverterNodeBehavior</code>	53
6.17.2 Member Function Documentation	53
6.17.2.1 <code>activateTriggerTargets</code>	53
6.17.2.2 <code>add</code>	53
6.17.2.3 <code>begin</code>	53
6.17.2.4 <code>change</code>	53
6.17.2.5 <code>end</code>	53
6.17.2.6 <code>getCapacity</code>	53
6.17.2.7 <code>getCreateMessage</code>	53
6.17.2.8 <code>getDeleteMessage</code>	54

6.17.2.9	getDrainNode	54
6.17.2.10	getFrom	54
6.17.2.11	getResources	54
6.17.2.12	getSourceNode	54
6.17.2.13	getTo	54
6.17.2.14	getTriggerEdge	54
6.17.2.15	getTypeld	54
6.17.2.16	getUpdateMessage	54
6.17.2.17	hasCapacity	54
6.17.2.18	hasResources	54
6.17.2.19	instanceof	54
6.17.2.20	recycle	55
6.17.2.21	setDrainNode	55
6.17.2.22	setFrom	55
6.17.2.23	setSourceNode	55
6.17.2.24	setTo	55
6.17.2.25	setTriggerEdge	55
6.17.2.26	step	55
6.17.2.27	stepPullAll	55
6.17.2.28	stepPullAny	55
6.17.2.29	stepPushAll	55
6.17.2.30	stepPushAny	55
6.17.2.31	sub	55
6.17.2.32	toString	56
6.17.2.33	toString	56
6.18	ConverterNodeBehavior Class Reference	56
6.18.1	Detailed Description	56
6.19	MM::Declaration Class Reference	57
6.19.1	Constructor & Destructor Documentation	58
6.19.1.1	Declaration	58
6.19.1.2	~Declaration	58
6.19.2	Member Function Documentation	58
6.19.2.1	addInterface	59
6.19.2.2	begin	60
6.19.2.3	change	60
6.19.2.4	end	60
6.19.2.5	getDefinition	61
6.19.2.6	getInterface	61
6.19.2.7	getTypeld	61
6.19.2.8	getTypeName	61

6.19.2.9 <code>instanceof</code>	61
6.19.2.10 <code>recycle</code>	61
6.19.2.11 <code>removeInterface</code>	62
6.19.2.12 <code>setDefinition</code>	62
6.19.2.13 <code>toString</code>	62
6.19.2.14 <code>toString</code>	62
6.19.2.15 <code>update</code>	62
6.20 Declaration Class Reference	63
6.20.1 Detailed Description	63
6.21 MM::Definition Class Reference	63
6.21.1 Constructor & Destructor Documentation	65
6.21.1.1 <code>Definition</code>	65
6.21.1.2 <code>Definition</code>	65
6.21.1.3 <code>~Definition</code>	65
6.21.2 Member Function Documentation	65
6.21.2.1 <code>addElement</code>	65
6.21.2.2 <code>addPullAllNode</code>	66
6.21.2.3 <code>addPullAnyNode</code>	66
6.21.2.4 <code>addPushAllNode</code>	66
6.21.2.5 <code>addPushAnyNode</code>	66
6.21.2.6 <code>clearElements</code>	66
6.21.2.7 <code>containsElement</code>	66
6.21.2.8 <code>deprioritize</code>	66
6.21.2.9 <code>findDeclaredDefinition</code>	67
6.21.2.10 <code>findNode</code>	67
6.21.2.11 <code>findQueriedDefinition</code>	67
6.21.2.12 <code>getElement</code>	68
6.21.2.13 <code>getElements</code>	68
6.21.2.14 <code>getPullAllNodes</code>	68
6.21.2.15 <code>getPullAnyNodes</code>	68
6.21.2.16 <code>getPushAllNodes</code>	68
6.21.2.17 <code>getPushAnyNodes</code>	68
6.21.2.18 <code>getTypeld</code>	68
6.21.2.19 <code>instanceof</code>	69
6.21.2.20 <code>prioritize</code>	69
6.21.2.21 <code>putElement</code>	69
6.21.2.22 <code>recycle</code>	69
6.21.2.23 <code>removeElement</code>	70
6.21.2.24 <code>removeElement</code>	70
6.21.2.25 <code>removePullAllNode</code>	70

6.21.2.26 removePullAnyNode	70
6.21.2.27 removePushAllNode	70
6.21.2.28 removePushAnyNode	70
6.21.2.29 setNameToElementMap	70
6.21.2.30 setParent	70
6.21.2.31 setPullAllNodes	71
6.21.2.32 setPullAnyNodes	71
6.21.2.33 setPushAllNodes	71
6.21.2.34 setPushAnyNodes	72
6.21.2.35 toString	72
6.21.2.36 toString	72
6.21.3 Member Data Documentation	73
6.21.3.1 COMMA_CHAR	73
6.21.3.2 LBRACE_CHAR	73
6.21.3.3 RBRACE_CHAR	73
6.22 Definition Class Reference	73
6.22.1 Detailed Description	73
6.23 MM::Machine::Delegate Class Reference	74
6.23.1 Detailed Description	75
6.23.2 Constructor & Destructor Documentation	75
6.23.2.1 Delegate	75
6.23.2.2 ~Delegate	75
6.23.3 Member Function Documentation	75
6.23.3.1 getTypeld	75
6.23.3.2 instanceof	75
6.23.3.3 update	76
6.24 Deletion Class Reference	76
6.24.1 Detailed Description	76
6.25 MM::Deletion Class Reference	76
6.25.1 Constructor & Destructor Documentation	78
6.25.1.1 Deletion	78
6.25.1.2 Deletion	78
6.25.1.3 ~Deletion	78
6.25.2 Member Function Documentation	78
6.25.2.1 getLocation	78
6.25.2.2 getTypeld	78
6.25.2.3 instanceof	78
6.25.2.4 recycle	78
6.25.2.5 toString	78
6.25.2.6 toString	79

6.26 DieExp Class Reference	79
6.26.1 Detailed Description	79
6.27 MM::DieExp Class Reference	79
6.27.1 Detailed Description	81
6.27.2 Constructor & Destructor Documentation	81
6.27.2.1 DieExp	81
6.27.2.2 DieExp	81
6.27.2.3 ~DieExp	81
6.27.3 Member Function Documentation	81
6.27.3.1 getMax	81
6.27.3.2 getTypeid	82
6.27.3.3 instanceof	82
6.27.3.4 recycle	82
6.27.3.5 toString	83
6.28 Disablement Class Reference	83
6.28.1 Detailed Description	83
6.29 MM::Disablement Class Reference	84
6.29.1 Constructor & Destructor Documentation	85
6.29.1.1 Disablement	85
6.29.1.2 Disablement	86
6.29.1.3 Disablement	86
6.29.1.4 ~Disablement	86
6.29.2 Member Function Documentation	86
6.29.2.1 getLocation	86
6.29.2.2 getMessage	86
6.29.2.3 getName	86
6.29.2.4 getTypeid	86
6.29.2.5 instanceof	86
6.29.2.6 recycle	86
6.29.2.7 toString	86
6.29.2.8 toString	86
6.30 DrainNodeBehavior Class Reference	87
6.30.1 Detailed Description	87
6.31 MM::DrainNodeBehavior Class Reference	87
6.31.1 Constructor & Destructor Documentation	89
6.31.1.1 DrainNodeBehavior	89
6.31.1.2 ~DrainNodeBehavior	89
6.31.2 Member Function Documentation	89
6.31.2.1 add	89
6.31.2.2 begin	89

6.31.2.3	change	89
6.31.2.4	end	89
6.31.2.5	getCapacity	89
6.31.2.6	getCreateMessage	90
6.31.2.7	getDeleteMessage	90
6.31.2.8	getResources	90
6.31.2.9	getTypeld	90
6.31.2.10	getUpdateMessage	90
6.31.2.11	hasCapacity	90
6.31.2.12	hasResources	90
6.31.2.13	instanceof	90
6.31.2.14	recycle	90
6.31.2.15	stepPullAll	91
6.31.2.16	stepPushAll	92
6.31.2.17	stepPushAny	92
6.31.2.18	sub	92
6.31.2.19	toString	92
6.31.2.20	toString	92
6.32	Edge Class Reference	93
6.32.1	Detailed Description	93
6.33	MM::Edge Class Reference	93
6.33.1	Constructor & Destructor Documentation	94
6.33.1.1	Edge	94
6.33.1.2	~Edge	95
6.33.2	Member Function Documentation	95
6.33.2.1	getExp	95
6.33.2.2	getSource	95
6.33.2.3	getSourceName	95
6.33.2.4	getTarget	96
6.33.2.5	getTargetName	96
6.33.2.6	getTypeld	96
6.33.2.7	instanceof	96
6.33.2.8	recycle	97
6.33.2.9	setExp	97
6.33.2.10	setSource	98
6.33.2.11	setTarget	98
6.33.2.12	toString	98
6.33.2.13	toString	98
6.34	MM::Element Class Reference	99
6.34.1	Constructor & Destructor Documentation	100

6.34.1.1	Element	100
6.34.1.2	~Element	100
6.34.2	Member Function Documentation	101
6.34.2.1	begin	101
6.34.2.2	change	101
6.34.2.3	end	101
6.34.2.4	getName	102
6.34.2.5	getTypeld	102
6.34.2.6	instanceof	103
6.34.2.7	isVisible	105
6.34.2.8	recycle	105
6.34.2.9	setName	106
6.34.2.10	setVisible	106
6.34.2.11	toString	106
6.34.2.12	toString	107
6.34.3	Member Data Documentation	107
6.34.3.1	name	107
6.34.3.2	visible	107
6.35	Element Class Reference	107
6.35.1	Detailed Description	107
6.36	MM::Enablement Class Reference	107
6.36.1	Constructor & Destructor Documentation	109
6.36.1.1	Enablement	109
6.36.1.2	Enablement	109
6.36.1.3	Enablement	109
6.36.1.4	~Enablement	109
6.36.2	Member Function Documentation	109
6.36.2.1	getLocation	109
6.36.2.2	getMessage	109
6.36.2.3	getName	109
6.36.2.4	getTypeld	109
6.36.2.5	instanceof	109
6.36.2.6	recycle	110
6.36.2.7	toString	110
6.36.2.8	toString	110
6.37	Enablement Class Reference	110
6.37.1	Detailed Description	110
6.38	MM::Evaluator Class Reference	111
6.38.1	Constructor & Destructor Documentation	112
6.38.1.1	Evaluator	112

6.38.1.2 ~Evaluator	112
6.38.2 Member Function Documentation	112
6.38.2.1 eval	112
6.38.2.2 eval	113
6.38.2.3 eval	113
6.38.2.4 eval	113
6.38.2.5 eval	114
6.38.2.6 eval	114
6.38.2.7 eval	115
6.38.2.8 eval	115
6.38.2.9 eval	115
6.38.2.10 eval	116
6.38.2.11 eval	116
6.38.2.12 eval	116
6.38.2.13 eval	117
6.38.2.14 eval	117
6.38.2.15 eval	117
6.38.2.16 eval	117
6.38.2.17 eval	118
6.38.2.18 eval	118
6.38.2.19 eval	118
6.38.2.20 eval	118
6.38.2.21 eval	118
6.38.2.22 eval	119
6.38.2.23 eval	119
6.38.2.24 eval	119
6.38.2.25 eval	119
6.38.2.26 eval	120
6.38.2.27 eval	120
6.38.2.28 getTypeld	120
6.38.2.29 instanceof	120
6.38.2.30 recycle	120
6.38.2.31 step	121
6.38.2.32 toString	121
6.39 Evaluator Class Reference	121
6.39.1 Detailed Description	121
6.40 MM::Event Class Reference	122
6.40.1 Constructor & Destructor Documentation	123
6.40.1.1 Event	123
6.40.1.2 ~Event	124

6.40.2 Member Function Documentation	124
6.40.2.1 getElement	124
6.40.2.2 getInstance	124
6.40.2.3 getMessage	124
6.40.2.4 getTypeld	124
6.40.2.5 instanceof	124
6.40.2.6 recycle	125
6.40.2.7 setElement	126
6.40.2.8 setInstance	126
6.40.2.9 toString	126
6.40.2.10 toString	126
6.40.3 Member Data Documentation	126
6.40.3.1 element	126
6.40.3.2 instance	127
6.41 Event Class Reference	127
6.41.1 Detailed Description	127
6.42 MM::Exp Class Reference	127
6.42.1 Constructor & Destructor Documentation	128
6.42.1.1 Exp	128
6.42.1.2 ~Exp	128
6.42.2 Member Function Documentation	129
6.42.2.1 getTypeld	129
6.42.2.2 instanceof	129
6.42.2.3 recycle	130
6.42.2.4 toString	131
6.43 Exp Class Reference	132
6.43.1 Detailed Description	132
6.44 FailEvent Class Reference	132
6.44.1 Detailed Description	132
6.45 MM::Failure Class Reference	132
6.45.1 Constructor & Destructor Documentation	134
6.45.1.1 Failure	134
6.45.1.2 Failure	134
6.45.1.3 ~Failure	134
6.45.2 Member Function Documentation	134
6.45.2.1 getLocation	134
6.45.2.2 getMessage	134
6.45.2.3 getTypeld	134
6.45.2.4 instanceof	134
6.45.2.5 recycle	135

6.45.2.6	toString	135
6.45.2.7	toString	135
6.46	MM::FlowEdge Class Reference	135
6.46.1	Constructor & Destructor Documentation	137
6.46.1.1	~FlowEdge	137
6.46.1.2	FlowEdge	137
6.46.2	Member Function Documentation	137
6.46.2.1	getTypeld	137
6.46.2.2	instanceof	137
6.46.2.3	recycle	137
6.46.2.4	toString	138
6.46.2.5	toString	138
6.47	FlowEdge Class Reference	138
6.47.1	Detailed Description	138
6.48	FlowEvent Class Reference	138
6.48.1	Detailed Description	139
6.49	MM::FlowEvent Class Reference	139
6.49.1	Constructor & Destructor Documentation	141
6.49.1.1	FlowEvent	141
6.49.1.2	~FlowEvent	141
6.49.2	Member Function Documentation	141
6.49.2.1	getActEdge	141
6.49.2.2	getActInstance	141
6.49.2.3	getActNode	141
6.49.2.4	getAmount	141
6.49.2.5	getMessage	141
6.49.2.6	getSourceInstance	141
6.49.2.7	getSourceNode	142
6.49.2.8	getTargetInstance	142
6.49.2.9	getTargetNode	142
6.49.2.10	getTypeld	142
6.49.2.11	instanceof	142
6.49.2.12	recycle	142
6.49.2.13	setSourceInstance	142
6.49.2.14	setTargetInstance	142
6.49.2.15	toString	142
6.49.2.16	toString	142
6.50	GateNodeBehavior Class Reference	143
6.50.1	Detailed Description	143
6.51	MM::GateNodeBehavior Class Reference	143

6.51.1	Constructor & Destructor Documentation	145
6.51.1.1	GateNodeBehavior	145
6.51.1.2	~GateNodeBehavior	145
6.51.2	Member Function Documentation	145
6.51.2.1	add	145
6.51.2.2	begin	145
6.51.2.3	change	146
6.51.2.4	end	146
6.51.2.5	getCapacity	146
6.51.2.6	getCreateMessage	146
6.51.2.7	getDeleteMessage	146
6.51.2.8	getResources	146
6.51.2.9	getTypeld	146
6.51.2.10	getUpdateMessage	146
6.51.2.11	hasCapacity	146
6.51.2.12	hasResources	146
6.51.2.13	instanceof	146
6.51.2.14	recycle	147
6.51.2.15	stepPullAll	147
6.51.2.16	stepPushAll	147
6.51.2.17	sub	147
6.51.2.18	toString	148
6.51.2.19	toString	148
6.52	Instance Class Reference	148
6.52.1	Detailed Description	148
6.53	MM::Instance Class Reference	149
6.53.1	Constructor & Destructor Documentation	150
6.53.1.1	Instance	150
6.53.1.2	~Instance	151
6.53.2	Member Function Documentation	151
6.53.2.1	add	151
6.53.2.2	begin	152
6.53.2.3	clearActive	152
6.53.2.4	clearDisabled	152
6.53.2.5	createInstances	152
6.53.2.6	deleteValue	153
6.53.2.7	destroyAllInstances	153
6.53.2.8	destroyInstance	154
6.53.2.9	destroyInstances	154
6.53.2.10	finalize	155

6.53.2.11 getCapacity	155
6.53.2.12 getDeclaration	156
6.53.2.13 getDefinition	156
6.53.2.14 getEvaluatedExp	157
6.53.2.15 getGateValue	157
6.53.2.16 getIndex	158
6.53.2.17 getInstance	158
6.53.2.18 getInstances	159
6.53.2.19 getInstances	159
6.53.2.20 getNewValue	159
6.53.2.21 getOldValue	160
6.53.2.22 getParent	160
6.53.2.23 getResources	160
6.53.2.24 getTypeld	161
6.53.2.25 getValue	161
6.53.2.26 hasCapacity	161
6.53.2.27 getResources	162
6.53.2.28 instanceof	162
6.53.2.29 isActive	163
6.53.2.30 isDisabled	163
6.53.2.31 isEvaluatedExp	164
6.53.2.32 isMarked	164
6.53.2.33 mark	164
6.53.2.34 nameToString	165
6.53.2.35 nameToString	165
6.53.2.36 recycle	165
6.53.2.37 setActive	166
6.53.2.38 setDisabled	166
6.53.2.39 setEvaluatedExp	167
6.53.2.40 setGateValue	167
6.53.2.41 setNewValue	168
6.53.2.42 setNextActive	168
6.53.2.43 setOldValue	168
6.53.2.44 setValue	169
6.53.2.45 sub	169
6.53.2.46 sweep	170
6.53.2.47 toString	170
6.53.2.48 toString	171
6.53.2.49 update	171
6.54 MM::Machine::InstanceObserver Class Reference	172

6.54.1 Detailed Description	173
6.54.2 Constructor & Destructor Documentation	173
6.54.2.1 InstanceObserver	173
6.54.2.2 ~InstanceObserver	174
6.54.3 Member Function Documentation	174
6.54.3.1 getTypeld	174
6.54.3.2 instanceof	174
6.54.3.3 update	174
6.55 InterfaceNode Class Reference	174
6.55.1 Detailed Description	175
6.56 MM::InterfaceNode Class Reference	175
6.56.1 Constructor & Destructor Documentation	176
6.56.1.1 InterfaceNode	177
6.56.1.2 ~InterfaceNode	177
6.56.2 Member Function Documentation	177
6.56.2.1 activateTriggerTargets	177
6.56.2.2 add	177
6.56.2.3 getBehavior	177
6.56.2.4 getCapacity	177
6.56.2.5 getDeclaration	177
6.56.2.6 getNode	177
6.56.2.7 getResources	177
6.56.2.8 getTypeld	177
6.56.2.9 hasCapacity	177
6.56.2.10 hasResources	177
6.56.2.11 instanceof	178
6.56.2.12 recycle	178
6.56.2.13 sub	178
6.56.2.14 toString	178
6.57 MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::Iterator Class Reference	179
6.57.1 Constructor & Destructor Documentation	179
6.57.1.1 Iterator	179
6.57.1.2 ~Iterator	179
6.57.2 Member Function Documentation	179
6.57.2.1 getNext	180
6.57.2.2 getPrev	180
6.57.2.3 hasNext	180
6.57.2.4 reset	181
6.58 MM::Vector< T >::Iterator Class Reference	181
6.58.1 Constructor & Destructor Documentation	181

6.58.1.1	Iterator	181
6.58.1.2	~Iterator	181
6.58.2	Member Function Documentation	181
6.58.2.1	getNext	182
6.58.2.2	hasNext	183
6.58.2.3	reset	184
6.59	Location Class Reference	184
6.59.1	Detailed Description	184
6.60	MM::Location Class Reference	184
6.60.1	Constructor & Destructor Documentation	185
6.60.1.1	Location	185
6.60.1.2	~Location	186
6.60.2	Member Function Documentation	186
6.60.2.1	getColumn	186
6.60.2.2	getLength	186
6.60.2.3	getLine	186
6.60.2.4	getTypeld	186
6.60.2.5	instanceof	186
6.60.2.6	recycle	187
6.60.2.7	toString	187
6.61	Machine Class Reference	188
6.61.1	Detailed Description	188
6.62	MM::Machine Class Reference	188
6.62.1	Constructor & Destructor Documentation	192
6.62.1.1	Machine	192
6.62.1.2	~Machine	192
6.62.2	Member Function Documentation	192
6.62.2.1	activate	192
6.62.2.2	addDefinitionObserver	193
6.62.2.3	addInstanceObserver	193
6.62.2.4	createActivation	193
6.62.2.5	createActivation	193
6.62.2.6	createActiveExp	193
6.62.2.7	createActiveExp	194
6.62.2.8	createAliasExp	194
6.62.2.9	createAliasExp	194
6.62.2.10	createAllExp	194
6.62.2.11	createAnonymousTriggerEdge	195
6.62.2.12	createAssertion	195
6.62.2.13	createAssertion	196

6.62.2.14 createBinExp	196
6.62.2.15 createBinExp	196
6.62.2.16 createBooleanValExp	197
6.62.2.17 createBooleanValExp	197
6.62.2.18 createConverterNode	197
6.62.2.19 createDeclaration	198
6.62.2.20 createDefinition	198
6.62.2.21 createDefinition	199
6.62.2.22 createDeletion	199
6.62.2.23 createDeletion	200
6.62.2.24 createDisablement	200
6.62.2.25 createDisablement	200
6.62.2.26 createDrainNode	201
6.62.2.27 createEdgeVector	201
6.62.2.28 createElementVector	201
6.62.2.29 createEnablement	201
6.62.2.30 createEnablement	202
6.62.2.31 createFailure	202
6.62.2.32 createFailure	203
6.62.2.33 createFlowEdge	203
6.62.2.34 createFlowEvent	203
6.62.2.35 createGateNode	204
6.62.2.36 createInstance	205
6.62.2.37 createInterfaceNode	205
6.62.2.38 createLocation	206
6.62.2.39 createModification	206
6.62.2.40 createModification	207
6.62.2.41 createModification	207
6.62.2.42 createName	207
6.62.2.43 createName	208
6.62.2.44 createName	208
6.62.2.45 createName	208
6.62.2.46 createName	209
6.62.2.47 createName2ElementMap	209
6.62.2.48 createName2NodeMap	209
6.62.2.49 createNodeVector	209
6.62.2.50 createNumberValExp	209
6.62.2.51 createNumberValExp	210
6.62.2.52 createOneExp	210
6.62.2.53 createOneExp	210

6.62.2.54 createOverrideExp	211
6.62.2.55 createOverrideExp	211
6.62.2.56 createPoolNode	211
6.62.2.57 createPrevention	212
6.62.2.58 createPrevention	212
6.62.2.59 createProgram	213
6.62.2.60 createProgram	213
6.62.2.61 createRangeValExp	213
6.62.2.62 createRangeValExp	214
6.62.2.63 createRefNode	214
6.62.2.64 createSourceNode	215
6.62.2.65 createStateEdge	215
6.62.2.66 createString	216
6.62.2.67 createTransformationVector	216
6.62.2.68 createTransition	216
6.62.2.69 createTransition	217
6.62.2.70 createTransition	217
6.62.2.71 createTriggerEvent	217
6.62.2.72 createTriggerEvent	218
6.62.2.73 createUnExp	218
6.62.2.74 createUnExp	219
6.62.2.75 createVarExp	219
6.62.2.76 createViolation	219
6.62.2.77 createViolation	220
6.62.2.78 eatWhiteSpace	220
6.62.2.79 eval	221
6.62.2.80 evalFile	222
6.62.2.81 getDefinition	223
6.62.2.82 getEvaluator	223
6.62.2.83 getInstance	223
6.62.2.84 getInstance	224
6.62.2.85 getInstanceName	224
6.62.2.86 getLog	224
6.62.2.87 getName	225
6.62.2.88 getReflector	225
6.62.2.89 getTypeld	225
6.62.2.90 instanceof	225
6.62.2.91 reflect	226
6.62.2.92 removeObserver	226
6.62.2.93 reset	226

6.62.2.94 reset	226
6.62.2.95 setDefinition	226
6.62.2.96 setInstance	226
6.62.2.97 step	227
6.62.2.98 step	227
6.62.2.99 step	227
6.62.3 Member Data Documentation	227
6.62.3.1 delegates	227
6.62.3.2 LOG_SIZE	227
6.63 MM::Map< MAP_KEY, MAP_VALUE, COMPARE > Class Template Reference	227
6.63.1 Constructor & Destructor Documentation	228
6.63.1.1 Map	228
6.63.1.2 Map	228
6.63.1.3 ~Map	228
6.63.2 Member Function Documentation	228
6.63.2.1 clear	228
6.63.2.2 contains	228
6.63.2.3 get	228
6.63.2.4 getIterator	229
6.63.2.5 getTypeld	229
6.63.2.6 instanceof	229
6.63.2.7 isEmpty	229
6.63.2.8 put	229
6.63.2.9 putAll	229
6.63.2.10 remove	229
6.64 Map Class Reference	230
6.64.1 Detailed Description	230
6.65 MM::Modification Class Reference	230
6.65.1 Constructor & Destructor Documentation	231
6.65.1.1 Modification	231
6.65.1.2 Modification	231
6.65.1.3 ~Modification	231
6.65.2 Member Function Documentation	231
6.65.2.1 getTypeld	231
6.65.2.2 instanceof	231
6.65.2.3 recycle	232
6.65.2.4 toString	232
6.66 Modification Class Reference	233
6.66.1 Detailed Description	233
6.67 MM::Name Class Reference	233

6.67.1	Constructor & Destructor Documentation	234
6.67.1.1	~Name	234
6.67.1.2	Name	234
6.67.2	Member Function Documentation	235
6.67.2.1	append	235
6.67.2.2	equals	235
6.67.2.3	equals	236
6.67.2.4	getBuffer	236
6.67.2.5	getLength	237
6.67.2.6	getLocation	237
6.67.2.7	getName	237
6.67.2.8	getPreName	238
6.67.2.9	getTypeld	238
6.67.2.10	instanceof	238
6.67.2.11	print	238
6.67.2.12	recycle	239
6.67.2.13	setLocation	240
6.67.2.14	setName	240
6.67.2.15	setPreName	241
6.67.2.16	toString	241
6.67.3	Friends And Related Function Documentation	243
6.67.3.1	operator<	243
6.68	Name Class Reference	243
6.68.1	Detailed Description	243
6.69	MM::Node Class Reference	243
6.69.1	Constructor & Destructor Documentation	245
6.69.1.1	Node	245
6.69.1.2	~Node	245
6.69.2	Member Function Documentation	245
6.69.2.1	activateTriggerTargets	245
6.69.2.2	add	246
6.69.2.3	addAlias	246
6.69.2.4	addCondition	246
6.69.2.5	addInput	246
6.69.2.6	addOutput	246
6.69.2.7	addTrigger	246
6.69.2.8	begin	246
6.69.2.9	change	246
6.69.2.10	end	246
6.69.2.11	getAliases	247

6.69.2.12 getAmount	247
6.69.2.13 getBehavior	247
6.69.2.14 getCapacity	248
6.69.2.15 getConditions	249
6.69.2.16 getInput	249
6.69.2.17 getOutput	250
6.69.2.18 getResources	250
6.69.2.19 getTriggers	250
6.69.2.20 getTypeld	251
6.69.2.21 hasCapacity	251
6.69.2.22 hasEdgeOwnership	251
6.69.2.23 hasResources	251
6.69.2.24 instanceof	251
6.69.2.25 isDisabled	252
6.69.2.26 isSatisfied	253
6.69.2.27 recycle	253
6.69.2.28 removeAlias	254
6.69.2.29 removeCondition	254
6.69.2.30 removeInput	254
6.69.2.31 removeOutput	254
6.69.2.32 removeTrigger	254
6.69.2.33 setAliases	254
6.69.2.34 setBehavior	254
6.69.2.35 setConditions	254
6.69.2.36 setEdgeOwnership	254
6.69.2.37 setInput	255
6.69.2.38 setOutput	255
6.69.2.39 setTriggers	255
6.69.2.40 step	255
6.69.2.41 sub	256
6.69.2.42 toString	256
6.69.2.43 toString	256
6.70 Node Class Reference	256
6.70.1 Detailed Description	257
6.71 MM::NodeBehavior Class Reference	257
6.71.1 Member Typedef Documentation	259
6.71.1.1 Act	259
6.71.1.2 How	259
6.71.1.3 IO	259
6.71.1.4 When	259

6.71.2 Member Enumeration Documentation	259
6.71.2.1 __Act	259
6.71.2.2 __How	259
6.71.2.3 __IO	259
6.71.2.4 __When	260
6.71.3 Constructor & Destructor Documentation	260
6.71.3.1 NodeBehavior	260
6.71.3.2 ~NodeBehavior	260
6.71.4 Member Function Documentation	260
6.71.4.1 activateTriggerTargets	260
6.71.4.2 add	260
6.71.4.3 begin	260
6.71.4.4 change	261
6.71.4.5 conformsTo	261
6.71.4.6 end	261
6.71.4.7 getAct	261
6.71.4.8 getAmount	261
6.71.4.9 getCapacity	261
6.71.4.10 getCreateMessage	261
6.71.4.11 getDeleteMessage	262
6.71.4.12 getHow	262
6.71.4.13 getIO	262
6.71.4.14 getResources	263
6.71.4.15 getTypeld	263
6.71.4.16 getUpdateMessage	263
6.71.4.17 getWhen	263
6.71.4.18 getWork	264
6.71.4.19 hasCapacity	264
6.71.4.20 hasResources	264
6.71.4.21 instanceof	264
6.71.4.22 recycle	265
6.71.4.23 setAct	266
6.71.4.24 setHow	266
6.71.4.25 setIO	266
6.71.4.26 setWhen	266
6.71.4.27 step	266
6.71.4.28 stepPullAll	267
6.71.4.29 stepPullAny	267
6.71.4.30 stepPushAll	268
6.71.4.31 stepPushAny	268

6.71.4.32 <code>sub</code>	268
6.71.4.33 <code>toString</code>	268
6.71.5 Member Data Documentation	269
6.71.5.1 <code>ACT_LEN</code>	269
6.71.5.2 <code>ACT_STR</code>	269
6.71.5.3 <code>HOW_LEN</code>	269
6.71.5.4 <code>HOW_STR</code>	270
6.71.5.5 <code>IO_LEN</code>	270
6.71.5.6 <code>IO_STR</code>	270
6.71.5.7 <code>WHEN_LEN</code>	270
6.71.5.8 <code>WHEN_STR</code>	270
6.72 NodeBehavior Class Reference	271
6.72.1 Detailed Description	271
6.73 MM::NodeWorkItem Class Reference	271
6.73.1 Constructor & Destructor Documentation	271
6.73.1.1 <code>NodeWorkItem</code>	271
6.73.1.2 <code>~NodeWorkItem</code>	271
6.73.2 Member Function Documentation	271
6.73.2.1 <code>getEdge</code>	272
6.73.2.2 <code>getInstance</code>	272
6.73.2.3 <code>getNode</code>	273
6.74 NodeWorkItem Class Reference	273
6.74.1 Detailed Description	273
6.75 NumberValExp Class Reference	273
6.75.1 Detailed Description	273
6.76 MM::NumberValExp Class Reference	274
6.76.1 Constructor & Destructor Documentation	275
6.76.1.1 <code>NumberValExp</code>	275
6.76.1.2 <code>NumberValExp</code>	276
6.76.1.3 <code>NumberValExp</code>	276
6.76.1.4 <code>NumberValExp</code>	276
6.76.1.5 <code>~NumberValExp</code>	276
6.76.2 Member Function Documentation	276
6.76.2.1 <code>getIntValue</code>	276
6.76.2.2 <code>getLocation</code>	276
6.76.2.3 <code>getTypeld</code>	276
6.76.2.4 <code>getValue</code>	276
6.76.2.5 <code>greaterEquals</code>	277
6.76.2.6 <code>instanceof</code>	277
6.76.2.7 <code>recycle</code>	277

6.76.2.8	toString	277
6.77	MM::Observable Class Reference	278
6.77.1	Constructor & Destructor Documentation	278
6.77.1.1	Observable	279
6.77.1.2	~Observable	279
6.77.2	Member Function Documentation	279
6.77.2.1	addObserver	279
6.77.2.2	getTypeld	279
6.77.2.3	instanceof	279
6.77.2.4	notifyObservers	280
6.77.2.5	recylce	280
6.77.2.6	removeObserver	280
6.78	Observable Class Reference	281
6.78.1	Detailed Description	281
6.79	Observer Class Reference	281
6.79.1	Detailed Description	281
6.80	MM::Observer Class Reference	281
6.80.1	Constructor & Destructor Documentation	282
6.80.1.1	Observer	282
6.80.1.2	~Observer	282
6.80.2	Member Function Documentation	282
6.80.2.1	getTypeld	282
6.80.2.2	instanceof	282
6.80.2.3	update	283
6.81	MM::OneExp Class Reference	283
6.81.1	Constructor & Destructor Documentation	285
6.81.1.1	OneExp	285
6.81.1.2	OneExp	285
6.81.1.3	~OneExp	285
6.81.2	Member Function Documentation	285
6.81.2.1	getTypeld	285
6.81.2.2	greaterEquals	286
6.81.2.3	instanceof	286
6.81.2.4	recycle	286
6.81.2.5	toString	286
6.82	OneExp Class Reference	287
6.82.1	Detailed Description	287
6.83	MM::Operator Class Reference	287
6.83.1	Member Typedef Documentation	287
6.83.1.1	OP	287

6.83.2 Member Enumeration Documentation	287
6.83.2.1 __OP	287
6.83.3 Member Data Documentation	288
6.83.3.1 OP_LEN	288
6.83.3.2 OP_STR	288
6.84 Operator Class Reference	289
6.84.1 Detailed Description	289
6.85 MM::OverrideExp Class Reference	289
6.85.1 Constructor & Destructor Documentation	290
6.85.1.1 OverrideExp	290
6.85.1.2 ~OverrideExp	291
6.85.1.3 ~OverrideExp	291
6.85.2 Member Function Documentation	291
6.85.2.1 getExp	291
6.85.2.2 getTypeld	291
6.85.2.3 instanceof	291
6.85.2.4 recycle	292
6.85.2.5 toString	292
6.86 OvertideExp Class Reference	293
6.86.1 Detailed Description	293
6.87 MM::PoolNodeBehavior Class Reference	293
6.87.1 Constructor & Destructor Documentation	295
6.87.1.1 PoolNodeBehavior	295
6.87.1.2 ~PoolNodeBehavior	295
6.87.2 Member Function Documentation	295
6.87.2.1 add	295
6.87.2.2 addInterface	297
6.87.2.3 begin	297
6.87.2.4 change	298
6.87.2.5 end	299
6.87.2.6 getAdd	299
6.87.2.7 getAmount	299
6.87.2.8 getAt	299
6.87.2.9 getCapacity	299
6.87.2.10 getCreateMessage	300
6.87.2.11 getDefinition	300
6.87.2.12 getDeleteMessage	300
6.87.2.13 getInterface	300
6.87.2.14 getMax	300
6.87.2.15 getOf	300

6.87.2.16 getResources	300
6.87.2.17 getTypeld	301
6.87.2.18 getTypeName	301
6.87.2.19 getUpdateMessage	301
6.87.2.20 hasCapacity	301
6.87.2.21 hasResources	301
6.87.2.22 instanceof	302
6.87.2.23 recycle	302
6.87.2.24 removeInterface	303
6.87.2.25 setAdd	303
6.87.2.26 setAt	303
6.87.2.27 setDefinition	303
6.87.2.28 setMax	303
6.87.2.29 stepPullAll	303
6.87.2.30 stepPushAll	304
6.87.2.31 sub	305
6.87.2.32 toString	306
6.87.2.33 toString	306
6.87.2.34 update	307
6.88 PoolNodeBehavior Class Reference	307
6.88.1 Detailed Description	307
6.89 MM::Prevention Class Reference	307
6.89.1 Constructor & Destructor Documentation	308
6.89.1.1 Prevention	308
6.89.1.2 Prevention	309
6.89.1.3 Prevention	309
6.89.1.4 ~Prevention	309
6.89.2 Member Function Documentation	309
6.89.2.1 getLocation	309
6.89.2.2 getMessage	309
6.89.2.3 getTypeld	309
6.89.2.4 instanceof	309
6.89.2.5 recycle	309
6.89.2.6 toString	309
6.89.2.7 toString	309
6.90 MM::Program Class Reference	310
6.90.1 Constructor & Destructor Documentation	311
6.90.1.1 Program	311
6.90.1.2 ~Program	311
6.90.2 Member Function Documentation	311

6.90.2.1	addTransformation	311
6.90.2.2	getTransformations	311
6.90.2.3	getTypeld	311
6.90.2.4	instanceof	311
6.90.2.5	recycle	312
6.90.2.6	toString	312
6.91	Program Class Reference	313
6.91.1	Detailed Description	313
6.92	MM::RangeValExp Class Reference	313
6.92.1	Constructor & Destructor Documentation	315
6.92.1.1	RangeValExp	315
6.92.1.2	RangeValExp	315
6.92.1.3	~RangeValExp	315
6.92.2	Member Function Documentation	315
6.92.2.1	getIntValue	315
6.92.2.2	getMax	315
6.92.2.3	getMin	316
6.92.2.4	getTypeld	316
6.92.2.5	greaterEquals	316
6.92.2.6	instanceof	316
6.92.2.7	recycle	316
6.92.2.8	toString	317
6.93	RangeValExp Class Reference	317
6.93.1	Detailed Description	317
6.94	MM::Recyclable Class Reference	317
6.94.1	Constructor & Destructor Documentation	319
6.94.1.1	Recyclable	319
6.94.1.2	~Recyclable	319
6.94.2	Member Function Documentation	319
6.94.2.1	getTypeld	319
6.94.2.2	instanceof	319
6.94.2.3	recycle	320
6.94.2.4	toString	322
6.95	Recyclable Class Reference	323
6.95.1	Detailed Description	323
6.96	MM::Recycler Class Reference	323
6.96.1	Constructor & Destructor Documentation	324
6.96.1.1	Recycler	324
6.96.1.2	~Recycler	324
6.96.2	Member Function Documentation	324

6.96.2.1	create	325
6.96.2.2	createBuffer	326
6.96.2.3	getTypeld	326
6.96.2.4	instanceof	326
6.96.2.5	recycle	326
6.96.2.6	uncreate	327
6.96.2.7	uncreate	327
6.96.3	Member Data Documentation	327
6.96.3.1	TYPE_STR	327
6.97	Recycler Class Reference	328
6.97.1	Detailed Description	328
6.98	Reflector Class Reference	328
6.98.1	Detailed Description	328
6.99	MM::Reflector Class Reference	328
6.99.1	Constructor & Destructor Documentation	329
6.99.1.1	Reflector	329
6.99.1.2	~Reflector	329
6.99.2	Member Function Documentation	329
6.99.2.1	addElement	329
6.99.2.2	deinit	329
6.99.2.3	getDefinition	329
6.99.2.4	getInstance	330
6.99.2.5	getTypeld	330
6.99.2.6	init	331
6.99.2.7	init	332
6.99.2.8	instanceof	332
6.99.2.9	merge	333
6.99.2.10	removeElement	333
6.100	RefNodeBehavior Class Reference	333
6.100.1	Detailed Description	333
6.101	MM::RefNodeBehavior Class Reference	334
6.101.1	Constructor & Destructor Documentation	335
6.101.1.1	RefNodeBehavior	335
6.101.1.2	~RefNodeBehavior	335
6.101.2	Member Function Documentation	335
6.101.2.1	add	335
6.101.2.2	begin	336
6.101.2.3	change	336
6.101.2.4	doTriggers	336
6.101.2.5	end	336

6.101.2.6 <code>getAlias</code>	336
6.101.2.7 <code>getCapacity</code>	336
6.101.2.8 <code>getCreateMessage</code>	336
6.101.2.9 <code>getDeleteMessage</code>	336
6.101.2.10 <code>getReference</code>	336
6.101.2.11 <code>getResources</code>	337
6.101.2.12 <code>getTypeld</code>	337
6.101.2.13 <code>getUpdateMessage</code>	337
6.101.2.14 <code>hasCapacity</code>	337
6.101.2.15 <code>hasResources</code>	337
6.101.2.16 <code>instanceof</code>	337
6.101.2.17 <code>recycle</code>	338
6.101.2.18 <code>setAlias</code>	338
6.101.2.19 <code>step</code>	338
6.101.2.20 <code>stepPullAll</code>	338
6.101.2.21 <code>stepPullAny</code>	338
6.101.2.22 <code>stepPushAll</code>	338
6.101.2.23 <code>stepPushAny</code>	338
6.101.2.24 <code>sub</code>	338
6.101.2.25 <code>toString</code>	338
6.101.2.26 <code>toString</code>	339
6.102 <code>SourceNodeBehavior</code> Class Reference	339
6.102.1 Detailed Description	339
6.103 <code>MM::SourceNodeBehavior</code> Class Reference	339
6.103.1 Constructor & Destructor Documentation	341
6.103.1.1 <code>SourceNodeBehavior</code>	341
6.103.1.2 <code>~SourceNodeBehavior</code>	341
6.103.2 Member Function Documentation	341
6.103.2.1 <code>add</code>	341
6.103.2.2 <code>begin</code>	341
6.103.2.3 <code>change</code>	341
6.103.2.4 <code>end</code>	341
6.103.2.5 <code>getCapacity</code>	341
6.103.2.6 <code>getCreateMessage</code>	341
6.103.2.7 <code>getDeleteMessage</code>	342
6.103.2.8 <code>getResources</code>	342
6.103.2.9 <code>getTypeld</code>	342
6.103.2.10 <code>getUpdateMessage</code>	342
6.103.2.11 <code>hasCapacity</code>	342
6.103.2.12 <code>hasResources</code>	342

6.103.2.13 <code>instanceof</code>	342
6.103.2.14 <code>recycle</code>	342
6.103.2.15 <code>stepPullAll</code>	343
6.103.2.16 <code>stepPullAny</code>	343
6.103.2.17 <code>stepPushAll</code>	343
6.103.2.18 <code>sub</code>	344
6.103.2.19 <code>toString</code>	344
6.103.2.20 <code>toString</code>	344
6.104 StateEdge Class Reference	344
6.104.1 Detailed Description	344
6.105 MM::StateEdge Class Reference	345
6.105.1 Constructor & Destructor Documentation	346
6.105.1.1 <code>~StateEdge</code>	346
6.105.1.2 <code>StateEdge</code>	347
6.105.2 Member Function Documentation	347
6.105.2.1 <code>getTypeld</code>	347
6.105.2.2 <code>instanceof</code>	347
6.105.2.3 <code>isAlias</code>	347
6.105.2.4 <code>isCondition</code>	347
6.105.2.5 <code>isTrigger</code>	347
6.105.2.6 <code>recycle</code>	348
6.105.2.7 <code>toString</code>	348
6.105.2.8 <code>toString</code>	348
6.106 String Class Reference	348
6.106.1 Detailed Description	348
6.107 MM::String Class Reference	349
6.107.1 Constructor & Destructor Documentation	350
6.107.1.1 <code>String</code>	350
6.107.1.2 <code>~String</code>	350
6.107.2 Member Function Documentation	350
6.107.2.1 <code>append</code>	350
6.107.2.2 <code>append</code>	351
6.107.2.3 <code>append</code>	351
6.107.2.4 <code>appendInt</code>	352
6.107.2.5 <code>clear</code>	352
6.107.2.6 <code>getBuffer</code>	353
6.107.2.7 <code>getSize</code>	354
6.107.2.8 <code>getTypeld</code>	354
6.107.2.9 <code>getUsed</code>	355
6.107.2.10 <code>instanceof</code>	355

6.107.2.1 <code>\linebreak</code>	355
6.107.2.12 <code>print</code>	356
6.107.2.13 <code>recycle</code>	356
6.107.2.14 <code>space</code>	357
6.107.2.15 <code>space</code>	357
6.107.2.16 <code>toString</code>	357
6.108 Transformation Class Reference	358
6.108.1 Detailed Description	358
6.109 MM::Transformation Class Reference	358
6.109.1 Constructor & Destructor Documentation	359
6.109.1.1 Transformation	359
6.109.1.2 ~Transformation	359
6.109.2 Member Function Documentation	359
6.109.2.1 <code>addElement</code>	360
6.109.2.2 <code>clearElements</code>	360
6.109.2.3 <code>getElements</code>	361
6.109.2.4 <code>getTypeld</code>	361
6.109.2.5 <code>instanceof</code>	361
6.109.2.6 <code>recycle</code>	362
6.109.2.7 <code>toString</code>	362
6.110 MM::Transition Class Reference	363
6.110.1 Constructor & Destructor Documentation	365
6.110.1.1 Transition	365
6.110.1.2 Transition	365
6.110.1.3 ~Transition	365
6.110.2 Member Function Documentation	365
6.110.2.1 <code>getTypeld</code>	365
6.110.2.2 <code>instanceof</code>	365
6.110.2.3 <code>recycle</code>	365
6.110.2.4 <code>toString</code>	366
6.111 Transition Class Reference	367
6.111.1 Detailed Description	367
6.112 TriggerEvent Class Reference	367
6.112.1 Detailed Description	367
6.113 MM::TriggerEvent Class Reference	367
6.113.1 Constructor & Destructor Documentation	369
6.113.1.1 TriggerEvent	369
6.113.1.2 TriggerEvent	369
6.113.1.3 TriggerEvent	369
6.113.1.4 ~TriggerEvent	369

6.113.2 Member Function Documentation	369
6.113.2.1 getLocation	369
6.113.2.2 getMessage	369
6.113.2.3 getTypeld	369
6.113.2.4 instanceof	369
6.113.2.5 recycle	370
6.113.2.6 toString	370
6.113.2.7 toString	370
6.114 TriggerExp Class Reference	370
6.114.1 Detailed Description	370
6.115 MM::TriggerExp Class Reference	371
6.115.1 Constructor & Destructor Documentation	372
6.115.1.1 TriggerExp	372
6.115.1.2 TriggerExp	372
6.115.1.3 ~TriggerExp	372
6.115.2 Member Function Documentation	372
6.115.2.1 getTypeld	372
6.115.2.2 instanceof	373
6.115.2.3 recycle	374
6.115.2.4 toString	374
6.115.3 Member Data Documentation	375
6.115.3.1 TRIGGER_CHAR	375
6.116 UnExp Class Reference	375
6.116.1 Detailed Description	375
6.117 MM::UnExp Class Reference	375
6.117.1 Detailed Description	376
6.117.2 Constructor & Destructor Documentation	376
6.117.2.1 UnExp	376
6.117.2.2 UnExp	376
6.117.2.3 ~UnExp	376
6.117.3 Member Function Documentation	377
6.117.3.1 getExp	377
6.117.3.2 getOperator	377
6.117.3.3 getTypeld	377
6.117.3.4 instanceof	377
6.117.3.5 recycle	378
6.117.3.6 toString	378
6.118 MM::ValExp Class Reference	379
6.118.1 Detailed Description	380
6.118.2 Constructor & Destructor Documentation	380

6.118.2.1 ValExp	380
6.118.2.2 ~ValExp	380
6.118.3 Member Function Documentation	380
6.118.3.1 getTypeld	380
6.118.3.2 greaterEquals	381
6.118.3.3 instanceof	381
6.118.3.4 recycle	382
6.118.3.5 toString	383
6.119 ValExp Class Reference	383
6.119.1 Detailed Description	383
6.120 VarExp Class Reference	384
6.120.1 Detailed Description	384
6.121 MM::VarExp Class Reference	384
6.121.1 Detailed Description	385
6.121.2 Constructor & Destructor Documentation	385
6.121.2.1 VarExp	385
6.121.2.2 ~VarExp	386
6.121.3 Member Function Documentation	386
6.121.3.1 getName	386
6.121.3.2 getTypeld	386
6.121.3.3 instanceof	386
6.121.3.4 recycle	387
6.121.3.5 toString	387
6.122 MM::Vector< T > Class Template Reference	387
6.122.1 Constructor & Destructor Documentation	388
6.122.1.1 Vector	388
6.122.1.2 ~Vector	388
6.122.2 Member Function Documentation	388
6.122.2.1 add	389
6.122.2.2 addAll	389
6.122.2.3 at	389
6.122.2.4 clear	389
6.122.2.5 contains	389
6.122.2.6 elementAt	390
6.122.2.7 getIterator	391
6.122.2.8 getNewIterator	391
6.122.2.9 getPosition	391
6.122.2.10 getTypeld	392
6.122.2.11 instanceof	392
6.122.2.12 isEmpty	392

6.122.2.13pop	392
6.122.2.14remove	392
6.122.2.15remove	392
6.122.2.16size	393
6.123MM::Violation Class Reference	393
6.123.1 Constructor & Destructor Documentation	394
6.123.1.1 Violation	395
6.123.1.2 Violation	395
6.123.1.3 Violation	395
6.123.1.4 ~Violation	395
6.123.2 Member Function Documentation	395
6.123.2.1 getLocation	395
6.123.2.2 getMessage	395
6.123.2.3 getTypeld	395
6.123.2.4 instanceof	395
6.123.2.5 recycle	395
6.123.2.6 toString	395
6.123.2.7 toString	395
6.123.3 Member Data Documentation	396
6.123.3.1 VIOLATE_LEN	396
6.123.3.2 VIOLATE_STR	396
6.124Violation Class Reference	396
6.124.1 Detailed Description	396
6.125yy_buffer_state Struct Reference	396
6.125.1 Member Data Documentation	397
6.125.1.1 yy_at_bol	397
6.125.1.2 yy_bs_column	397
6.125.1.3 yy_bs_lineno	397
6.125.1.4 yy_buf_pos	397
6.125.1.5 yy_buf_size	397
6.125.1.6 yy_buffer_status	397
6.125.1.7 yy_ch_buf	397
6.125.1.8 yy_fill_buffer	397
6.125.1.9 yy_input_file	397
6.125.1.10yy_is_interactive	397
6.125.1.11yy_is_our_buffer	397
6.125.1.12yy_n_chars	397
6.126yy_trans_info Struct Reference	397
6.126.1 Member Data Documentation	397
6.126.1.1 yy_nxt	397

6.126.1.2 <code>yy_verify</code>	397
6.127 <code>yyalloc</code> Union Reference	398
6.127.1 Member Data Documentation	398
6.127.1.1 <code>yyls_alloc</code>	398
6.127.1.2 <code>yyss_alloc</code>	398
6.127.1.3 <code>yyvs_alloc</code>	398
6.128 <code>YYLTYPE</code> Struct Reference	398
6.128.1 Member Data Documentation	398
6.128.1.1 <code>first_column</code>	398
6.128.1.2 <code>first_line</code>	399
6.128.1.3 <code>last_column</code>	399
6.128.1.4 <code>last_line</code>	399
6.129 <code>YYSTYPE</code> Union Reference	399
6.129.1 Member Data Documentation	399
6.129.1.1 <code>act</code>	399
6.129.1.2 <code>element</code>	400
6.129.1.3 <code>eList</code>	400
6.129.1.4 <code>exp</code>	400
6.129.1.5 <code>how</code>	400
6.129.1.6 <code>io</code>	400
6.129.1.7 <code>name</code>	400
6.129.1.8 <code>program</code>	400
6.129.1.9 <code>str</code>	400
6.129.1.10 <code>t</code>	400
6.129.1.11 <code>tList</code>	400
6.129.1.12 <code>val</code>	400
6.129.1.13 <code>when</code>	400
7 File Documentation	401
7.1 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Activation.cpp File Reference	401
7.2 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Activation.h File Reference	402
7.3 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/ActiveExp.cpp File Reference	402
7.4 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/ActiveExp.h File Reference	402
7.4.1 Detailed Description	403
7.5 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/AliasExp.cpp File Reference	403
7.5.1 Detailed Description	404
7.6 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/AliasExp.h File Reference	404
7.6.1 Detailed Description	404
7.7 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/AllExp.cpp File Reference	405
7.7.1 Detailed Description	405

7.8 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/AllExp.h File Reference	405
7.8.1 Detailed Description	406
7.9 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Assertion.cpp File Reference	406
7.10 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Assertion.h File Reference	406
7.10.1 Detailed Description	407
7.11 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/BinExp.cpp File Reference	407
7.12 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/BinExp.h File Reference	407
7.12.1 Detailed Description	408
7.13 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/BooleanValExp.cpp File Reference	408
7.13.1 Detailed Description	409
7.14 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/BooleanValExp.h File Reference	409
7.14.1 Detailed Description	409
7.15 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/ConverterNodeBehavior.cpp File Reference	411
7.16 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/ConverterNodeBehavior.h File Reference	412
7.16.1 Detailed Description	412
7.17 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Declaration.cpp File Reference	414
7.18 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Declaration.h File Reference	415
7.18.1 Detailed Description	415
7.19 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Definition.cpp File Reference	415
7.20 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Definition.h File Reference	416
7.20.1 Detailed Description	416
7.21 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Deletion.cpp File Reference	416
7.22 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Deletion.h File Reference	417
7.22.1 Detailed Description	417
7.23 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/DieExp.cpp File Reference	417
7.23.1 Detailed Description	418
7.24 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/DieExp.h File Reference	418
7.24.1 Detailed Description	418
7.25 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Disablement.cpp File Reference	419
7.26 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Disablement.h File Reference	419
7.26.1 Detailed Description	420
7.27 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/DrainNodeBehavior.cpp File Reference	421
7.28 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/DrainNodeBehavior.h File Reference	422
7.28.1 Detailed Description	422
7.29 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Edge.cpp File Reference	422
7.30 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Edge.h File Reference	423
7.30.1 Detailed Description	423
7.31 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Element.cpp File Reference	423
7.31.1 Detailed Description	424
7.32 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Element.h File Reference	424

7.32.1 Detailed Description	424
7.33 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Enablement.cpp File Reference	425
7.34 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Enablement.h File Reference	425
7.34.1 Detailed Description	426
7.35 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Evaluator.cpp File Reference	427
7.36 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Evaluator.h File Reference	428
7.36.1 Detailed Description	428
7.37 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Event.cpp File Reference	428
7.37.1 Detailed Description	429
7.38 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Event.h File Reference	429
7.38.1 Detailed Description	429
7.39 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Exp.cpp File Reference	429
7.39.1 Detailed Description	430
7.40 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Exp.h File Reference	430
7.40.1 Detailed Description	430
7.41 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Failure.cpp File Reference	431
7.42 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Failure.h File Reference	431
7.43 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/FlowEdge.cpp File Reference	432
7.44 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/FlowEdge.h File Reference	432
7.44.1 Detailed Description	432
7.45 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/FlowEvent.cpp File Reference	433
7.46 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/FlowEvent.h File Reference	433
7.46.1 Detailed Description	434
7.47 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/GateNodeBehavior.cpp File Reference	434
7.48 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/GateNodeBehavior.h File Reference	434
7.48.1 Detailed Description	435
7.49 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Instance.cpp File Reference	436
7.49.1 Detailed Description	437
7.50 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Instance.h File Reference	437
7.50.1 Detailed Description	437
7.51 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/InterfaceNode.cpp File Reference	438
7.52 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/InterfaceNode.h File Reference	438
7.52.1 Detailed Description	439
7.53 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/lex.mm.cpp File Reference	439
7.53.1 Macro Definition Documentation	442
7.53.1.1 BEGIN	442
7.53.1.2 ECHO	442
7.53.1.3 EOB_ACT_CONTINUE_SCAN	442
7.53.1.4 EOB_ACT_END_OF_FILE	442
7.53.1.5 EOB_ACT_LAST_MATCH	442

7.53.1.6 ERROR	442
7.53.1.7 FLEX_BETA	442
7.53.1.8 FLEX_SCANNER	442
7.53.1.9 FLEXINT_H	442
7.53.1.10 IDPART	442
7.53.1.11 IDPART2	442
7.53.1.12 IN_M_COMMENT	442
7.53.1.13 IN_S_COMMENT	442
7.53.1.14 INITIAL	442
7.53.1.15 INT16_MAX	442
7.53.1.16 INT16_MIN	442
7.53.1.17 INT32_MAX	442
7.53.1.18 INT32_MIN	442
7.53.1.19 INT8_MAX	443
7.53.1.20 INT8_MIN	443
7.53.1.21 LEX_ID_SIZE	443
7.53.1.22 REJECT	443
7.53.1.23 UINT16_MAX	443
7.53.1.24 UINT32_MAX	443
7.53.1.25 UINT8_MAX	443
7.53.1.26 unput	443
7.53.1.27 YY_AT_BOL	443
7.53.1.28 YY_BREAK	443
7.53.1.29 YY_BUF_SIZE	443
7.53.1.30 YY_BUFFER_EOF_PENDING	443
7.53.1.31 YY_BUFFER_NEW	443
7.53.1.32 YY_BUFFER_NORMAL	443
7.53.1.33 YY_CURRENT_BUFFER	443
7.53.1.34 YY_CURRENT_BUFFER_LVALUE	443
7.53.1.35 YY_DECL	443
7.53.1.36 YY_DECL_IS_OURS	443
7.53.1.37 YY_DO_BEFORE_ACTION	443
7.53.1.38 YY_END_OF_BUFFER	443
7.53.1.39 YY_END_OF_BUFFER_CHAR	443
7.53.1.40 YY_EXIT_FAILURE	444
7.53.1.41 YY_EXTRA_TYPE	444
7.53.1.42 YY_FATAL_ERROR	444
7.53.1.43 YY_FLEX_MAJOR_VERSION	444
7.53.1.44 YY_FLEX_MINOR_VERSION	444
7.53.1.45 YY_FLEX_SUBMINOR_VERSION	444

7.53.1.46 YY_FLUSH_BUFFER	444
7.53.1.47 YY_INPUT	444
7.53.1.48 YY_INT_ALIGNED	444
7.53.1.49 YY_LESS_LINENO	444
7.53.1.50 YY_MORE_ADJ	444
7.53.1.51 yy_new_buffer	444
7.53.1.52 YY_NEW_FILE	444
7.53.1.53 YY_NULL	444
7.53.1.54 YY_NUM_RULES	444
7.53.1.55 YY_READ_BUF_SIZE	444
7.53.1.56 YY_RESTORE YY_MORE_OFFSET	444
7.53.1.57 YY_RULE_SETUP	445
7.53.1.58 YY_SC_TO_UI	445
7.53.1.59 yy_set_bol	445
7.53.1.60 yy_set_interactive	445
7.53.1.61 YY_SKIP YYWRAP	445
7.53.1.62 YY_START	445
7.53.1.63 YY_START_STACK_INCR	445
7.53.1.64 YY_STATE_BUF_SIZE	445
7.53.1.65 YY_STATE_EOF	445
7.53.1.66 YY_STRUCT YY_BUFFER_STATE	445
7.53.1.67 YY_TYPEDEF YY_BUFFER_STATE	445
7.53.1.68 YY_TYPEDEF YY_SIZE_T	445
7.53.1.69 YY_USER_ACTION	445
7.53.1.70 yyconst	446
7.53.1.71 yyless	446
7.53.1.72 yymore	446
7.53.1.73 yymore	446
7.53.1.74 YYSTATE	446
7.53.1.75 YYTABLES_NAME	446
7.53.1.76 yyterminate	446
7.53.1.77 yytext_ptr	446
7.53.1.78 yywrap	446
7.53.2 Typedef Documentation	446
7.53.2.1 flex_int16_t	446
7.53.2.2 flex_int32_t	446
7.53.2.3 flex_int8_t	446
7.53.2.4 flex_uint16_t	446
7.53.2.5 flex_uint32_t	446
7.53.2.6 flex_uint8_t	446

7.53.2.7 YY_BUFFER_STATE	447
7.53.2.8 YY_CHAR	447
7.53.2.9 yy_size_t	447
7.53.2.10 yy_state_type	447
7.53.3 Function Documentation	447
7.53.3.1 if	447
7.53.3.2 isatty	447
7.53.3.3 while	447
7.53.3.4 yy_create_buffer	447
7.53.3.5 yy_delete_buffer	448
7.53.3.6 yy_flush_buffer	448
7.53.3.7 yy_scan_buffer	449
7.53.3.8 yy_scan_bytes	449
7.53.3.9 yy_scan_string	450
7.53.3.10 yy_switch_to_buffer	451
7.53.3.11 yyalloc	451
7.53.3.12 yyfree	452
7.53.3.13 yyget_debug	452
7.53.3.14 yyget_extra	452
7.53.3.15 yyget_in	452
7.53.3.16 yyget_leng	452
7.53.3.17 yyget_lineno	452
7.53.3.18 yyget_out	452
7.53.3.19 yyget_text	452
7.53.3.20 yylex	452
7.53.3.21 yylex_destroy	453
7.53.3.22 yypop_buffer_state	453
7.53.3.23 yypush_buffer_state	453
7.53.3.24 yyrealloc	453
7.53.3.25 yyrestart	454
7.53.3.26 yyset_debug	455
7.53.3.27 yyset_extra	455
7.53.3.28 yyset_in	455
7.53.3.29 yyset_lineno	455
7.53.3.30 yyset_out	455
7.53.4 Variable Documentation	455
7.53.4.1 inID	455
7.53.4.2 yy_act	455
7.53.4.3 yy_bp	455
7.53.4.4 yy_cp	456

7.53.4.5 YY_DECL	456
7.53.4.6 yy flex_debug	456
7.53.4.7 yycolumn	456
7.53.4.8 yyid	456
7.53.4.9 yyidpos	456
7.53.4.10 yyin	456
7.53.4.11 yyleng	456
7.53.4.12 yylineno	456
7.53.4.13 yyout	456
7.53.4.14 yystr	456
7.53.4.15 yytext	456
7.54 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Location.cpp File Reference	456
7.54.1 Detailed Description	457
7.55 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Location.h File Reference	457
7.55.1 Detailed Description	457
7.56 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Machine.cpp File Reference	459
7.56.1 Function Documentation	460
7.56.1.1 MM_parse	461
7.56.1.2 MM_parseFile	463
7.57 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Machine.h File Reference	464
7.57.1 Detailed Description	464
7.58 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Map.h File Reference	464
7.58.1 Detailed Description	465
7.59 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Message.cs File Reference	465
7.59.1 Detailed Description	466
7.60 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/mm.h File Reference	467
7.60.1 Detailed Description	468
7.61 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/mm.tab.cpp File Reference	468
7.61.1 Macro Definition Documentation	471
7.61.1.1 YY_	471
7.61.1.2 YY_LOCATION_PRINT	471
7.61.1.3 YY_REDUCE_PRINT	471
7.61.1.4 YY_STACK_PRINT	471
7.61.1.5 YY_SYMBOL_PRINT	471
7.61.1.6 YYABORT	471
7.61.1.7 YYACCEPT	471
7.61.1.8 YYBACKUP	471
7.61.1.9 YYBISON	472
7.61.1.10 YYBISON_VERSION	472
7.61.1.11 yyclearin	472

7.61.1.12 YYCOPY	472
7.61.1.13 YYDEBUG	472
7.61.1.14 YYDPRINTF	472
7.61.1.15 YYEMPTY	472
7.61.1.16 YYEOF	472
7.61.1.17 YYERRCODE	472
7.61.1.18 yyerrok	472
7.61.1.19 YYERROR	472
7.61.1.20 YYERROR_VERBOSE	472
7.61.1.21 YYFAIL	472
7.61.1.22 YYFINAL	472
7.61.1.23 YYFREE	472
7.61.1.24 YYID	472
7.61.1.25 YYINITDEPTH	472
7.61.1.26 YYLAST	472
7.61.1.27 YYLEX	472
7.61.1.28 YYLLOC_DEFAULT	472
7.61.1.29 YYLSP_NEEDED	473
7.61.1.30 YYMALLOC	473
7.61.1.31 YYMAXDEPTH	473
7.61.1.32 YYMAXUTOK	473
7.61.1.33 YYNNTS	473
7.61.1.34 YYNRULES	473
7.61.1.35 YYNSTATES	473
7.61.1.36 YYNTOKENS	473
7.61.1.37 YYPACT_NINF	473
7.61.1.38 YYPOPSTACK	473
7.61.1.39 YYPULL	473
7.61.1.40 YYPURE	473
7.61.1.41 YYPUSH	473
7.61.1.42 YYRECOVERING	473
7.61.1.43 YYRHSLOC	473
7.61.1.44 YYSIZE_MAXIMUM	473
7.61.1.45 YYSIZE_T	473
7.61.1.46 YYSKELETON_NAME	473
7.61.1.47 YYSTACK_ALLOC	473
7.61.1.48 YYSTACK_ALLOC_MAXIMUM	473
7.61.1.49 YYSTACK_BYTES	473
7.61.1.50 YYSTACK_FREE	474
7.61.1.51 YYSTACK_GAP_MAXIMUM	474

7.61.1.52 YYSTACK_RELOCATE	474
7.61.1.53 yystype	474
7.61.1.54 YYSTYPE_IS_DECLARED	474
7.61.1.55 YYSTYPE_IS_TRIVIAL	474
7.61.1.56 YYTABLE_NINF	474
7.61.1.57 YYTERROR	474
7.61.1.58 YYTOKEN_TABLE	474
7.61.1.59 YYTOKENTYPE	474
7.61.1.60 YYTRANSLATE	474
7.61.1.61 YYUNDEFTOK	474
7.61.1.62 YYUSE	474
7.61.2 Typedef Documentation	474
7.61.2.1 YY_BUFFER_STATE	474
7.61.2.2 YYSTYPE	474
7.61.2.3 yytype_int16	474
7.61.2.4 yytype_int8	474
7.61.2.5 yytype_uint16	474
7.61.2.6 yytype_uint8	474
7.61.3 Enumeration Type Documentation	474
7.61.3.1 yytokentype	474
7.61.4 Function Documentation	478
7.61.4.1 for	478
7.61.4.2 if	478
7.61.4.3 main	478
7.61.4.4 MM_parse	479
7.61.4.5 MM_parseFile	481
7.61.4.6 switch	482
7.61.4.7 while	482
7.61.4.8 yy_scan_buffer	482
7.61.4.9 YY_SYMBOL_PRINT	483
7.61.4.10 yyerror	483
7.61.4.11 yylex	483
7.61.4.12 yyparse	484
7.61.4.13 YYUSE	485
7.61.5 Variable Documentation	485
7.61.5.1 mm	485
7.61.5.2 program	485
7.61.5.3 yychar	485
7.61.5.4 yycolumn	485
7.61.5.5 yyd	485

7.61.5.6 yyin	485
7.61.5.7 yylen	485
7.61.5.8 yylineno	485
7.61.5.9 yyloc	485
7.61.5.10 yylocationp	485
7.61.5.11 yyval	485
7.61.5.12 yynerrs	485
7.61.5.13 yys	485
7.61.5.14 yysrc	485
7.61.5.15 yytype	485
7.61.5.16 yyvaluep	485
7.62 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/mm.tab.hpp File Reference	485
7.62.1 Macro Definition Documentation	487
7.62.1.1 yytype	487
7.62.1.2 YYLTYPE_IS_DECLARED	487
7.62.1.3 YYLTYPE_IS_TRIVIAL	487
7.62.1.4 yystype	487
7.62.1.5 YYSTYPE_IS_DECLARED	487
7.62.1.6 YYSTYPE_IS_TRIVIAL	487
7.62.2 Typedef Documentation	487
7.62.2.1 YYLTYPE	487
7.62.2.2 YYSTYPE	487
7.62.3 Enumeration Type Documentation	487
7.62.3.1 yytokentype	487
7.62.4 Variable Documentation	490
7.62.4.1 yyloc	490
7.62.4.2 yyval	490
7.63 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Modification.cpp File Reference	490
7.64 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Modification.h File Reference	491
7.64.1 Detailed Description	491
7.65 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Name.cpp File Reference	491
7.65.1 Detailed Description	492
7.66 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Name.h File Reference	492
7.66.1 Detailed Description	492
7.67 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Node.cpp File Reference	494
7.68 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Node.h File Reference	495
7.68.1 Detailed Description	495
7.69 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/NodeBehavior.cpp File Reference	496
7.70 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/NodeBehavior.h File Reference	497
7.70.1 Detailed Description	497

7.71 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/NodeWorkItem.cpp File Reference	497
7.72 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/NodeWorkItem.h File Reference	498
7.72.1 Detailed Description	498
7.73 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/NumberValExp.cpp File Reference	498
7.73.1 Detailed Description	499
7.74 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/NumberValExp.h File Reference	499
7.74.1 Detailed Description	499
7.75 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Observable.cpp File Reference	499
7.76 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Observable.h File Reference	500
7.76.1 Detailed Description	500
7.77 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Observer.cpp File Reference	500
7.78 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Observer.h File Reference	501
7.78.1 Detailed Description	501
7.79 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/OneExp.cpp File Reference	502
7.80 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/OneExp.h File Reference	502
7.80.1 Detailed Description	502
7.81 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Operator.cpp File Reference	503
7.81.1 Detailed Description	503
7.82 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Operator.h File Reference	503
7.82.1 Detailed Description	504
7.83 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/OverrideExp.cpp File Reference	504
7.83.1 Detailed Description	504
7.84 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/OverrideExp.h File Reference	505
7.84.1 Detailed Description	505
7.85 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/PoolNodeBehavior.cpp File Reference	506
7.86 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/PoolNodeBehavior.h File Reference	507
7.86.1 Detailed Description	507
7.87 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Prevention.cpp File Reference	508
7.88 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Prevention.h File Reference	508
7.89 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Program.cpp File Reference	509
7.90 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Program.h File Reference	509
7.90.1 Detailed Description	509
7.91 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/RangeValExp.cpp File Reference	510
7.91.1 Detailed Description	510
7.92 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/RangeValExp.h File Reference	510
7.92.1 Detailed Description	511
7.93 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Recyclable.cpp File Reference	511
7.94 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Recyclable.h File Reference	511
7.94.1 Detailed Description	512
7.95 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Recycler.cpp File Reference	512

7.96 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Recycler.h File Reference	512
7.96.1 Detailed Description	512
7.97 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Reflector.cpp File Reference	514
7.98 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Reflector.h File Reference	515
7.98.1 Detailed Description	515
7.99 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/RefNodeBehavior.cpp File Reference	516
7.100 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/RefNodeBehavior.h File Reference	516
7.100.1 Detailed Description	517
7.101 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/SourceNodeBehavior.cpp File Reference	518
7.102 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/SourceNodeBehavior.h File Reference	519
7.102.1 Detailed Description	519
7.103 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/StateEdge.cpp File Reference	519
7.104 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/StateEdge.h File Reference	520
7.104.1 Detailed Description	520
7.105 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/String.cpp File Reference	520
7.106 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/String.h File Reference	521
7.106.1 Detailed Description	521
7.107 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Transformation.cpp File Reference	522
7.108 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Transformation.h File Reference	522
7.108.1 Detailed Description	522
7.109 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Transition.cpp File Reference	523
7.110 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Transition.h File Reference	523
7.110.1 Detailed Description	523
7.111 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/TriggerEvent.cpp File Reference	524
7.112 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/TriggerEvent.h File Reference	524
7.112.1 Detailed Description	525
7.113 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/TriggerExp.cpp File Reference	525
7.113.1 Detailed Description	525
7.114 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/TriggerExp.h File Reference	526
7.114.1 Detailed Description	526
7.115 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Types.h File Reference	527
7.115.1 Detailed Description	528
7.115.2 Macro Definition Documentation	528
7.115.2.1 MM_FALSE	528
7.115.2.2 MM_MAX_RESOURCES	528
7.115.2.3 MM_NULL	528
7.115.2.4 MM_printf	528
7.115.2.5 MM_TRUE	528
7.116 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/UnExp.cpp File Reference	528
7.116.1 Detailed Description	529

7.117C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/UnExp.h File Reference	529
7.117.1 Detailed Description	530
7.118C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/ValExp.cpp File Reference	530
7.118.1 Detailed Description	530
7.119C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/ValExp.h File Reference	530
7.119.1 Detailed Description	531
7.120C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/VarExp.cpp File Reference	531
7.120.1 Detailed Description	531
7.121C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/VarExp.h File Reference	532
7.121.1 Detailed Description	532
7.122C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Vector.h File Reference	532
7.123C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Violation.cpp File Reference	533
7.124C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Violation.h File Reference	534
7.124.1 Detailed Description	534
7.125C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/YYLTYPE.h File Reference	534
7.125.1 Detailed Description	534
7.125.2 Macro Definition Documentation	535
7.125.2.1 YYLTYPE_IS_DECLARED	535
7.125.3 Typedef Documentation	535
7.125.3.1 YYLTYPE	535
Index	536

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

LibMM	17
MM	19

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Activation	28
ActiveExp	32
AliasExp	32
AllExp	39
Assertion	39
BinExp	42
BooleanValExp	47
MM::Name::Compare	50
MM::Node::Compare	51
ConverterNodeBehavior	56
Declaration	63
Definition	73
Deletion	76
DieExp	79
Disablement	83
DrainNodeBehavior	87
Edge	93
Element	107
Enablement	110
Evaluator	121
Event	127
Exp	132
FailEvent	132
FlowEdge	138
FlowEvent	138
GateNodeBehavior	143
Instance	148
InterfaceNode	174
MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::Iterator	179
MM::Vector< T >::Iterator	181
Location	184
Machine	188
MM::Map< MAP_KEY, MAP_VALUE, COMPARE >	227
Map	230
MM::Map< MM::MM::Element *, MM::MM::Vector< MM::MM::Instance * > * >	227
MM::Map< MM::MM::Exp *, MM::INT32 >	227
MM::Map< MM::MM::Name *, MM::MM::Element *, MM::MM::Name::Compare >	227
MM::Map< MM::MM::Name *, MM::MM::Node *, MM::MM::Name::Compare >	227

MM::Map< MM::MM::Node *, MM::MM::Vector< MM::Edge * >::Iterator * >	227
MM::Map< MM::MM::Node *, MM::UINT32 >	227
Modification	233
Name	243
Node	256
NodeBehavior	271
MM::NodeWorkItem	271
NodeWorkItem	273
NumberValExp	273
MM::Observable	278
MM::Definition	63
MM::Instance	149
Observable	281
Observer	281
MM::Observer	281
MM::Declaration	57
MM::Instance	149
MM::Machine::Delegate	74
MM::Machine::InstanceObserver	172
MM::PoolNodeBehavior	293
OneExp	287
MM::Operator	287
Operator	289
OvertideExp	293
PoolNodeBehavior	307
Program	313
RangeValExp	317
MM::Recyclable	317
MM::Element	99
MM::Assertion	39
MM::Declaration	57
MM::Definition	63
MM::Deletion	76
MM::Edge	93
MM::FlowEdge	135
MM::StateEdge	345
MM::Event	122
MM::Activation	25
MM::Disablement	84
MM::Enablement	107
MM::Failure	132
MM::FlowEvent	139
MM::Prevention	307
MM::TriggerEvent	367
MM::Violation	393
MM::Node	243
MM::InterfaceNode	175
MM::Evaluator	111
MM::Exp	127
MM::ActiveExp	28
MM::AliasExp	32
MM::AllExp	36
MM::BinExp	42
MM::DieExp	79
MM::OverrideExp	289
MM::TriggerExp	371
MM::UnExp	375

MM::ValExp	379
MM::BooleanValExp	47
MM::NumberValExp	274
MM::OneExp	283
MM::RangeValExp	313
MM::VarExp	384
MM::Instance	149
MM::Location	184
MM::Name	233
MM::NodeBehavior	257
MM::ConverterNodeBehavior	51
MM::DrainNodeBehavior	87
MM::GateNodeBehavior	143
MM::PoolNodeBehavior	293
MM::RefNodeBehavior	334
MM::SourceNodeBehavior	339
MM::Program	310
MM::String	349
MM::Transformation	358
MM::Modification	230
MM::Transition	363
Recyclable	323
MM::Recycler	323
MM::Machine	188
Recycler	328
Reflector	328
MM::Reflector	328
RefNodeBehavior	333
SourceNodeBehavior	339
StateEdge	344
String	348
Transformation	358
Transition	367
TriggerEvent	367
TriggerExp	370
UnExp	375
ValExp	383
VarExp	384
MM::Vector< T >	387
MM::Vector< MM::Element * >	387
MM::Vector< MM::MM::Edge * >	387
MM::Vector< MM::MM::Element * >	387
MM::Vector< MM::MM::Evaluator::NodeInstance * >	387
MM::Vector< MM::MM::Machine::MM::Machine::Delegate * >	387
MM::Vector< MM::MM::Node * >	387
MM::Vector< MM::MM::Recyclable * >	387
MM::Vector< MM::Observer * >	387
MM::Vector< MM::Transformation * >	387
Violation	396
yy_buffer_state	396
yy_trans_info	397
yyalloc	398
YYLTYPE	398
YYSTYPE	399

Chapter 3

Class Index

3.1 Class List

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

MM::Activation	25
Activation	
The Activation abstraction is a transition event that expresses a node instance is activated	28
MM::ActiveExp	28
ActiveExp	
The ActiveExp abstraction is a program element which expresses that a target node instance of a state edge with an ActiveExp is only active if the source node instance is active	32
AliasExp	
The AliasExp abstraction expresses that a source of a state edge that has an AliasExp is an alias of the target of that edge	32
MM::AliasExp	32
MM::AllExp	
The AllExp abstraction expresses a that the full available amount of resources will flow along a FlowEdge from a source node instance to a target node instance	36
AllExp	
The AllExp abstraction expresses a that the full available amount of resources will flow along a FlowEdge from a source node instance to a target node instance	39
Assertion	
The Assertion abstraction is a program element that generates Signals when its expression is false in some instance	39
MM::Assertion	39
BinExp	
The BinExp abstraction defines binary expressions	42
MM::BinExp	42
BooleanValExp	
The BooleanValExp abstraction defines boolean value expressions	47
MM::BooleanValExp	47
MM::Name::Compare	50
MM::Node::Compare	51
MM::ConverterNodeBehavior	
The ConverterNodeBehavior abstraction defines the behavior of converter nodes	51
ConverterNodeBehavior	
The ConverterNodeBehavior abstraction defines the behavior of converter nodes	56
MM::Declaration	57
Declaration	
The Declaration abstraction defines that a single named instance of a specified definition exists in each instance of the definition that contains the declaration	63
MM::Definition	63
Definition	
The definition abstraction and the elements contained within it define the structure and behavior of instances created for the global scope, instance pools or declarations	73

MM::Machine::Delegate	74
Deletion	
Deletion elements enable deleting elements from definitions using the qualified name, through evaluation between steps	76
MM::Deletion	76
DieExp	
The DieExp abstraction enables simulating a roll of a die, and can be used to randomize the amount of resources that can or must flow along a resource connection (flow edge)	79
MM::DieExp	
Defines the DieExp class	79
Disablement	
The Disable abstraction defines a node instance was disabled during a step because one of its conditions is false	83
MM::Disablement	84
DrainNodeBehavior	
The DrainNodeBehavior abstraction defines the behavior of drain nodes	87
MM::DrainNodeBehavior	87
Edge	
The Edge abstraction is an abstract element connecting a source node and a target node	93
MM::Edge	93
MM::Element	
The Element abstraction is the abstract superclass of all program elements	99
MM::Element	99
Enablement	
The Enablement abstraction defines a node instance was enabled during a step because all of its conditions are true	110
MM::Evaluator	111
Evaluator	
The Evaluator evaluates program steps and the resulting changes to instances	121
MM::Event	122
Event	
The Event abstraction is the abstract superclass of all transition elements	127
MM::Exp	127
Exp	
The Exp abstraction is the abstract super class of all expessions	132
FailEvent	
The Failure abstraction defines a node instance failed during a step	132
MM::Failure	132
MM::FlowEdge	
The FlowEdge abstraction is a program element that defines resource connections, where the amount of resources that may or must flow between source and target node instances is defined by its expression	135
FlowEdge	
The FlowEdge abstraction is a program element that defines resource connections, where the amount of resources that may or must flow between source and target node instances is defined by its expression	138
FlowEvent	
The FlowEvent abstraction defines the amount of resources that flows between source and target node instances during a step	138
MM::FlowEvent	139
GateNodeBehavior	
The GateNodeBehavior abstraction defines the behavior of gate nodes	143
MM::GateNodeBehavior	143
Instance	
The instance abstraction defines instance data that is defined by its definition, and is manipulated by the evaluator during steps	148
MM::Instance	149
MM::Machine::InstanceObserver	172

InterfaceNode	
The InterfaceNode abstraction defines interface nodes on declarations and instance pools that result from observable nodes inside their respective definitions	174
MM::InterfaceNode	175
MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::Iterator	179
MM::Vector< T >::Iterator	181
Location	
The Location abstraction defines textual source locations	184
MM::Location	184
Machine	
The Machine abstraction manages the creation of model elements, such that it can recycle and reuse them	188
MM::Machine	188
MM::Map< MAP_KEY, MAP_VALUE, COMPARE >	227
Map	
The Map abstraction enables quickly looking up stored values using a storage key	230
MM::Modification	230
Modification	
The Modification abstraction consists of elements that are modifications made by the reflector to definitions between steps	233
MM::Name	233
Name	
The Name abstraction is used to compose identifiers into qualified names used to lookup named language elements	243
MM::Node	243
Node	
The Node abstraction defines program nodes that are connected by edges, and its behavior can be modified by changing its node behavior	256
MM::NodeBehavior	257
NodeBehavior	
The NodeBehavior abstraction is the abstract super class of all node behaviors and makes node behavior a modifiable strategy	271
MM::NodeWorkItem	271
NodeWorkItem	
The NodeWorkItem abstraction defines a piece of work to be acted out by a node in an instance on an edge	273
NumberValExp	
The NumberValExp abstraction defines number value expressions	273
MM::NumberValExp	274
MM::Observable	
Observable	
The Observable abstraction enables notifying Observers of changes to Observable objects when notifyObservers is called	281
Observer	
The Observer abstraction enables observing changes to Obsevable objects via the update call-back	281
MM::Observer	281
MM::OneExp	
OneExp	
The OneExp abstraction expresses shorthand expressions on edges. \rightarrow : OneExp has a flow of one: NumberValueExp(1) . * .> : BinExp has lhs OneExp and rhs OneExp, then isTrigger() = MM_TRUE . == .> : BinExp has lhs OneExp and rhs OneExp: lhs OneExp refers to source and rhs OneExp refers to target	287
MM::Operator	287
Operator	
The Operator abstraction defines operator lexicals and codes	289
MM::OverrideExp	289

OvertideExp	
The <code>OvertideExp</code> abstraction expresses an overriden expression	293
MM::PoolNodeBehavior	293
PoolNodeBehavior	
The <code>PoolNodeBehavior</code> abstraction defines the behavior of pool nodes	307
MM::Prevention	307
MM::Program	310
Program	
A <code>Program</code> is a collection of elements that define state and behavior	313
MM::RangeValExp	313
RangeValExp	
The <code>RangeValExp</code> abstraction defines range value expressions	317
MM::Recyclable	317
Recyclable	
Objects that are recyclable can be recycled by their recycler, which is the factory in which the object was created	323
MM::Recycler	323
Recycler	
The <code>Recycler</code> abstraction is used to recycle recyclable objects	328
Reflector	
The <code>Reflector</code> reflects program changes to definitions and instances	328
MM::Reflector	328
RefNodeBehavior	
The <code>RefNodeBehavior</code> abstraction expresses that the behavior of a node is defined elsewhere	333
MM::RefNodeBehavior	334
SourceNodeBehavior	
The <code>SourceNodeBehavior</code> abstraction defines the behavior of source nodes	339
MM::SourceNodeBehavior	339
StateEdge	
The <code>StateEdge</code> abstraction defines state edges, which are conditions, aliases or triggers	344
MM::StateEdge	345
String	
The <code>String</code> abstraction defines bounded <code>String</code> buffers	348
MM::String	349
Transformation	
The <code>Transformation</code> abstraction is the abstract super class of all model transformations	358
MM::Transformation	358
MM::Transition	
The <code>MM::Transition</code> abstraction	363
Transition	
The <code>Transition</code> abstraction trace elements that define steps	367
TriggerEvent	
The <code>TriggerEvent</code> abstraction defines a trigger fired	367
MM::TriggerEvent	367
TriggerExp	
The <code>TriggerExp</code> abstraction expresses that a source of a state edge activates the target of that edge if all of the edge the source operates on are satisfied (meaning a flow of one or more exists for that edge)	370
MM::TriggerExp	371
UnExp	
The <code>UnExp</code> abstraction defines unary expressions	375
MM::UnExp	
Defines the <code>UnExp</code> class	375
MM::ValExp	
Defines the <code>ValExp</code> class	379
ValExp	
The <code>ValExp</code> abstraction defines the abstract super class of value expressions	383
VarExp	
The <code>VarExp</code> abstraction defines expressions referencing pools	384

MM::VarExp	
Defines the VarExp class	384
MM::Vector< T >	387
MM::Violation	393
Violation	
The Violation abstraction is a transition event that expresses an Assertion has been violated	396
yy_buffer_state	396
yy_trans_info	397
yyalloc	398
YYLTYPE	398
YYSTYPE	399

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Activation.cpp	401
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Activation.h	402
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/ActiveExp.cpp	402
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/ActiveExp.h	402
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/AliasExp.cpp	403
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/AliasExp.h	404
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/AllExp.cpp	405
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/AllExp.h	405
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Assertion.cpp	406
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Assertion.h	406
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/BinExp.cpp	407
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/BinExp.h	407
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/BooleanValExp.cpp	408
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/BooleanValExp.h	409
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/ConverterNodeBehavior.cpp	411
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/ConverterNodeBehavior.h	412
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Declaration.cpp	414
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Declaration.h	415
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Definition.cpp	415
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Definition.h	416
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Deletion.cpp	416
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Deletion.h	417
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/DieExp.cpp	417
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/DieExp.h	418
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Disablement.cpp	419
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Disablement.h	419
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/DrainNodeBehavior.cpp	421
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/DrainNodeBehavior.h	422
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Edge.cpp	422
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Edge.h	423
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Element.cpp	423
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Element.h	424
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Enablement.cpp	425
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Enablement.h	425
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Evaluator.cpp	427
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Evaluator.h	428
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Event.cpp	428
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Event.h	429

C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Exp.cpp	429
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Exp.h	430
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Failure.cpp	431
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Failure.h	431
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/FlowEdge.cpp	432
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/FlowEdge.h	432
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/FlowEvent.cpp	433
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/FlowEvent.h	433
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/GateNodeBehavior.cpp	434
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/GateNodeBehavior.h	434
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Instance.cpp	436
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Instance.h	437
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/InterfaceNode.cpp	438
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/InterfaceNode.h	438
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/lex.mm.cpp	439
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Location.cpp	456
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Location.h	457
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Machine.cpp	459
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Machine.h	464
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Map.h	464
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Message.cs	465
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/mm.h	467
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/mm.tab.cpp	468
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/mm.tab.hpp	485
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Modification.cpp	490
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Modification.h	491
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Name.cpp	491
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Name.h	492
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Node.cpp	494
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Node.h	495
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/NodeBehavior.cpp	496
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/NodeBehavior.h	497
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/NodeWorkItem.cpp	497
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/NodeWorkItem.h	498
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/NumberValExp.cpp	498
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/NumberValExp.h	499
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Observable.cpp	499
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Observable.h	500
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Observer.cpp	500
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Observer.h	501
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/OneExp.cpp	502
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/OneExp.h	502
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Operator.cpp	503
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Operator.h	503
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/OverrideExp.cpp	504
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/OverrideExp.h	505
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/PoolNodeBehavior.cpp	506
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/PoolNodeBehavior.h	507
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Prevention.cpp	508
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Prevention.h	508
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Program.cpp	509
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Program.h	509
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/RangeValExp.cpp	510
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/RangeValExp.h	510
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Recyclable.cpp	511
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Recyclable.h	511
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Recycler.cpp	512
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Recycler.h	512

C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Reflector.cpp	514
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Reflector.h	515
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/RefNodeBehavior.cpp	516
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/RefNodeBehavior.h	516
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/SourceNodeBehavior.cpp	518
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/SourceNodeBehavior.h	519
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/StateEdge.cpp	519
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/StateEdge.h	520
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/String.cpp	520
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/String.h	521
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Transformation.cpp	522
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Transformation.h	522
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Transition.cpp	523
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Transition.h	523
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/TriggerEvent.cpp	524
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/TriggerEvent.h	524
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/TriggerExp.cpp	525
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/TriggerExp.h	526
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Types.h	527
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/UnExp.cpp	528
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/UnExp.h	529
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/ValExp.cpp	530
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/ValExp.h	530
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/VarExp.cpp	531
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/VarExp.h	532
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Vector.h	532
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Violation.cpp	533
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Violation.h	534
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/YYLTYPE.h	534

Chapter 5

Namespace Documentation

5.1 Package LibMM

Enumerations

- enum MESSAGE {
MESSAGE.MSG_ERROR, MESSAGE.MSG_NEW_TYPE, MESSAGE.MSG_NEW_DECL, MESSAGE.MSG_NEW_POOL,
MESSAGE.MSG_NEW_SOURCE, MESSAGE.MSG_NEW_DRAIN, MESSAGE.MSG_NEW_GATE, MESSAGE.MSG_NEW_REF,
MESSAGE.MSG_NEW_CONVERTER, MESSAGE.MSG_NEW_CONDITION, MESSAGE.MSG_NEW_TRIGGER,
MESSAGE.MSG_NEW_FLOW,
MESSAGE.MSG_DEL_TYPE, MESSAGE.MSG_DEL_DECL, MESSAGE.MSG_DEL_POOL, MESSAGE.MSG_DEL_SOURCE,
MESSAGE.MSG_DEL_DRAIN, MESSAGE.MSG_DEL_GATE, MESSAGE.MSG_DEL_REF, MESSAGE.MSG_DEL_CONVERTER,
MESSAGE.MSG_DEL_CONDITION, MESSAGE.MSG_DEL_TRIGGER, MESSAGE.MSG_DEL_FLOW, MESSAGE.MSG_UPD_TYPE,
MESSAGE.MSG_UPD_DECL, MESSAGE.MSG_UPD_POOL, MESSAGE.MSG_UPD_SOURCE, MESSAGE.MSG_UPD_DRAIN,
MESSAGE.MSG_UPD_GATE, MESSAGE.MSG_UPD_REF, MESSAGE.MSG_UPD_CONVERTER, MESSAGE.MSG_UPD_CONDITION,
MESSAGE.MSG_UPD_TRIGGER, MESSAGE.MSG_UPD_FLOW, MESSAGE.MSG_NEW_INST, MESSAGE.MSG_DEL_INST,
MESSAGE.MSG_ADD_VALUE, MESSAGE.MSG_SUB_VALUE, MESSAGE.MSG_HAS_VALUE, MESSAGE.MSG_TRIGGER,
MESSAGE.MSG_ACTIVATE, MESSAGE.MSG_ENABLE, MESSAGE.MSG_DISABLE, MESSAGE.MSG_VIOLATE,
MESSAGE.MSG_FAIL, MESSAGE.MSG_PREVENT }

MESSAGE enumeration defines messages sent to observers.

5.1.1 Enumeration Type Documentation

5.1.1.1 enum LibMM.MESSAGE

MESSAGE enumeration defines messages sent to observers.

Note

Using define hackery for single file maintenance this file is C++
Using define hackery for single file maintenance this file is C#

Enumerator

MSG_ERROR
MSG_NEW_TYPE
MSG_NEW_DECL
MSG_NEW_POOL
MSG_NEW_SOURCE
MSG_NEW_DRAIN
MSG_NEW_GATE
MSG_NEW_REF
MSG_NEW_CONVERTER
MSG_NEW_CONDITION
MSG_NEW_TRIGGER
MSG_NEW_FLOW
MSG_DEL_TYPE
MSG_DEL_DECL
MSG_DEL_POOL
MSG_DEL_SOURCE
MSG_DEL_DRAIN
MSG_DEL_GATE
MSG_DEL_REF
MSG_DEL_CONVERTER
MSG_DEL_CONDITION
MSG_DEL_TRIGGER
MSG_DEL_FLOW
MSG_UPD_TYPE
MSG_UPD_DECL
MSG_UPD_POOL
MSG_UPD_SOURCE
MSG_UPD_DRAIN
MSG_UPD_GATE
MSG_UPD_REF
MSG_UPD_CONVERTER
MSG_UPD_CONDITION
MSG_UPD_TRIGGER
MSG_UPD_FLOW
MSG_NEW_INST
MSG_DEL_INST
MSG_ADD_VALUE
MSG_SUB_VALUE
MSG_HAS_VALUE
MSG_TRIGGER
MSG_ACTIVATE
MSG_ENABLE
MSG_DISABLE
MSG_VIOLATE
MSG_FAIL
MSG_PREVENT

5.2 MM Namespace Reference

Classes

- class [Activation](#)
- class [ActiveExp](#)
- class [AliasExp](#)
- class [AllExp](#)
- class [Assertion](#)
- class [BinExp](#)
- class [BooleanValExp](#)
- class [ConverterNodeBehavior](#)
- class [Declaration](#)
- class [Definition](#)
- class [Deletion](#)
- class [DieExp](#)

Defines the [DieExp](#) class.

- class [Disablement](#)
- class [DrainNodeBehavior](#)
- class [Edge](#)
- class [Element](#)
- class [Enablement](#)
- class [Evaluator](#)
- class [Event](#)
- class [Exp](#)
- class [Failure](#)
- class [FlowEdge](#)
- class [FlowEvent](#)
- class [GateNodeBehavior](#)
- class [Instance](#)
- class [InterfaceNode](#)
- class [Location](#)
- class [Machine](#)
- class [Map](#)
- class [Modification](#)
- class [Name](#)
- class [Node](#)
- class [NodeBehavior](#)
- class [NodeWorkItem](#)
- class [NumberValExp](#)
- class [Observable](#)
- class [Observer](#)
- class [OneExp](#)
- class [Operator](#)
- class [OverrideExp](#)
- class [PoolNodeBehavior](#)
- class [Prevention](#)
- class [Program](#)
- class [RangeValExp](#)
- class [Recyclable](#)
- class [Recycler](#)
- class [Reflector](#)
- class [RefNodeBehavior](#)
- class [SourceNodeBehavior](#)

- class [StateEdge](#)
- class [String](#)
- class [Transformation](#)
- class [Transition](#)
- class [TriggerEvent](#)
- class [TriggerExp](#)
- class [UnExp](#)

Defines the [UnExp](#) class.
- class [ValExp](#)

Defines the [ValExp](#) class.
- class [VarExp](#)

Defines the [VarExp](#) class.
- class [Vector](#)
- class [Violation](#)

Typedefs

- typedef unsigned char [UINT8](#)
- typedef long [INT32](#)
- typedef unsigned long [UINT32](#)
- typedef char [CHAR](#)
- typedef void [VOID](#)
- typedef bool [BOOLEAN](#)
- typedef MM::VOID(* [CALLBACK](#))(MM::UINT32 caller, MM::UINT32 message, MM::UINT32 instance, MM::UINT32 element, MM::UINT32 val)
- typedef enum [MM::TID](#) TID

Enumerations

- enum [TID](#) {
 [T_NULL](#), [T_Recycler](#), [T_Recyclable](#), [T_Observer](#),
[T_Observable](#), [T_String](#), [T_Map](#), [T_Vector](#),
[T_Reflector](#), [T_Machine](#), [T_Delegate](#), [T_InstanceObserver](#),
[T_Program](#), [T_Evaluator](#), [T_Instance](#), [T_Transformation](#),
[T_Transition](#), [T_Modification](#), [T_Event](#), [T_FlowEvent](#),
[T_TriggerEvent](#), [T_Failure](#), [T_Activation](#), [T_Violation](#),
[T_Prevention](#), [T_Enablement](#), [T_Disablement](#), [T_Location](#),
[T_Name](#), [T_Element](#), [T_Definition](#), [T_Declaration](#),
[T_Assertion](#), [T_Deletion](#), [T_Edge](#), [T_FlowEdge](#),
[T_StateEdge](#), [T_Node](#), [T_InterfaceNode](#), [T_NodeBehavior](#),
[T_PoolNodeBehavior](#), [T_DrainNodeBehavior](#), [T_GateNodeBehavior](#), [T_SourceNodeBehavior](#),
[T_RefNodeBehavior](#), [T_ConverterNodeBehavior](#), [T_Exp](#), [T_TriggerExp](#),
[T_AliasExp](#), [T_OneExp](#), [T_ActiveExp](#), [T_DieExp](#),
[T_OverrideExp](#), [T_VarExp](#), [T_AllExp](#), [T_BinExp](#),
[T_UnExp](#), [T_ValExp](#), [T_BooleanValExp](#), [T_NumberValExp](#),
[T_RangeValExp](#) }

5.2.1 Detailed Description

The [ActiveExp](#) abstraction expresses that a target of a state edge with an [ActiveExp](#) is only active if the source is active.

The [AliasExp](#) abstraction expresses that a source of a state edge with an [AliasExp](#) is an alias of the target of that edge.

The [AllExp](#) abstraction expresses a that the full available amount will flow along a [FlowEdge](#) from a source node to target node.

Note

A - all -> B equals A - A -> B

The [BinExp](#) abstraction defines binary expressions.

The [BooleanValExp](#) abstraction defines boolean value expressions.

The [DieExp](#) abstraction expresses that a target of a state edge with an [DieExp](#) is only active if the source is active.

The [Element](#) abstraction is the abstract superclass of all program elements.

The [Event](#) abstraction is the abstract superclass of all transition elements.

The [Exp](#) abstraction is the abstract super class of all expessions.

The [Instance](#) abstraction defines instances of type definitions.

The [Location](#) abstraction defines textual source locations.

The [Name](#) abstraction is used to compose identifiers.

The [NumberValExp](#) abstraction defines number value expressions.

The [OneExp](#) abstraction expresses shorthand on edges \rightarrow : [OneExp](#) has a flow of one: [NumberValueExp\(1\)](#) . * \rightarrow : [BinExp](#) has lhs [OneExp](#) and rhs [OneExp](#): [isTrigger\(\)](#) = MM_TRUE . == \rightarrow : [BinExp](#) has lhs [OneExp](#) and rhs [OneExp](#): lhs [OneExp](#) refers to source and rhs [OneExp](#) refers to target

The [Operator](#) abstraction defines operator lexicals and codes.

The [OverrideExp](#) abstraction expresses an overriden expression.

The [RangeValExp](#) abstraction defines range value expressions.

The [TriggerExp](#) abstraction expresses that a source of a state edge activates the target of that edge if all of the edge the source operates on are satisfied (meaning a flow of one or more exists for that edge).

Note

The notion of satisfied does not respect all or any modifiers of nodes.

The [UnExp](#) abstraction defines unary expressions.

The [ValExp](#) abstraction defines value expressions.

The [VarExp](#) abstraction defines expressions that reference a pool.

/namespace MM /class Vector /brief The [Vector](#) abstraction stores elements by index. /note Currently wraps STL, we intend to remove this dependency. /note The [Vector](#) class hides internal details of the STL implementation of std::map /file [Vector.h](#) /author Riemer van Rozen /date September 18th 2013

5.2.2 Typedef Documentation

5.2.2.1 [typedef bool MM::BOOLEAN](#)

void type

5.2.2.2 [typedef MM::VOID\(* MM::CALLBACK\)\(MM::UINT32 caller,MM::UINT32 message,MM::UINT32 instance,MM::UINT32 element,MM::UINT32 val\)](#)

5.2.2.3 [typedef char MM::CHAR](#)

unsigned 32 bits integer type

5.2.2.4 `typedef long MM::INT32`

unsigned byte type

5.2.2.5 `typedef enum MM::TID MM::TID`**5.2.2.6 `typedef unsigned long MM::UINT32`**

signed 32 bits integer type

5.2.2.7 `typedef unsigned char MM::UINT8`**5.2.2.8 `typedef void MM::VOID`**

character type

5.2.3 Enumeration Type Documentation**5.2.3.1 `enum MM::TID`**

Enumerator

T_NULL

T_Recycler

T_Recyclable

T_Observer

T_Observable

T_String

T_Map

T_Vector

T_Reflector

T_Machine

T_Delegate

T_InstanceObserver

T_Program

T_Evaluator

T_Instance

T_Transformation

T_Transition

T_Modification

T_Event

T_FlowEvent

T_TriggerEvent

T_Failure

T_Activation

T_Violation

T_Prevention

T_Enablement
T_Disablement
T_Location
T_Name
T_Element
T_Definition
T_Declaration
T_Assertion
T_Deletion
T_Edge
T_FlowEdge
T_StateEdge
T_Node
T_InterfaceNode
T_NodeBehavior
T_PoolNodeBehavior
T_DrainNodeBehavior
T_GateNodeBehavior
T_SourceNodeBehavior
T_RefNodeBehavior
T_ConverterNodeBehavior
T_Exp
T_TriggerExp
T_AliasExp
T_OneExp
T_ActiveExp
T_DieExp
T_OverrideExp
T_VarExp
T_AllExp
T_BinExp
T_UnExp
T_ValExp
T_BooleanValExp
T_NumberValExp
T_RangeValExp

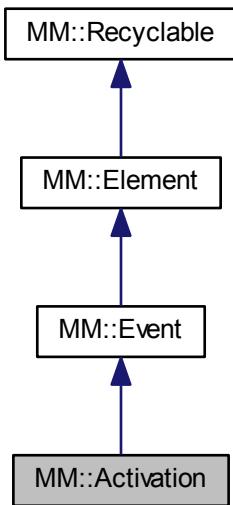
Chapter 6

Class Documentation

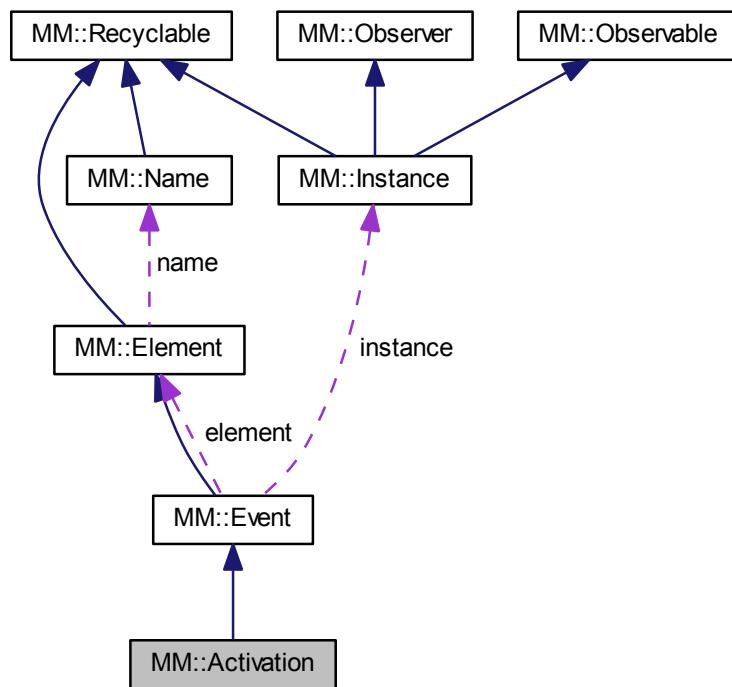
6.1 MM::Activation Class Reference

```
#include <Activation.h>
```

Inheritance diagram for MM::Activation:



Collaboration diagram for MM::Activation:



Public Member Functions

- `Activation (MM::Name *name)`
- `Activation (MM::Location *loc, MM::Name *name)`
- `Activation (MM::Instance *instance, MM::Node *node)`
- `~Activation ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::Location * getLocation ()`
- `MM::MESSAGE getMessage ()`
- `MM::VOID toString (MM::String *buf)`
- `MM::VOID toString (MM::String *buf, MM::UINT32 indent)`

Additional Inherited Members

6.1.1 Constructor & Destructor Documentation

6.1.1.1 Activation::Activation (`MM::Name * name`)

activation source location

6.1.1.2 Activation::Activation (**MM::Location** * *loc*, **MM::Name** * *name*)

6.1.1.3 Activation::Activation (**MM::Instance** * *instance*, **MM::Node** * *node*)

6.1.1.4 Activation::~Activation ()

6.1.2 Member Function Documentation

6.1.2.1 **MM::Location** * Activation::getLocation ()

6.1.2.2 **MM::MESSAGE** Activation::getMessage () [virtual]

Implements [MM::Event](#).

6.1.2.3 **MM::TID** Activation::getTypeId () [virtual]

Reimplemented from [MM::Event](#).

6.1.2.4 **MM::BOOLEAN** Activation::instanceof (**MM::TID** *tid*) [virtual]

Reimplemented from [MM::Event](#).

Here is the call graph for this function:



6.1.2.5 **MM::VOID** Activation::recycle (**MM::Recycler** * *r*) [virtual]

Reimplemented from [MM::Recyclable](#).

Here is the call graph for this function:



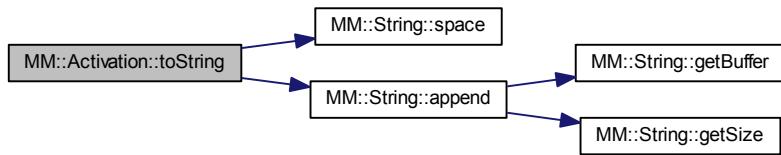
6.1.2.6 **MM::VOID** Activation::toString (**MM::String** * *buf*) [virtual]

Implements [MM::Event](#).

6.1.2.7 **MM::VOID** Activation::toString (**MM::String** * *buf*, **MM::UINT32** *indent*) [virtual]

Implements [MM::Event](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Activation.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Activation.cpp](#)

6.2 Activation Class Reference

The [Activation](#) abstraction is a transition event that expresses a node instance is activated.

```
#include <Activation.h>
```

6.2.1 Detailed Description

The [Activation](#) abstraction is a transition event that expresses a node instance is activated.

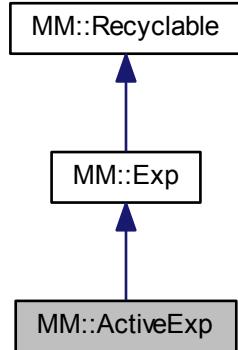
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Activation.h](#)

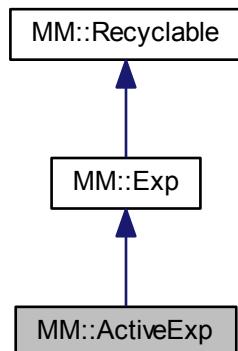
6.3 MM::ActiveExp Class Reference

```
#include <ActiveExp.h>
```

Inheritance diagram for MM::ActiveExp:



Collaboration diagram for MM::ActiveExp:



Public Member Functions

- `ActiveExp (MM::Name *name)`
- `ActiveExp (MM::Name *name, MM::Location *loc)`
- `~ActiveExp ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`

Retrieves the type id of a `ActiveExp` object.

- `MM::BOOLEAN instanceof (MM::TID tid)`

Assesses if an object is an instance of a type tid.

- `MM::Name * getName ()`

Retrieves the name of an `ActiveExp` object.

- MM::VOID **toString** (MM::String *buf)
Serializes an ActiveExp object into a String buffer.

Additional Inherited Members

6.3.1 Constructor & Destructor Documentation

6.3.1.1 ActiveExp::ActiveExp (MM::Name * name)

source location

Constructs a ActiveExp object.

Parameters

<i>name</i>	name
-------------	------

Returns

new ActiveExp object

6.3.1.2 ActiveExp::ActiveExp (MM::Name * name, MM::Location * loc)

Constructs a ActiveExp object.

Parameters

<i>name</i>	name
<i>loc</i>	source location

Returns

new ActiveExp object

6.3.1.3 ActiveExp::~ActiveExp ()

Destructs an ActiveExp object.

6.3.2 Member Function Documentation

6.3.2.1 MM::Name * ActiveExp::getName ()

Retrieves the name of an ActiveExp object.

Returns

name

6.3.2.2 MM::TID ActiveExp::getTypeId () [virtual]

Retrieves the type id of a ActiveExp object.

Returns

type id

Reimplemented from MM::Exp.

6.3.2.3 MM::BOOLEAN ActiveExp::instanceof (MM::TID *tid*) [virtual]

Assesses if an object is an instance of a type tid.

Parameters

<i>tid</i>	type id
------------	---------

Returns

MM_TRUE if this object is instance of tid, MM_FALSE otherwise

Reimplemented from [MM::Exp](#).

Here is the call graph for this function:



6.3.2.4 MM::VOID ActiveExp::recycle (MM::Recycler * *r*) [virtual]

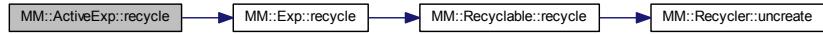
Recycles an [ActiveExp](#) object in a [Recycler](#).

Parameters

<i>r</i>	Recycler
----------	--------------------------

Implements [MM::Exp](#).

Here is the call graph for this function:



6.3.2.5 MM::VOID ActiveExp::toString (MM::String * *buf*) [virtual]

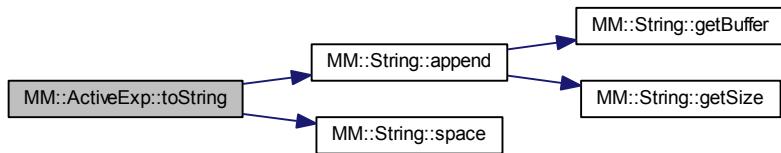
Serializes an [ActiveExp](#) object into a [String](#) buffer.

Parameters

<i>buf</i>	String buffer to serialize this object into
------------	---

Implements [MM::Exp](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[ActiveExp.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[ActiveExp.cpp](#)

6.4 ActiveExp Class Reference

The [ActiveExp](#) abstraction is a program element which expresses that a target node instance of a state edge with an [ActiveExp](#) is only active if the source node instance is active.

```
#include <ActiveExp.h>
```

6.4.1 Detailed Description

The [ActiveExp](#) abstraction is a program element which expresses that a target node instance of a state edge with an [ActiveExp](#) is only active if the source node instance is active.

The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[ActiveExp.h](#)

6.5 AliasExp Class Reference

The [AliasExp](#) abstraction expresses that a source of a state edge that has an [AliasExp](#) is an alias of the target of that edge.

```
#include <AliasExp.h>
```

6.5.1 Detailed Description

The [AliasExp](#) abstraction expresses that a source of a state edge that has an [AliasExp](#) is an alias of the target of that edge.

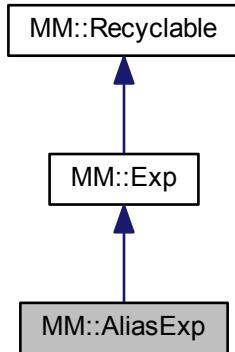
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[AliasExp.h](#)

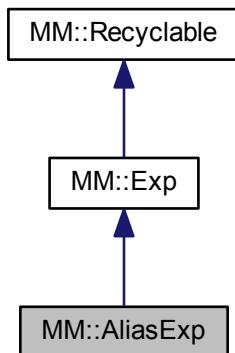
6.6 MM::AliasExp Class Reference

```
#include <AliasExp.h>
```

Inheritance diagram for MM::AliasExp:



Collaboration diagram for MM::AliasExp:



Public Member Functions

- [AliasExp \(\)](#)
- [AliasExp \(MM::Location *loc\)](#)
- [~AliasExp \(\)](#)
- [MM::VOID recycle \(MM::Recycler *r\)](#)
- [MM::TID getTypeId \(\)](#)

Retrieves the type id of a [AliasExp](#) object.
- [MM::BOOLEAN instanceof \(MM::TID tid\)](#)

Assesses if an object is an instance of a type tid.
- [MM::VOID toString \(MM::String *buf\)](#)

Serializes an [AliasExp](#) object into a [String](#) buffer.

Additional Inherited Members

6.6.1 Constructor & Destructor Documentation

6.6.1.1 AliasExp::AliasExp ()

source location

Constructs an [AliasExp](#) object.

Returns

new [AliasExp](#) object

Constructs an [AliasExp](#) object.

Parameters

<i>loc</i>	source location
------------	-----------------

Returns

new [AliasExp](#) object

6.6.1.2 AliasExp::AliasExp (MM::Location * *loc*)

Destructs an [AliasExp](#) object.

6.6.1.3 AliasExp::~AliasExp ()

6.6.2 Member Function Documentation

6.6.2.1 MM::TID AliasExp::getTypeId () [virtual]

Retrieves the type id of a [AliasExp](#) object.

Returns

type id

Reimplemented from [MM::Exp](#).

6.6.2.2 MM::BOOLEAN AliasExp::instanceof (MM::TID *tid*) [virtual]

Assesses if an object is an instance of a type tid.

Parameters

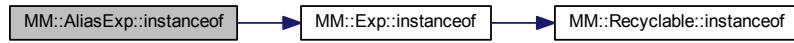
<i>tid</i>	type id
------------	---------

Returns

MM_TRUE if this object is instance of tid, MM_FALSE otherwise

Reimplemented from [MM::Exp](#).

Here is the call graph for this function:

**6.6.2.3 MM::VOID AliasExp::recycle (MM::Recycler * r) [virtual]**

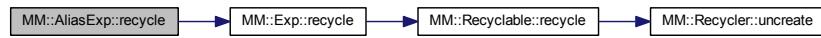
Recycles an [AliasExp](#) object in a [Recycler](#).

Parameters

<code>r</code>	Recycler
----------------	--------------------------

Implements [MM::Exp](#).

Here is the call graph for this function:

**6.6.2.4 MM::VOID AliasExp::toString (MM::String * buf) [virtual]**

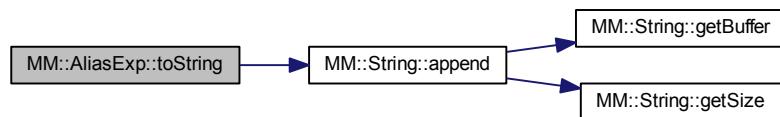
Serializes an [AliasExp](#) object into a [String](#) buffer.

Parameters

<code>buf</code>	String buffer to serialize this object into
------------------	---

Implements [MM::Exp](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

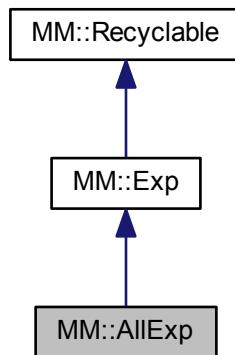
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[AliasExp.h](#)

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[AliasExp.cpp](#)

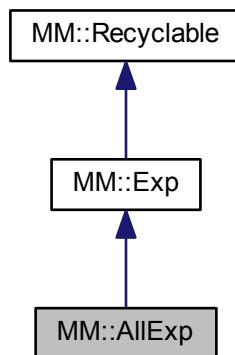
6.7 MM::AllExp Class Reference

```
#include <AllExp.h>
```

Inheritance diagram for MM::AllExp:



Collaboration diagram for MM::AllExp:



Public Member Functions

- [AllExp \(\)](#)
- [AllExp \(MM::Location *loc\)](#)
- [~AllExp \(\)](#)
- [MM::VOID recycle \(MM::Recycler *r\)](#)

- **MM::TID getTypeId ()**
Retrieves the type id of a AllExp object.
- **MM::BOOLEAN instanceof (MM::TID tid)**
Assesses if an object is an instance of a type tid.
- **MM::VOID toString (MM::String *buf)**
Serializes an AllExp object into a String buffer.

Additional Inherited Members

6.7.1 Constructor & Destructor Documentation

6.7.1.1 AllExp::AllExp ()

source location

Constructs an AllExp object.

Returns

new AllExp object

Constructs an AllExp object.

Parameters

source	location
--------	----------

Returns

new AllExp object

6.7.1.2 AllExp::AllExp (MM::Location * loc)

6.7.1.3 AllExp::~AllExp ()

Deconstructs a VarExp object.

6.7.2 Member Function Documentation

6.7.2.1 MM::TID AllExp::getTypeId () [virtual]

Retrieves the type id of a AllExp object.

Returns

type id

Reimplemented from [MM::Exp](#).

6.7.2.2 MM::BOOLEAN AllExp::instanceof (MM::TID tid) [virtual]

Assesses if an object is an instance of a type tid.

Parameters

<i>tid</i>	type id
------------	---------

Returns

MM_TRUE if this object is instance of tid, MM_FALSE otherwise

Reimplemented from [MM::Exp](#).

Here is the call graph for this function:

**6.7.2.3 MM::VOID AllExp::recycle(MM::Recycler * r) [virtual]**

Recycles an [AllExp](#) object in a [Recycler](#).

Parameters

<i>r</i>	Recycler
----------	----------

Implements [MM::Exp](#).

Here is the call graph for this function:

**6.7.2.4 MM::VOID AllExp::toString(MM::String * buf) [virtual]**

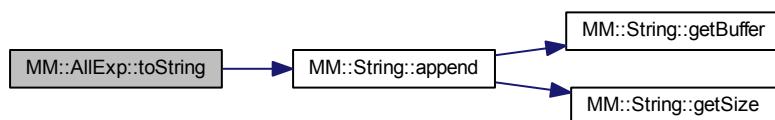
Serializes an [AllExp](#) object into a [String](#) buffer.

Parameters

<i>buf</i>	String buffer to serialize this object into
------------	---

Implements [MM::Exp](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[AllExp.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[AllExp.cpp](#)

6.8 AllExp Class Reference

The [AllExp](#) abstraction expresses a that the full available amount of resources will flow along a [FlowEdge](#) from a source node instance to a target node instance.

```
#include <AllExp.h>
```

6.8.1 Detailed Description

The [AllExp](#) abstraction expresses a that the full available amount of resources will flow along a [FlowEdge](#) from a source node instance to a target node instance.

Note

A - all -> B equals A - A -> B

The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[AllExp.h](#)

6.9 Assertion Class Reference

The [Assertion](#) abstraction is a program element that generates Signals when its expression is false in some instance.

```
#include <Assertion.h>
```

6.9.1 Detailed Description

The [Assertion](#) abstraction is a program element that generates Signals when its expression is false in some instance.

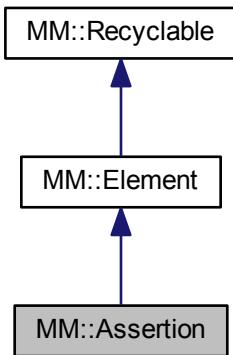
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Assertion.h](#)

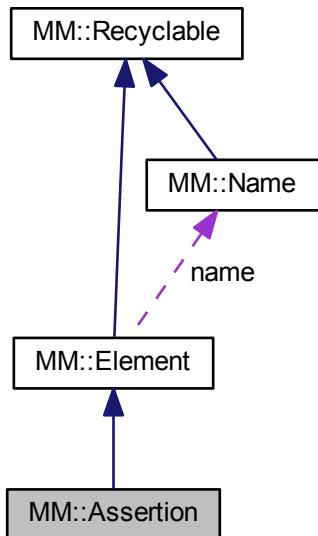
6.10 MM::Assertion Class Reference

```
#include <Assertion.h>
```

Inheritance diagram for MM::Assertion:



Collaboration diagram for MM::Assertion:



Public Member Functions

- `Assertion (MM::Name *name, MM::Exp *exp, MM::CHAR *msg, MM::Location *loc)`
- `Assertion (MM::Name *name, MM::Exp *exp, MM::CHAR *msg)`
- `~Assertion ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`

- `MM::Exp * getExp ()`
- `MM::CHAR * getMessage ()`
- `MM::UINT32 getMessageLength ()`
- `MM::Location * getLocation ()`
- `MM::VOID toString (MM::String *buf)`
- `MM::VOID toString (MM::String *buf, MM::UINT32 indent)`

Additional Inherited Members

6.10.1 Constructor & Destructor Documentation

6.10.1.1 `Assertion::Assertion (MM::Name * name, MM::Exp * exp, MM::CHAR * msg, MM::Location * loc)`

assert source location

6.10.1.2 `Assertion::Assertion (MM::Name * name, MM::Exp * exp, MM::CHAR * msg)`

6.10.1.3 `Assertion::~Assertion ()`

6.10.2 Member Function Documentation

6.10.2.1 `MM::Exp * Assertion::getExp ()`

6.10.2.2 `MM::Location * Assertion::getLocation ()`

6.10.2.3 `MM::CHAR * Assertion::getMessage ()`

6.10.2.4 `MM::UINT32 Assertion::getMessageLength ()`

6.10.2.5 `MM::TID Assertion::getTypeld () [virtual]`

Reimplemented from [MM::Element](#).

6.10.2.6 `MM::BOOLEAN Assertion::instanceof (MM::TID tid) [virtual]`

Reimplemented from [MM::Element](#).

Here is the call graph for this function:



6.10.2.7 `MM::VOID Assertion::recycle (MM::Recycler * r) [virtual]`

Reimplemented from [MM::Recyclable](#).

Here is the call graph for this function:



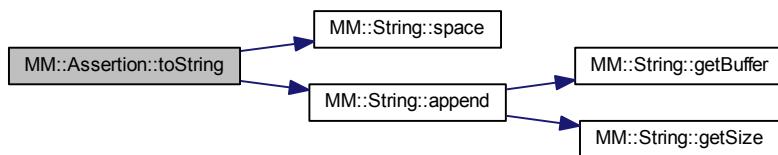
6.10.2.8 MM::VOID Assertion::toString (MM::String * buf) [virtual]

Implements [MM::Element](#).

6.10.2.9 MM::VOID Assertion::toString (MM::String * buf, MM::UINT32 indent) [virtual]

Implements [MM::Element](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Assertion.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Assertion.cpp](#)

6.11 BinExp Class Reference

The [BinExp](#) abstraction defines binary expressions.

```
#include <BinExp.h>
```

6.11.1 Detailed Description

The [BinExp](#) abstraction defines binary expressions.

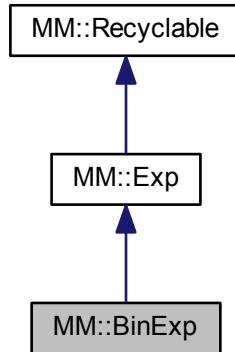
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[BinExp.h](#)

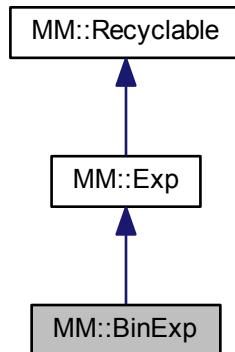
6.12 MM::BinExp Class Reference

```
#include <BinExp.h>
```

Inheritance diagram for MM::BinExp:



Collaboration diagram for MM::BinExp:



Public Member Functions

- `BinExp (MM::Exp *e1, MM::Operator::OP op, MM::Exp *e2)`
- `BinExp (MM::Exp *e1, MM::Operator::OP op, MM::Exp *e2, MM::Location *loc)`
- `~BinExp ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`
Retrieves the type id of a `Exp` object.
- `MM::BOOLEAN instanceof (MM::TID tid)`
Assesses if an object is an instance of a type tid.
 - `MM::Exp * getLhsExp ()`
 - `MM::Exp * getRhsExp ()`
 - `MM::Operator::OP getOperator ()`

- MM::BOOLEAN isTriggerExp ()
- MM::BOOLEAN isAlaiaExp ()
- MM::VOID toString (MM::String *buf)

Additional Inherited Members

6.12.1 Constructor & Destructor Documentation

6.12.1.1 BinExp::BinExp (MM::Exp * e1, MM::Operator::OP op, MM::Exp * e2)

operator source location

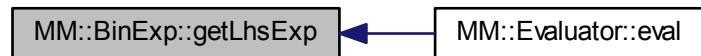
6.12.1.2 BinExp::BinExp (MM::Exp * e1, MM::Operator::OP op, MM::Exp * e2, MM::Location * loc)

6.12.1.3 BinExp::~BinExp ()

6.12.2 Member Function Documentation

6.12.2.1 MM::Exp * BinExp::getLhsExp ()

Here is the caller graph for this function:



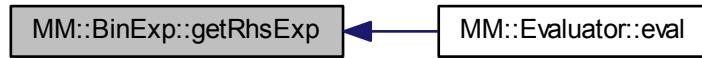
6.12.2.2 MM::Operator::OP BinExp::getOperator ()

Here is the caller graph for this function:



6.12.2.3 **MM::Exp * BinExp::getRhsExp()**

Here is the caller graph for this function:

6.12.2.4 **MM::TID BinExp::getTypeld() [virtual]**

Retrieves the type id of a [Exp](#) object.

Retrieves the type id of a [TriggerExp](#) object.

Returns

type id

Reimplemented from [MM::Exp](#).

6.12.2.5 **MM::BOOLEAN BinExp::instanceof(MM::TID tid) [virtual]**

Assesses if an object is an instance of a type tid.

Parameters

<i>tid</i>	type id
------------	---------

Returns

MM_TRUE if this object is instance of tid, MM_FALSE otherwise

Reimplemented from [MM::Exp](#).

Here is the call graph for this function:

6.12.2.6 **MM::BOOLEAN MM::BinExp::isAliaExp()**

6.12.2.7 MM::BOOLEAN BinExp::isTriggerExp ()

Here is the caller graph for this function:



6.12.2.8 MM::VOID BinExp::recycle (MM::Recycler * r) [virtual]

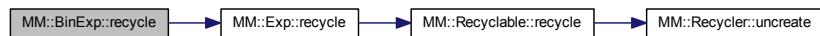
Recycles an [Exp](#) object in a [Recycler](#).

Parameters

<code>r</code>	Recycler object
----------------	---------------------------------

Implements [MM::Exp](#).

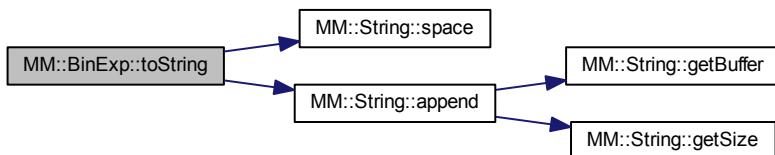
Here is the call graph for this function:



6.12.2.9 MM::VOID BinExp::toString (MM::String * buf) [virtual]

Implements [MM::Exp](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[BinExp.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[BinExp.cpp](#)

6.13 BooleanValExp Class Reference

The [BooleanValExp](#) abstraction defines boolean value expressions.

```
#include <BooleanValExp.h>
```

6.13.1 Detailed Description

The [BooleanValExp](#) abstraction defines boolean value expressions.

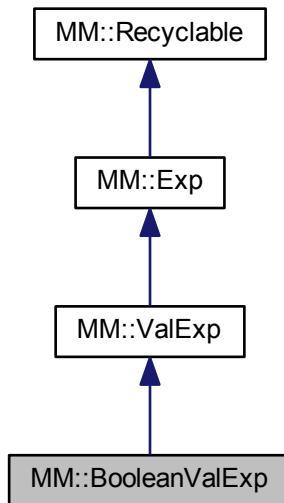
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[BooleanValExp.h](#)

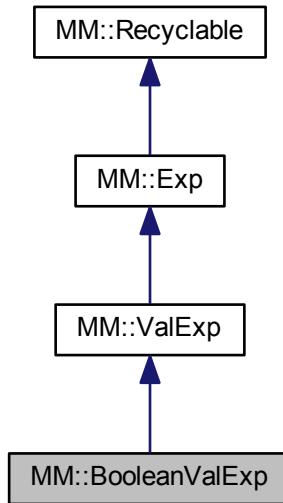
6.14 MM::BooleanValExp Class Reference

```
#include <BooleanValExp.h>
```

Inheritance diagram for MM::BooleanValExp:



Collaboration diagram for MM::BooleanValExp:



Public Member Functions

- `BooleanValExp (MM::BOOLEAN val)`
- `BooleanValExp (MM::BOOLEAN val, MM::Location *loc)`
- `~BooleanValExp ()`
- `MM::TID getTypeId ()`

Retrieves the type id of a `Exp` object.

- `MM::BOOLEAN instanceof (MM::TID tid)`

Assesses if an object is an instance of a type tid.

- `MM::VOID recycle (MM::Recycler *r)`
- `MM::BOOLEAN getValue ()`
- `MM::Location * getLocation ()`
- `MM::BOOLEAN greaterEquals (MM::UINT32 val)`
- `MM::VOID toString (MM::String *buf)`

Additional Inherited Members

6.14.1 Constructor & Destructor Documentation

6.14.1.1 BooleanValExp::BooleanValExp (`MM::BOOLEAN val`)

source location

6.14.1.2 BooleanValExp::BooleanValExp (`MM::BOOLEAN val, MM::Location * loc`)

6.14.1.3 BooleanValExp::~BooleanValExp ()

6.14.2 Member Function Documentation

6.14.2.1 **MM::Location * BooleanValExp::getLocation ()**

6.14.2.2 **MM::TID BooleanValExp::getTypeId () [virtual]**

Retrieves the type id of a [Exp](#) object.

Retrieves the type id of a [TriggerExp](#) object.

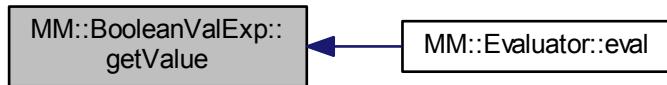
Returns

type id

Reimplemented from [MM::ValExp](#).

6.14.2.3 **MM::BOOLEAN BooleanValExp::getValue ()**

Here is the caller graph for this function:



6.14.2.4 **MM::BOOLEAN BooleanValExp::greaterEquals (MM::UINT32 val) [virtual]**

Implements [MM::ValExp](#).

6.14.2.5 **MM::BOOLEAN BooleanValExp::instanceof (MM::TID tid) [virtual]**

Assesses if an object is an instance of a type tid.

Parameters

<i>tid</i>	type id
------------	---------

Returns

MM_TRUE if this object is instance of tid, MM_FALSE otherwise

Reimplemented from [MM::ValExp](#).

Here is the call graph for this function:



6.14.2.6 MM::VOID BooleanValExp::recycle (MM::Recycler * r) [virtual]

Recycles an [Exp](#) object in a [Recycler](#).

Parameters

<i>r</i>	Recycler object
----------	---------------------------------

Implements [MM::Exp](#).

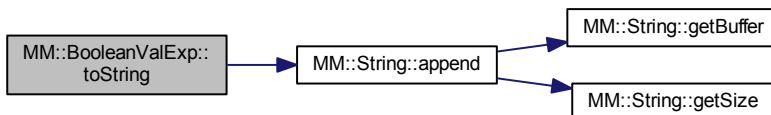
Here is the call graph for this function:



6.14.2.7 MM::VOID BooleanValExp::toString (MM::String * buf) [virtual]

Implements [MM::ValExp](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[BooleanValExp.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[BooleanValExp.cpp](#)

6.15 MM::Name::Compare Class Reference

```
#include <Name.h>
```

Public Member Functions

- bool [operator\(\)](#) ([MM::Name](#) const *n1, [MM::Name](#) const *n2) const

6.15.1 Detailed Description

/brief The [Compare](#) class defines comparison between names

6.15.2 Member Function Documentation

6.15.2.1 `bool MM::Name::Compare::operator() (MM::Name const * n1, MM::Name const * n2) const [inline]`

The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Name.h](#)

6.16 MM::Node::Compare Class Reference

```
#include <Node.h>
```

Public Member Functions

- `bool operator() (Node const *n1, Node const *n2) const`

6.16.1 Member Function Documentation

6.16.1.1 `bool MM::Node::Compare::operator() (Node const * n1, Node const * n2) const [inline]`

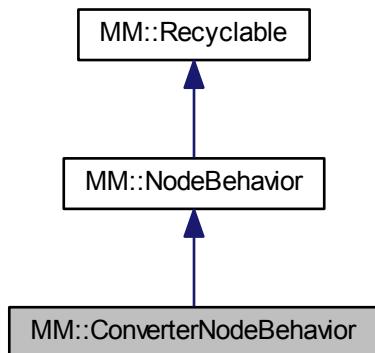
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Node.h](#)

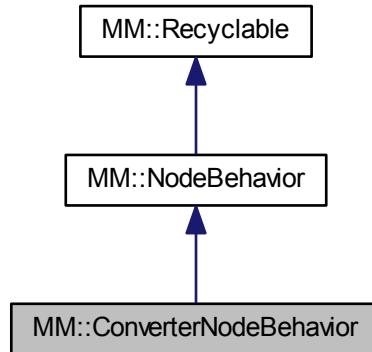
6.17 MM::ConverterNodeBehavior Class Reference

```
#include <ConverterNodeBehavior.h>
```

Inheritance diagram for MM::ConverterNodeBehavior:



Collaboration diagram for MM::ConverterNodeBehavior:



Public Member Functions

- ConverterNodeBehavior (MM::NodeBehavior::IO io, MM::NodeBehavior::When when, MM::Name *from, MM::Name *to)
- ~ConverterNodeBehavior ()
- MM::VOID recycle (MM::Recycler *r)
- MM::TID getTypeId ()
- MM::BOOLEAN instanceof (MM::TID tid)
- MM::Name * getFrom ()
- MM::Name * getTo ()
- MM::VOID setTriggerEdge (MM::Edge *triggerEdge)
- MM::VOID setSourceNode (MM::Node *sourceNode)
- MM::VOID setDrainNode (MM::Node *drainNode)
- MM::Edge * getTriggerEdge ()
- MM::Node * getSourceNode ()
- MM::Node * getDrainNode ()
- MM::VOID setFrom (MM::Name *from)
- MM::VOID setTo (MM::Name *to)
- MM::UINT32 getCreateMessage ()
- MM::UINT32 getUpdateMessage ()
- MM::UINT32 getDeleteMessage ()
- MM::VOID step (MM::Node *n, MM::Instance *i, MM::Machine *m, MM::Transition *t)
- MM::VOID stepPullAny (MM::Node *node, MM::Instance *i, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)
- MM::VOID stepPushAny (MM::Node *node, MM::Instance *i, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)
- MM::VOID stepPullAll (MM::Node *node, MM::Instance *i, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)
- MM::VOID stepPushAll (MM::Node *node, MM::Instance *i, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)
- MM::VOID begin (MM::Instance *i, MM::Machine *m, MM::Node *n)
- MM::VOID end (MM::Instance *i, MM::Machine *m, MM::Node *n)
- MM::VOID change (MM::Instance *i, MM::Machine *m, MM::Node *n)
- MM::VOID add (MM::Instance *i, MM::Machine *m, MM::Node *n, MM::UINT32 amount)

- MM::VOID sub (MM::Instance *i, MM::Machine *m, MM::Node *n, MM::UINT32 amount)
- MM::UINT32 getCapacity (MM::Instance *i, MM::Node *n)
- MM::UINT32 getResources (MM::Instance *i, MM::Node *n)
- MM::BOOLEAN hasCapacity (MM::Instance *i, MM::Node *n, MM::UINT32 amount)
- MM::BOOLEAN hasResources (MM::Instance *i, MM::Node *n, MM::UINT32 amount)
- MM::VOID activateTriggerTargets (MM::Node *node, MM::Instance *i, MM::Machine *m)
- MM::VOID toString (MM::String *buf)
- MM::VOID toString (MM::String *buf, MM::Name *name)

Additional Inherited Members

6.17.1 Constructor & Destructor Documentation

6.17.1.1 ConverterNodeBehavior::ConverterNodeBehavior (MM::NodeBehavior::IO *io*, MM::NodeBehavior::When *when*, MM::Name * *from*, MM::Name * *to*)

6.17.1.2 ConverterNodeBehavior::~ConverterNodeBehavior ()

6.17.2 Member Function Documentation

6.17.2.1 MM::VOID ConverterNodeBehavior::activateTriggerTargets (MM::Node * *node*, MM::Instance * *i*, MM::Machine * *m*) [virtual]

Reimplemented from [MM::NodeBehavior](#).

6.17.2.2 MM::VOID ConverterNodeBehavior::add (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*, MM::UINT32 *amount*) [virtual]

Implements [MM::NodeBehavior](#).

6.17.2.3 MM::VOID ConverterNodeBehavior::begin (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*) [virtual]

Implements [MM::NodeBehavior](#).

6.17.2.4 MM::VOID ConverterNodeBehavior::change (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*) [virtual]

Implements [MM::NodeBehavior](#).

6.17.2.5 MM::VOID ConverterNodeBehavior::end (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*) [virtual]

Implements [MM::NodeBehavior](#).

6.17.2.6 MM::UINT32 ConverterNodeBehavior::getCapacity (MM::Instance * *i*, MM::Node * *n*) [virtual]

Implements [MM::NodeBehavior](#).

6.17.2.7 MM::UINT32 ConverterNodeBehavior::getCreateMessage () [virtual]

Implements [MM::NodeBehavior](#).

6.17.2.8 **MM::UINT32 ConverterNodeBehavior::getDeleteMessage() [virtual]**

Implements [MM::NodeBehavior](#).

6.17.2.9 **MM::Node * ConverterNodeBehavior::getDrainNode()**

6.17.2.10 **MM::Name * ConverterNodeBehavior::getFrom()**

6.17.2.11 **MM::UINT32 ConverterNodeBehavior::getResources(MM::Instance * i, MM::Node * n) [virtual]**

Implements [MM::NodeBehavior](#).

6.17.2.12 **MM::Node * ConverterNodeBehavior::getSourceNode()**

6.17.2.13 **MM::Name * ConverterNodeBehavior::getTo()**

6.17.2.14 **MM::Edge * ConverterNodeBehavior::getTriggerEdge()**

6.17.2.15 **MM::TID ConverterNodeBehavior::getTypeld() [virtual]**

Reimplemented from [MM::NodeBehavior](#).

6.17.2.16 **MM::UINT32 ConverterNodeBehavior::getUpdateMessage() [virtual]**

Implements [MM::NodeBehavior](#).

6.17.2.17 **MM::BOOLEAN ConverterNodeBehavior::hasCapacity(MM::Instance * i, MM::Node * n, MM::UINT32 amount) [virtual]**

Implements [MM::NodeBehavior](#).

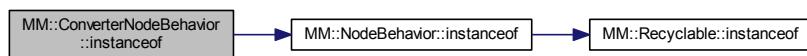
6.17.2.18 **MM::BOOLEAN ConverterNodeBehavior::hasResources(MM::Instance * i, MM::Node * n, MM::UINT32 amount) [virtual]**

Implements [MM::NodeBehavior](#).

6.17.2.19 **MM::BOOLEAN ConverterNodeBehavior::instanceof(MM::TID tid) [virtual]**

Reimplemented from [MM::NodeBehavior](#).

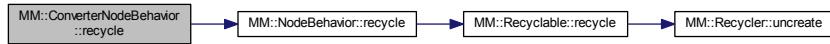
Here is the call graph for this function:



6.17.2.20 **MM::VOID ConverterNodeBehavior::recycle (MM::Recycler * r) [virtual]**

Reimplemented from [MM::Recyclable](#).

Here is the call graph for this function:



6.17.2.21 **MM::VOID ConverterNodeBehavior::setDrainNode (MM::Node * drainNode)**

6.17.2.22 **MM::VOID ConverterNodeBehavior::setFrom (MM::Name * from)**

6.17.2.23 **MM::VOID ConverterNodeBehavior::setSourceNode (MM::Node * sourceNode)**

6.17.2.24 **MM::VOID ConverterNodeBehavior::setTo (MM::Name * to)**

6.17.2.25 **MM::VOID ConverterNodeBehavior::setTriggerEdge (MM::Edge * triggerEdge)**

6.17.2.26 **MM::VOID ConverterNodeBehavior::step (MM::Node * n, MM::Instance * i, MM::Machine * m, MM::Transition * t) [virtual]**

Reimplemented from [MM::NodeBehavior](#).

6.17.2.27 **MM::VOID ConverterNodeBehavior::stepPullAll (MM::Node * node, MM::Instance * i, MM::Vector<MM::NodeWorkItem * > * work, MM::Machine * m, MM::Transition * tr) [virtual]**

Implements [MM::NodeBehavior](#).

6.17.2.28 **MM::VOID ConverterNodeBehavior::stepPullAny (MM::Node * node, MM::Instance * i, MM::Vector<MM::NodeWorkItem * > * work, MM::Machine * m, MM::Transition * tr) [virtual]**

Reimplemented from [MM::NodeBehavior](#).

6.17.2.29 **MM::VOID ConverterNodeBehavior::stepPushAll (MM::Node * node, MM::Instance * i, MM::Vector<MM::NodeWorkItem * > * work, MM::Machine * m, MM::Transition * tr) [virtual]**

Implements [MM::NodeBehavior](#).

6.17.2.30 **MM::VOID ConverterNodeBehavior::stepPushAny (MM::Node * node, MM::Instance * i, MM::Vector<MM::NodeWorkItem * > * work, MM::Machine * m, MM::Transition * tr) [virtual]**

Reimplemented from [MM::NodeBehavior](#).

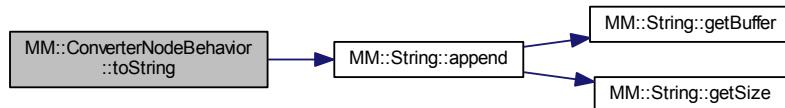
6.17.2.31 **MM::VOID ConverterNodeBehavior::sub (MM::Instance * i, MM::Machine * m, MM::Node * n, MM::UINT32 amount) [virtual]**

Implements [MM::NodeBehavior](#).

6.17.2.32 MM::VOID ConverterNodeBehavior::toString (MM::String * buf) [virtual]

Implements [MM::Recyclable](#).

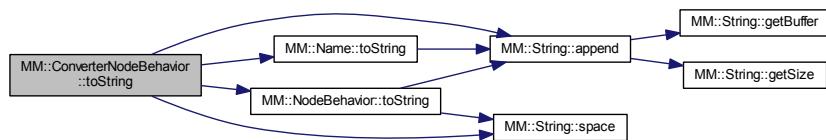
Here is the call graph for this function:



6.17.2.33 MM::VOID ConverterNodeBehavior::toString (MM::String * buf, MM::Name * name) [virtual]

Reimplemented from [MM::NodeBehavior](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[ConverterNodeBehavior.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[ConverterNodeBehavior.cpp](#)

6.18 ConverterNodeBehavior Class Reference

The [ConverterNodeBehavior](#) abstraction defines the behavior of converter nodes.

```
#include <ConverterNodeBehavior.h>
```

6.18.1 Detailed Description

The [ConverterNodeBehavior](#) abstraction defines the behavior of converter nodes.

Note

Strategy

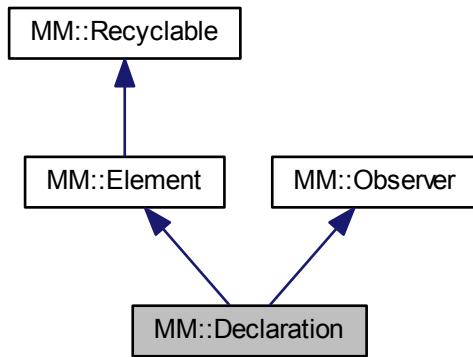
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[ConverterNodeBehavior.h](#)

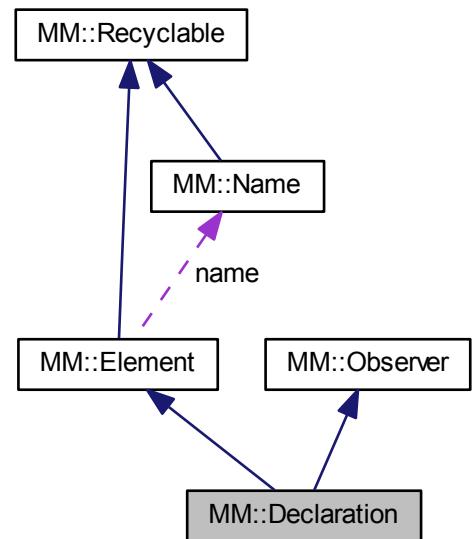
6.19 MM::Declaration Class Reference

```
#include <Declaration.h>
```

Inheritance diagram for MM::Declaration:



Collaboration diagram for MM::Declaration:



Public Member Functions

- `Declaration (MM::Name *type, MM::Name *name, MM::Map< MM::Name *, MM::Node *, MM::Name::Compare > *interfaces)`

- `~Declaration ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::VOID update (MM::Observable *observable, MM::VOID *aux, MM::UINT32 message, MM::VOID *object)`
- `MM::Name * getTypeName ()`
- `MM::VOID setDefinition (MM::Definition *def)`
- `MM::Definition * getDefinition ()`
- `MM::Node * getInterface (MM::Name *name)`
- `MM::VOID addInterface (MM::Machine *m, MM::Node *node)`
- `MM::VOID removeInterface (MM::Machine *m, MM::Node *node)`
- `MM::VOID begin (MM::Instance *i, MM::Machine *m)`
- `MM::VOID end (MM::Instance *i, MM::Machine *m)`
- `MM::VOID change (MM::Instance *i, MM::Machine *m)`
- `MM::VOID toString (MM::String *buf)`
- `MM::VOID toString (MM::String *buf, MM::UINT32 indent)`

Additional Inherited Members

6.19.1 Constructor & Destructor Documentation

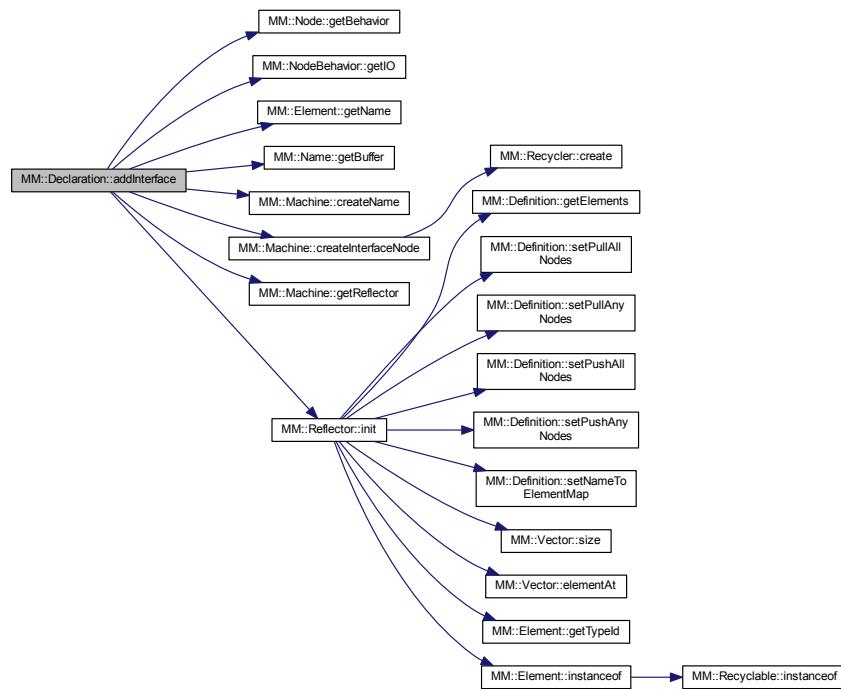
6.19.1.1 `Declaration::Declaration (MM::Name * type, MM::Name * name, MM::Map<MM::Name *, MM::Node *, MM::Name::Compare > * interfaces)`

6.19.1.2 `Declaration::~Declaration ()`

6.19.2 Member Function Documentation

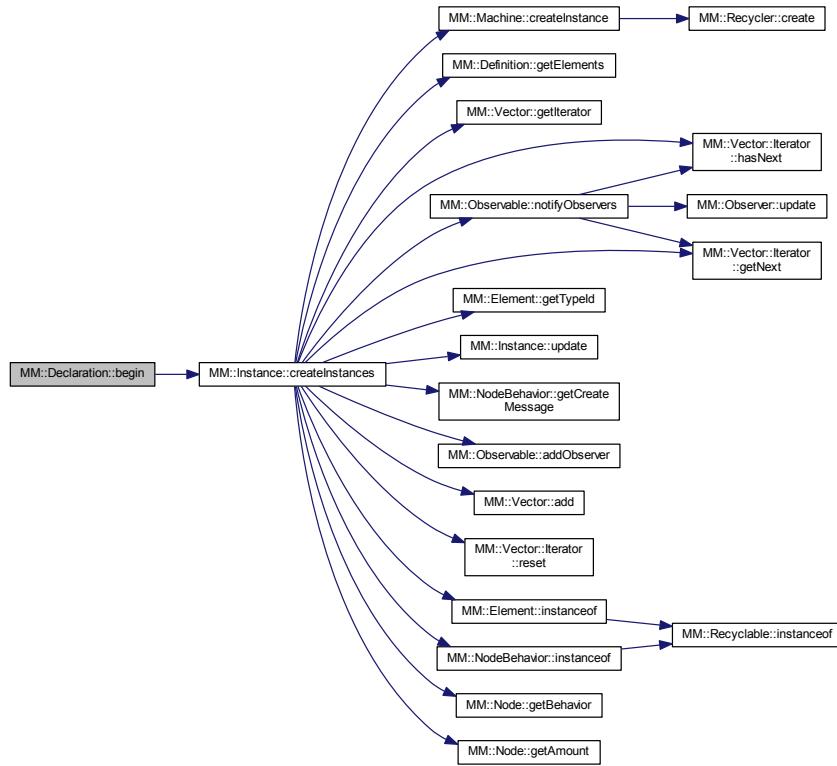
6.19.2.1 MM::VOID Declaration::addInterface (MM::Machine * m, MM::Node * node)

Here is the call graph for this function:



6.19.2.2 MM::VOID Declaration::begin (MM::Instance * i, MM::Machine * m)

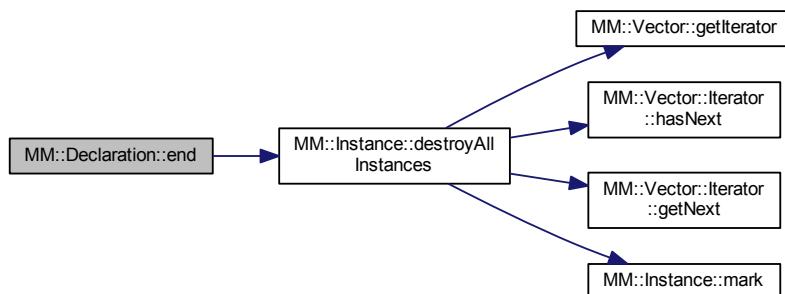
Here is the call graph for this function:



6.19.2.3 MM::VOID Declaration::change (MM::Instance * i, MM::Machine * m)

6.19.2.4 MM::VOID Declaration::end (MM::Instance * i, MM::Machine * m)

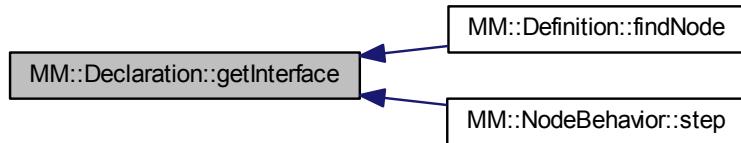
Here is the call graph for this function:



6.19.2.5 **MM::Definition * Declaration::getDefinition ()**

6.19.2.6 **MM::Node * Declaration::getInterface (MM::Name * name)**

Here is the caller graph for this function:



6.19.2.7 **MM::TID Declaration::getTypId () [virtual]**

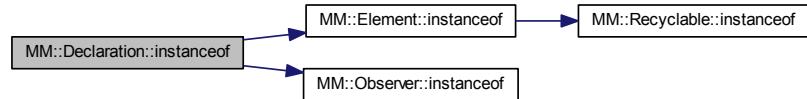
Reimplemented from [MM::Element](#).

6.19.2.8 **MM::Name * Declaration::getTypeName ()**

6.19.2.9 **MM::BOOLEAN Declaration::instanceof (MM::TID tid) [virtual]**

Reimplemented from [MM::Element](#).

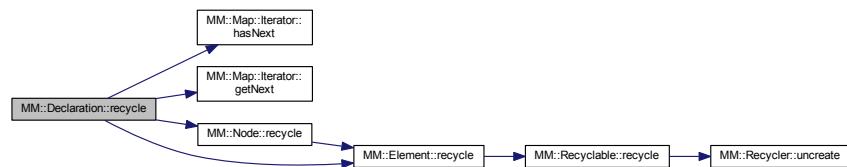
Here is the call graph for this function:



6.19.2.10 **MM::VOID Declaration::recycle (MM::Recycler * r) [virtual]**

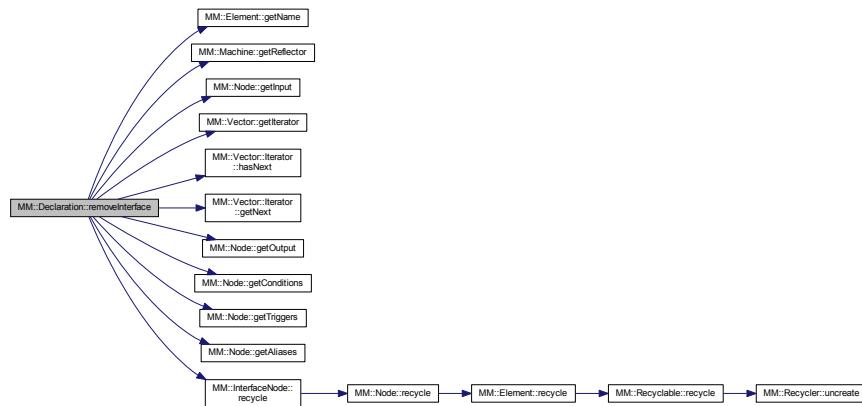
Reimplemented from [MM::Recyclable](#).

Here is the call graph for this function:



6.19.2.11 MM::VOID Declaration::removeInterface (MM::Machine * *m*, MM::Node * *node*)

Here is the call graph for this function:



6.19.2.12 MM::VOID Declaration::setDefinition (MM::Definition * *def*)

6.19.2.13 MM::VOID Declaration::toString (MM::String * *buf*) [virtual]

Implements [MM::Element](#).

6.19.2.14 MM::VOID Declaration::toString (MM::String * *buf*, MM::UINT32 *indent*) [virtual]

Implements [MM::Element](#).

Here is the call graph for this function:



6.19.2.15 MM::VOID Declaration::update (MM::Observable * *observable*, MM::VOID * *aux*, MM::UINT32 *message*, MM::VOID * *object*) [virtual]

Implements [MM::Observer](#).

The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Declaration.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Declaration.cpp](#)

6.20 Declaration Class Reference

The [Declaration](#) abstraction defines that a single named instance of a specified definition exists in each instance of the definition that contains the declaration.

```
#include <Declaration.h>
```

6.20.1 Detailed Description

The [Declaration](#) abstraction defines that a single named instance of a specified definition exists in each instance of the definition that contains the declaration.

Note

Declarations are observable such that elements internal to its type definition become exposed through interface nodes.

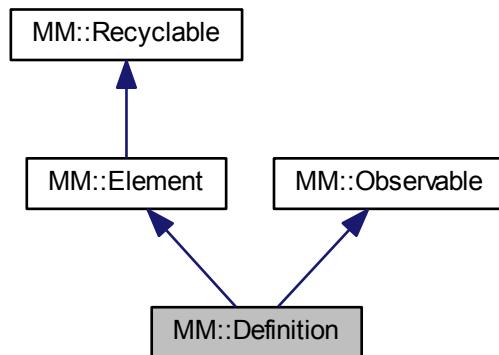
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Declaration.h](#)

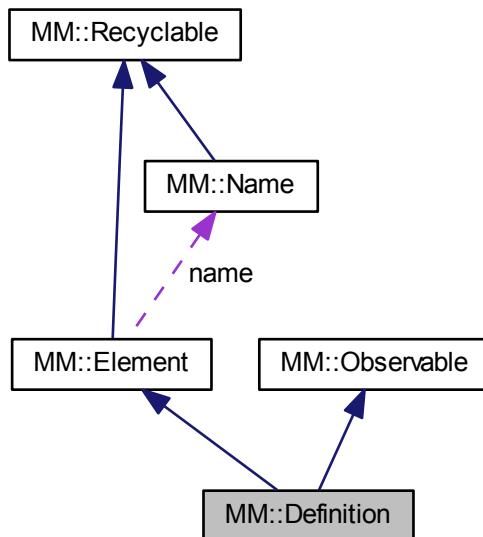
6.21 MM::Definition Class Reference

```
#include <Definition.h>
```

Inheritance diagram for MM::Definition:



Collaboration diagram for MM::Definition:



Public Member Functions

- `Definition (MM::Name *name, MM::Vector< MM::Element * > *elements)`
- `Definition (MM::Vector< MM::Element * > *elements)`
- `~Definition ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::Vector< MM::Node * > * getPullAllNodes ()`
- `MM::Vector< MM::Node * > * getPullAnyNodes ()`
- `MM::Vector< MM::Node * > * getPushAllNodes ()`
- `MM::Vector< MM::Node * > * getPushAnyNodes ()`
- `MM::VOID prioritize (MM::Node *node)`
- `MM::VOID deprioritize (MM::Node *node)`
- `MM::VOID addPullAllNode (MM::Node *node)`
- `MM::VOID addPullAnyNode (MM::Node *node)`
- `MM::VOID addPushAllNode (MM::Node *node)`
- `MM::VOID addPushAnyNode (MM::Node *node)`
- `MM::VOID removePullAllNode (MM::Node *node)`
- `MM::VOID removePullAnyNode (MM::Node *node)`
- `MM::VOID removePushAllNode (MM::Node *node)`
- `MM::VOID removePushAnyNode (MM::Node *node)`
- `MM::VOID setPullAllNodes (MM::Vector< MM::Node * > *pullAllNodes)`
- `MM::VOID setPullAnyNodes (MM::Vector< MM::Node * > *pullAnyNodes)`
- `MM::VOID setPushAllNodes (MM::Vector< MM::Node * > *pushAllNodes)`
- `MM::VOID setPushAnyNodes (MM::Vector< MM::Node * > *pushAnyNodes)`
- `MM::VOID setNameToElementMap (MM::Map< MM::Name *, MM::Element *, MM::Name::Compare > *n2e)`

- MM::VOID setParent (MM::Definition *parent)
- MM::Node * findNode (MM::Name *name, MM::NodeBehavior::IO direction)
- MM::Definition * findDeclaredDefinition (MM::Name *name)
- MM::Definition * findQueriedDefinition (MM::Name *name)
- MM::Element * getElement (MM::Name *name)
- MM::VOID putElement (MM::Name *name, MM::Element *element)
- MM::VOID removeElement (MM::Name *name)
- MM::VOID addElement (MM::Element *element)
- MM::VOID removeElement (MM::Element *element)
- MM::BOOLEAN containsElement (MM::Element *element)
- MM::VOID clearElements ()
- MM::Vector< MM::Element * > * getElements ()
- MM::VOID toString (MM::String *buf)
- MM::VOID toString (MM::String *buf, MM::UINT32 indent)

Static Public Attributes

- static const MM::CHAR LBRACE_CHAR = '{'
- static const MM::CHAR RBRACE_CHAR = '}'
- static const MM::CHAR COMMA_CHAR = ','

Additional Inherited Members

6.21.1 Constructor & Destructor Documentation

6.21.1.1 Definition::Definition (MM::Name * *name*, MM::Vector< MM::Element * > * *elements*)

name -> *element*

6.21.1.2 Definition::Definition (MM::Vector< MM::Element * > * *elements*)

6.21.1.3 Definition::~Definition ()

6.21.2 Member Function Documentation

6.21.2.1 MM::VOID Definition::addElement (MM::Element * *element*)

Here is the caller graph for this function:



6.21.2.2 MM::VOID MM::Definition::addPullAllNode (MM::Node * node)

6.21.2.3 MM::VOID MM::Definition::addPullAnyNode (MM::Node * node)

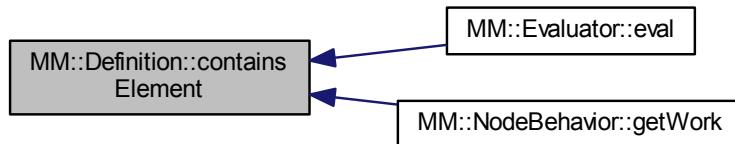
6.21.2.4 MM::VOID MM::Definition::addPushAllNode (MM::Node * node)

6.21.2.5 MM::VOID MM::Definition::addPushAnyNode (MM::Node * node)

6.21.2.6 MM::VOID Definition::clearElements ()

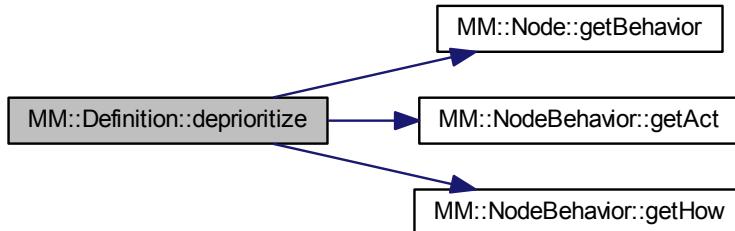
6.21.2.7 MM::BOOLEAN Definition::containsElement (MM::Element * element)

Here is the caller graph for this function:



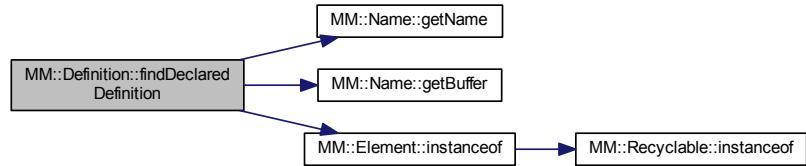
6.21.2.8 MM::VOID Definition::deprioritize (MM::Node * node)

Here is the call graph for this function:

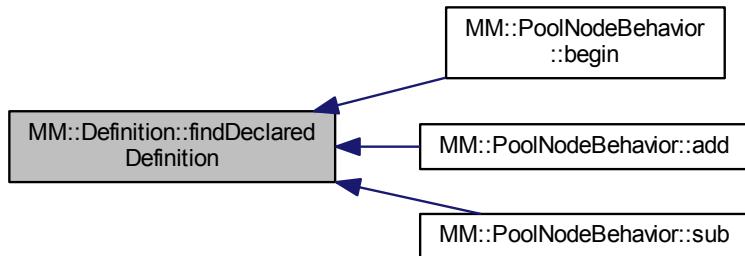


6.21.2.9 **MM::Definition * Definition::findDeclaredDefinition (MM::Name * name)**

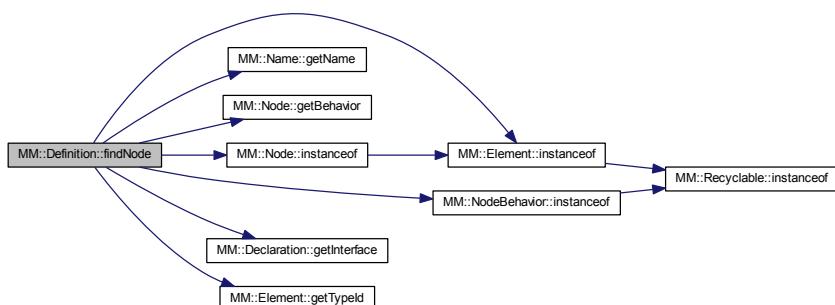
Here is the call graph for this function:



Here is the caller graph for this function:

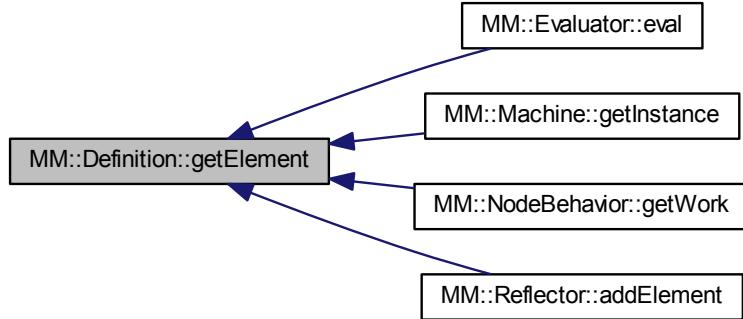
6.21.2.10 **MM::Node * Definition::findNode (MM::Name * name, MM::NodeBehavior::IO direction)**

Here is the call graph for this function:

6.21.2.11 **MM::Definition* MM::Definition::findQueriedDefinition (MM::Name * name)**

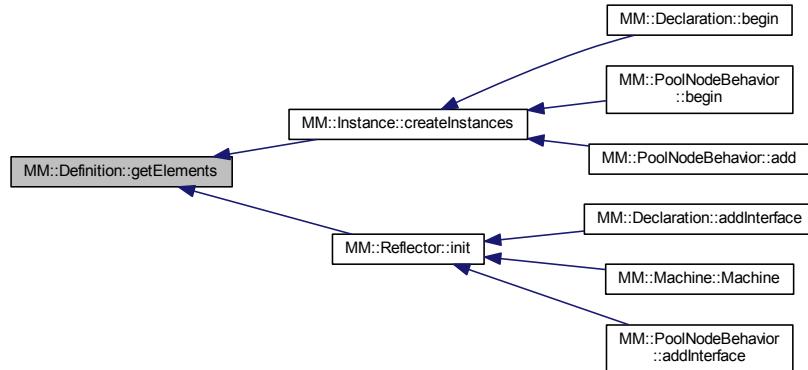
6.21.2.12 MM::Element * Definition::getElement (MM::Name * name)

Here is the caller graph for this function:



6.21.2.13 MM::Vector< MM::Element * > * Definition::getElements ()

Here is the caller graph for this function:



6.21.2.14 MM::Vector< MM::Node * > * Definition::getPullAllNodes ()

6.21.2.15 MM::Vector< MM::Node * > * Definition::getPullAnyNodes ()

6.21.2.16 MM::Vector< MM::Node * > * Definition::getPushAllNodes ()

6.21.2.17 MM::Vector< MM::Node * > * Definition::getPushAnyNodes ()

6.21.2.18 MM::TID Definition::getTypeId () [virtual]

Reimplemented from [MM::Element](#).

6.21.2.19 MM::BOOLEAN Definition::instanceof(MM::TID *tid*) [virtual]

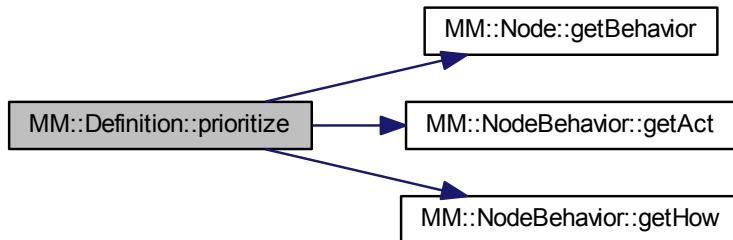
Reimplemented from [MM::Element](#).

Here is the call graph for this function:



6.21.2.20 MM::VOID Definition::prioritize(MM::Node * node)

Here is the call graph for this function:



6.21.2.21 MM::VOID Definition::putElement(MM::Name * name, MM::Element * element)

6.21.2.22 MM::VOID Definition::recycle(MM::Recycler * r) [virtual]

Reimplemented from [MM::Recyclable](#).

Here is the call graph for this function:



6.21.2.23 MM::VOID Definition::removeElement (MM::Name * name)

Here is the caller graph for this function:



6.21.2.24 MM::VOID Definition::removeElement (MM::Element * element)

6.21.2.25 MM::VOID MM::Definition::removePullAllNode (MM::Node * node)

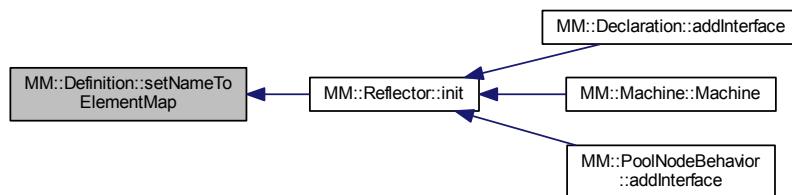
6.21.2.26 MM::VOID MM::Definition::removePushAnyNode (MM::Node * node)

6.21.2.27 MM::VOID MM::Definition::removePushAllNode (MM::Node * node)

6.21.2.28 MM::VOID MM::Definition::removePushAnyNode (MM::Node * node)

6.21.2.29 MM::VOID Definition::setNameToElementMap (MM::Map< MM::Name *, MM::Element *, MM::Name::Compare > * n2e)

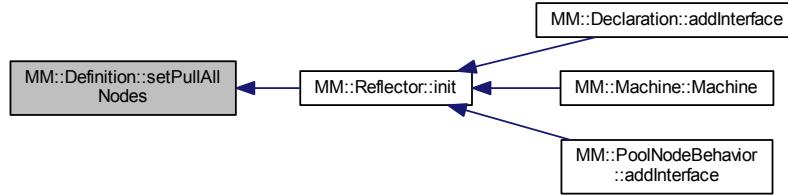
Here is the caller graph for this function:



6.21.2.30 MM::VOID Definition::setParent (MM::Definition * parent)

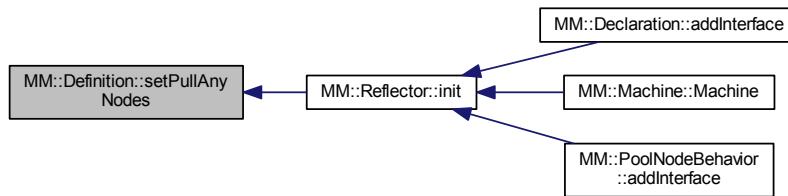
6.21.2.31 MM::VOID Definition::setPullAllNodes (MM::Vector< MM::Node * > * pullAllNodes)

Here is the caller graph for this function:



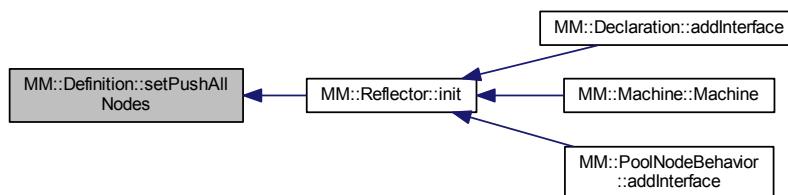
6.21.2.32 MM::VOID Definition::setPullAnyNodes (MM::Vector< MM::Node * > * pullAnyNodes)

Here is the caller graph for this function:



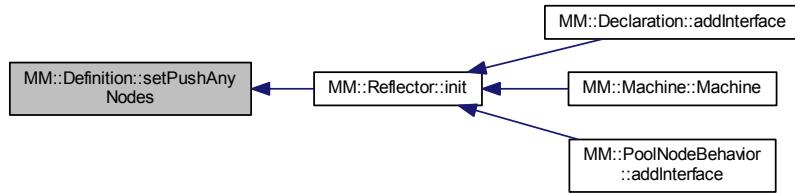
6.21.2.33 MM::VOID Definition::setPushAllNodes (MM::Vector< MM::Node * > * pushAllNodes)

Here is the caller graph for this function:



6.21.2.34 MM::VOID Definition::setPushAnyNodes (MM::Vector< MM::Node * > * pullAnyNodes)

Here is the caller graph for this function:



6.21.2.35 MM::VOID Definition::toString (MM::String * buf) [virtual]

Implements [MM::Element](#).

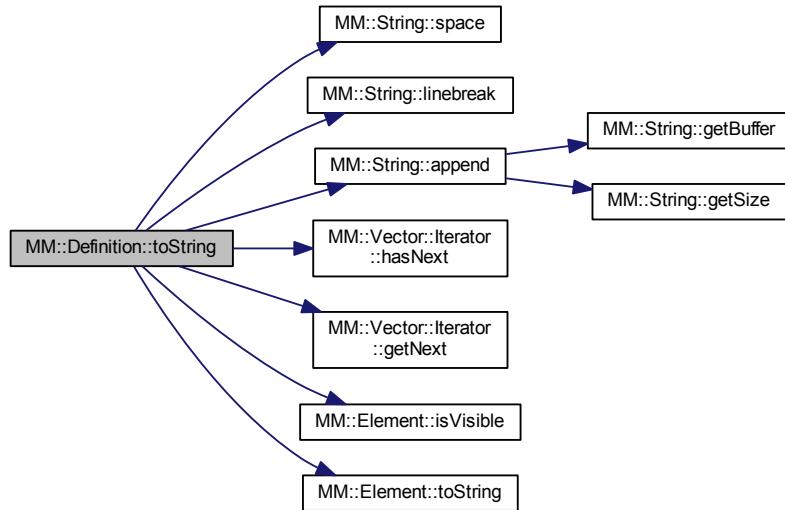
Here is the caller graph for this function:



6.21.2.36 MM::VOID Definition::toString (MM::String * buf, MM::UINT32 indent) [virtual]

Implements [MM::Element](#).

Here is the call graph for this function:



6.21.3 Member Data Documentation

6.21.3.1 const MM::CHAR Definition::COMMA_CHAR = ',' [static]

right brace character

6.21.3.2 const MM::CHAR Definition::LBRACE_CHAR = '{' [static]

6.21.3.3 const MM::CHAR Definition::RBRACE_CHAR = '}' [static]

left brace character

The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Definition.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Definition.cpp](#)

6.22 Definition Class Reference

The definition abstraction and the elements contained within it define the structure and behavior of instances created for the global scope, instance pools or declarations.

```
#include <Definition.h>
```

6.22.1 Detailed Description

The definition abstraction and the elements contained within it define the structure and behavior of instances created for the global scope, instance pools or declarations.

Note

Definitions are observable, such that notifications about structural changes are sent to observers, enabling structural reflection.

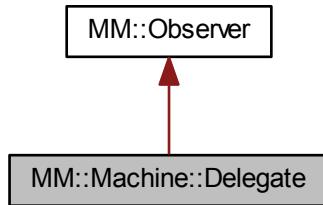
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Definition.h](#)

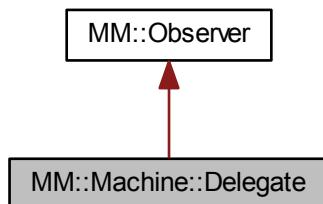
6.23 MM::Machine::Delegate Class Reference

```
#include <Machine.h>
```

Inheritance diagram for MM::Machine::Delegate:



Collaboration diagram for MM::Machine::Delegate:



Public Member Functions

- [Delegate \(MM::Observable *observable, MM::UINT32 caller, MM::CALLBACK callback\)](#)
- [~Delegate \(\)](#)
- [MM::TID getTypeId \(\)](#)
- [MM::BOOLEAN instanceof \(MM::TID tid\)](#)
- [MM::VOID update \(MM::Observable *observable, MM::VOID *aux, MM::UINT32 message, MM::VOID *object\)](#)

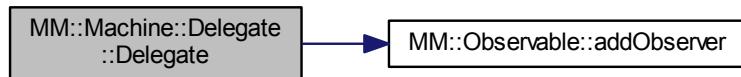
6.23.1 Detailed Description

global scope instance

6.23.2 Constructor & Destructor Documentation

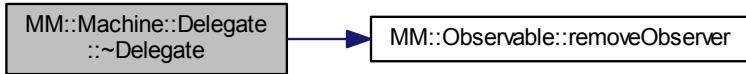
6.23.2.1 `MM::Machine::Delegate::Delegate (MM::Observable * observable, MM::UINT32 caller, MM::CALLBACK callback) [inline]`

Here is the call graph for this function:



6.23.2.2 `MM::Machine::Delegate::~Delegate () [inline]`

Here is the call graph for this function:



6.23.3 Member Function Documentation

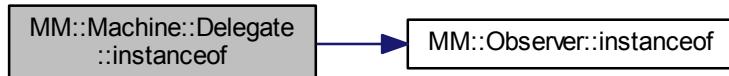
6.23.3.1 `MM::TID MM::Machine::Delegate::getTypeId () [inline], [virtual]`

Reimplemented from [MM::Observer](#).

6.23.3.2 `MM::BOOLEAN MM::Machine::Delegate::instanceof (MM::TID tid) [inline], [virtual]`

Reimplemented from [MM::Observer](#).

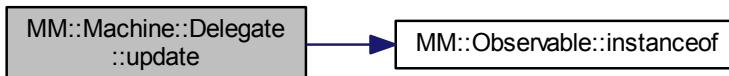
Here is the call graph for this function:



6.23.3.3 MM::VOID MM::Machine::Delegate::update (MM::Observable * observable, MM::VOID * aux, MM::UINT32 message, MM::VOID * object) [inline], [virtual]

Implements [MM::Observer](#).

Here is the call graph for this function:



The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Machine.h](#)

6.24 Deletion Class Reference

[Deletion](#) elements enable deleting elements from definitions using the qualified name, through evaluation between steps.

```
#include <Deletion.h>
```

6.24.1 Detailed Description

[Deletion](#) elements enable deleting elements from definitions using the qualified name, through evaluation between steps.

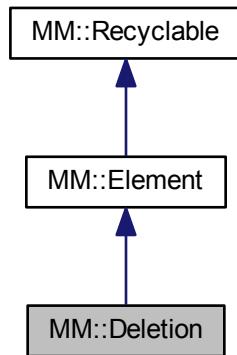
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Deletion.h](#)

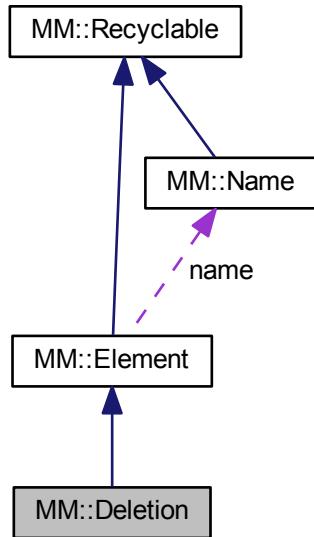
6.25 MM::Deletion Class Reference

```
#include <Deletion.h>
```

Inheritance diagram for MM::Deletion:



Collaboration diagram for MM::Deletion:



Public Member Functions

- `Deletion (MM::Name *name)`
- `Deletion (MM::Location *loc, MM::Name *name)`
- `~Deletion ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`

- MM::Location * getLocation ()
- MM::VOID toString (MM::String *buf)
- MM::VOID toString (MM::String *buf, MM::UINT32 indent)

Additional Inherited Members

6.25.1 Constructor & Destructor Documentation

6.25.1.1 Deletion::Deletion (MM::Name * name)

delete source location

6.25.1.2 Deletion::Deletion (MM::Location * loc, MM::Name * name)

6.25.1.3 Deletion::~Deletion ()

6.25.2 Member Function Documentation

6.25.2.1 MM::Location * Deletion::getLocation ()

6.25.2.2 MM::TID Deletion::getTypeId() [virtual]

Reimplemented from [MM::Element](#).

6.25.2.3 MM::BOOLEAN Deletion::instanceof (MM::TID tid) [virtual]

Reimplemented from [MM::Element](#).

Here is the call graph for this function:



6.25.2.4 MM::VOID Deletion::recycle (MM::Recycler * r) [virtual]

Reimplemented from [MM::Recyclable](#).

Here is the call graph for this function:



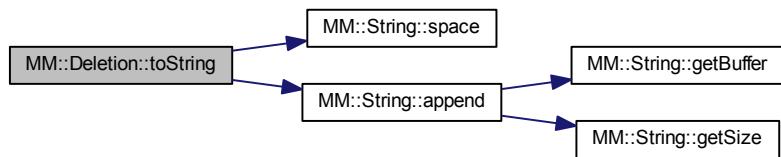
6.25.2.5 MM::VOID Deletion::toString (MM::String * buf) [virtual]

Implements [MM::Element](#).

6.25.2.6 MM::VOID Deletion::toString (MM::String * buf, MM::UINT32 indent) [virtual]

Implements [MM::Element](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Deletion.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Deletion.cpp](#)

6.26 DieExp Class Reference

The [DieExp](#) abstraction enables simulating a roll of a die, and can be used to randomize the amount of resources that can or must flow along a resource connection (flow edge).

```
#include <DieExp.h>
```

6.26.1 Detailed Description

The [DieExp](#) abstraction enables simulating a roll of a die, and can be used to randomize the amount of resources that can or must flow along a resource connection (flow edge).

Note

The expression is evaluated at most once during a step.

The documentation for this class was generated from the following file:

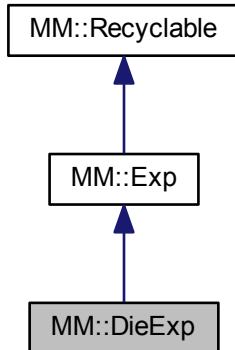
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[DieExp.h](#)

6.27 MM::DieExp Class Reference

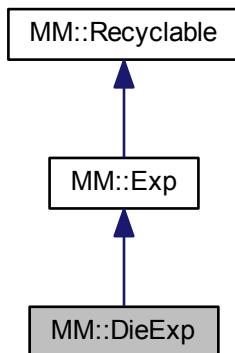
Defines the [DieExp](#) class.

```
#include <DieExp.h>
```

Inheritance diagram for MM::DieExp:



Collaboration diagram for MM::DieExp:



Public Member Functions

- `DieExp (MM::UINT32 max)`
- `DieExp (MM::UINT32 max, MM::Location *maxLoc, MM::Location *dieLoc)`
- `~DieExp ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`

Retrieves the type id of a DieExp object.

- `MM::BOOLEAN instanceof (MM::TID tid)`

Assesses if an object is an instance of a type tid.

- `MM::UINT32 getMax ()`

Serializes a DieExp object into a String buffer.

- [MM::VOID toString \(MM::String *buf\)](#)
Serializes a [DieExp](#) object into a [String](#) buffer.

Additional Inherited Members

6.27.1 Detailed Description

Defines the [DieExp](#) class.

6.27.2 Constructor & Destructor Documentation

6.27.2.1 DieExp::DieExp (MM::UINT32 max)

die location

Constructs a [DieExp](#) object.

Parameters

<i>name</i>	name
-------------	------

Returns

new [DieExp](#) object

6.27.2.2 DieExp::DieExp (MM::UINT32 max, MM::Location * maxLoc, MM::Location * dieLoc)

Constructs a [DieExp](#) object.

Parameters

<i>max</i>	maximum die roll
<i>maxLoc</i>	source location of max keyword
<i>dieLoc</i>	source location of die keyword

Returns

new [DieExp](#) object

6.27.2.3 DieExp::~DieExp ()

Deconstructs a [DieExp](#) object.

6.27.3 Member Function Documentation

6.27.3.1 MM::UINT32 DieExp::getMax ()

Serializes a [DieExp](#) object into a [String](#) buffer.

Parameters

<code>buf</code>	String buffer to serialize this object into
------------------	---

Here is the caller graph for this function:

**6.27.3.2 MM::TID DieExp::getTypeId() [virtual]**

Retrieves the type id of a [DieExp](#) object.

Returns

type id

Reimplemented from [MM::Exp](#).

6.27.3.3 MM::BOOLEAN DieExp::instanceof(MM::TID tid) [virtual]

Assesses if an object is an instance of a type tid.

Parameters

<code>tid</code>	type id
------------------	---------

Returns

MM_TRUE if this object is instance of tid, MM_FALSE otherwise

Reimplemented from [MM::Exp](#).

Here is the call graph for this function:

**6.27.3.4 MM::VOID DieExp::recycle(MM::Recycler * r) [virtual]**

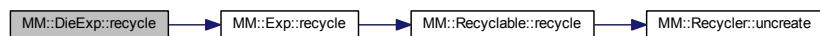
Recycles an [DieExp](#) object in a [Recycler](#).

Parameters

<i>r</i>	Recycler
----------	----------

Implements [MM::Exp](#).

Here is the call graph for this function:



6.27.3.5 MM::VOID DieExp::toString (MM::String * buf) [virtual]

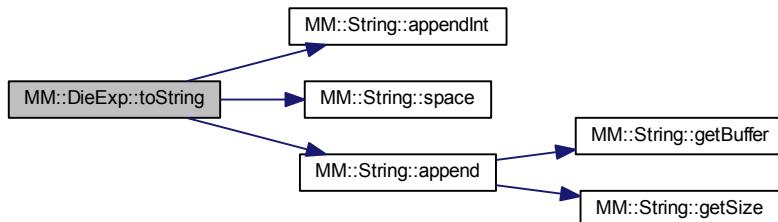
Serializes a [DieExp](#) object into a [String](#) buffer.

Parameters

<i>buf</i>	String buffer to serialize this object into
------------	---

Implements [MM::Exp](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[DieExp.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[DieExp.cpp](#)

6.28 Disablement Class Reference

The Disable abstraction defines a node instance was disabled during a step because one of its conditions is false.

```
#include <Disablement.h>
```

6.28.1 Detailed Description

The Disable abstraction defines a node instance was disabled during a step because one of its conditions is false.

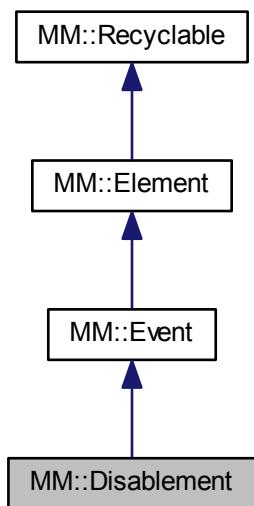
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Disablement.h](#)

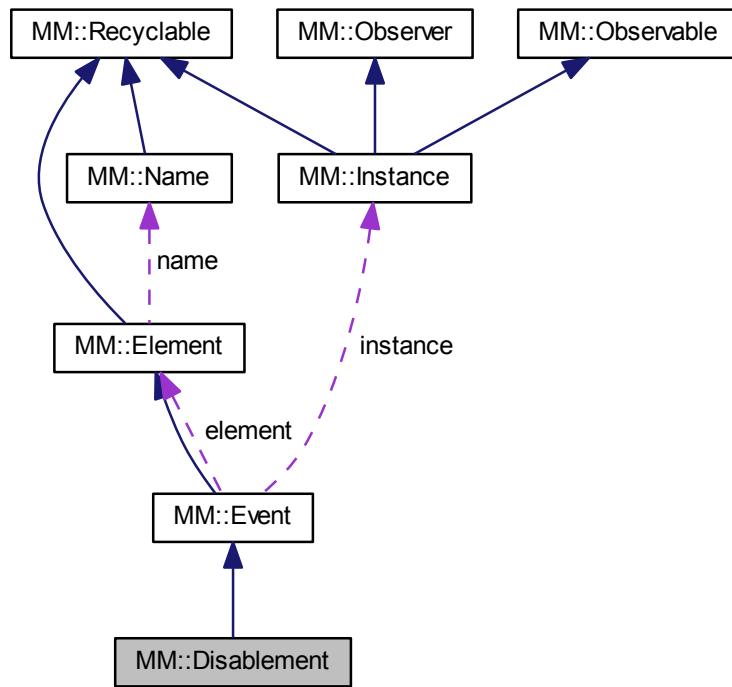
6.29 MM::Disablement Class Reference

```
#include <Disablement.h>
```

Inheritance diagram for MM::Disablement:



Collaboration diagram for MM::Disablement:



Public Member Functions

- `Disablement (MM::Name *name)`
- `Disablement (MM::Location *loc, MM::Name *name)`
- `Disablement (MM::Instance *instance, MM::Node *node)`
- `~Disablement ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::Location * getLocation ()`
- `MM::Name * getName ()`
- `MM::MESSAGE getMessage ()`
- `MM::VOID toString (MM::String *buf)`
- `MM::VOID toString (MM::String *buf, MM::UINT32 indent)`

Additional Inherited Members

6.29.1 Constructor & Destructor Documentation

6.29.1.1 Disablement::Disablement (`MM::Name * name`)

disable keyword location (parsed)

6.29.1.2 `Disablement::Disablement (MM::Location * loc, MM::Name * name)`

6.29.1.3 `Disablement::Disablement (MM::Instance * instance, MM::Node * node)`

6.29.1.4 `Disablement::~Disablement ()`

6.29.2 Member Function Documentation

6.29.2.1 `MM::Location * Disablement::getLocation ()`

6.29.2.2 `MM::MESSAGE Disablement::getMessage () [virtual]`

Implements [MM::Event](#).

6.29.2.3 `MM::Name* MM::Disablement::getName ()`

6.29.2.4 `MM::TID Disablement::getTypeld () [virtual]`

Reimplemented from [MM::Event](#).

6.29.2.5 `MM::BOOLEAN Disablement::instanceof (MM::TID tid) [virtual]`

Reimplemented from [MM::Event](#).

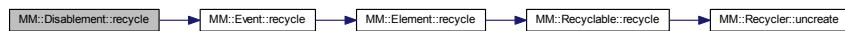
Here is the call graph for this function:



6.29.2.6 `MM::VOID Disablement::recycle (MM::Recycler * r) [virtual]`

Reimplemented from [MM::Recyclable](#).

Here is the call graph for this function:



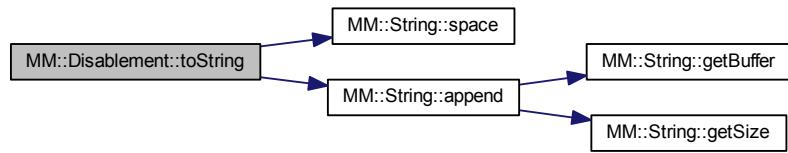
6.29.2.7 `MM::VOID Disablement::toString (MM::String * buf) [virtual]`

Implements [MM::Event](#).

6.29.2.8 `MM::VOID Disablement::toString (MM::String * buf, MM::UINT32 indent) [virtual]`

Implements [MM::Event](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Disablement.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Disablement.cpp](#)

6.30 DrainNodeBehavior Class Reference

The [DrainNodeBehavior](#) abstraction defines the behavior of drain nodes.

```
#include <DrainNodeBehavior.h>
```

6.30.1 Detailed Description

The [DrainNodeBehavior](#) abstraction defines the behavior of drain nodes.

Note

Strategy

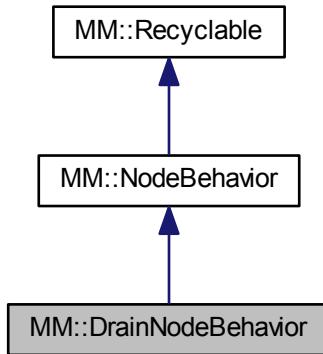
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[DrainNodeBehavior.h](#)

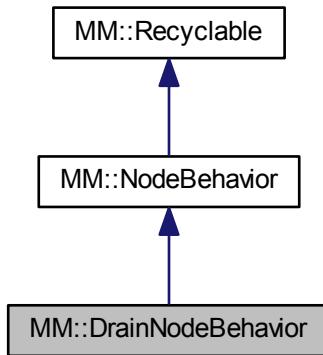
6.31 MM::DrainNodeBehavior Class Reference

```
#include <DrainNodeBehavior.h>
```

Inheritance diagram for MM::DrainNodeBehavior:



Collaboration diagram for MM::DrainNodeBehavior:



Public Member Functions

- `DrainNodeBehavior (MM::NodeBehavior::IO io, MM::NodeBehavior::When when, MM::NodeBehavior::How how)`
- `~DrainNodeBehavior ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::UINT32 getCreateMessage ()`
- `MM::UINT32 getUpdateMessage ()`
- `MM::UINT32 getDeleteMessage ()`
- `MM::VOID stepPushAny (MM::Node *node, MM::Instance *i, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)`

- MM::VOID stepPullAll (MM::Node *node, MM::Instance *i, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)
- MM::VOID stepPushAll (MM::Node *node, MM::Instance *i, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)
- MM::VOID begin (MM::Instance *i, MM::Machine *m, MM::Node *n)
- MM::VOID end (MM::Instance *i, MM::Machine *m, MM::Node *n)
- MM::VOID change (MM::Instance *i, MM::Machine *m, MM::Node *n)
- MM::VOID add (MM::Instance *i, MM::Machine *m, MM::Node *n, MM::UINT32 amount)
- MM::VOID sub (MM::Instance *i, MM::Machine *m, MM::Node *n, MM::UINT32 amount)
- MM::UINT32 getCapacity (MM::Instance *i, MM::Node *n)
- MM::UINT32 getResources (MM::Instance *i, MM::Node *n)
- MM::BOOLEAN hasCapacity (MM::Instance *i, MM::Node *n, MM::UINT32 amount)
- MM::BOOLEAN hasResources (MM::Instance *i, MM::Node *n, MM::UINT32 amount)
- MM::VOID toString (MM::String *buf)
- MM::VOID toString (MM::String *buf, MM::Name *name)

Additional Inherited Members

6.31.1 Constructor & Destructor Documentation

6.31.1.1 DrainNodeBehavior::DrainNodeBehavior (MM::NodeBehavior::IO *io*, MM::NodeBehavior::When *when*, MM::NodeBehavior::How *how*)

drain keyword length

6.31.1.2 DrainNodeBehavior::~DrainNodeBehavior ()

6.31.2 Member Function Documentation

6.31.2.1 MM::VOID DrainNodeBehavior::add (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*, MM::UINT32 *amount*) [virtual]

Implements [MM::NodeBehavior](#).

6.31.2.2 MM::VOID DrainNodeBehavior::begin (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*) [virtual]

Implements [MM::NodeBehavior](#).

6.31.2.3 MM::VOID DrainNodeBehavior::change (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*) [virtual]

Implements [MM::NodeBehavior](#).

6.31.2.4 MM::VOID DrainNodeBehavior::end (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*) [virtual]

Implements [MM::NodeBehavior](#).

6.31.2.5 MM::UINT32 DrainNodeBehavior::getCapacity (MM::Instance * *i*, MM::Node * *n*) [virtual]

Implements [MM::NodeBehavior](#).

6.31.2.6 **MM::UINT32 DrainNodeBehavior::getCreateMessage() [virtual]**

Implements [MM::NodeBehavior](#).

6.31.2.7 **MM::UINT32 DrainNodeBehavior::getDeleteMessage() [virtual]**

Implements [MM::NodeBehavior](#).

6.31.2.8 **MM::UINT32 DrainNodeBehavior::getResources(MM::Instance * i, MM::Node * n) [virtual]**

Implements [MM::NodeBehavior](#).

6.31.2.9 **MM::TID DrainNodeBehavior::getTypeld() [virtual]**

Reimplemented from [MM::NodeBehavior](#).

6.31.2.10 **MM::UINT32 DrainNodeBehavior::getUpdateMessage() [virtual]**

Implements [MM::NodeBehavior](#).

6.31.2.11 **MM::BOOLEAN DrainNodeBehavior::hasCapacity(MM::Instance * i, MM::Node * n, MM::UINT32 amount) [virtual]**

Implements [MM::NodeBehavior](#).

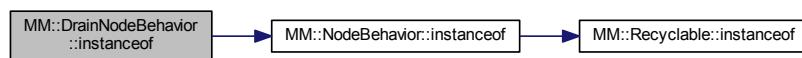
6.31.2.12 **MM::BOOLEAN DrainNodeBehavior::hasResources(MM::Instance * i, MM::Node * n, MM::UINT32 amount) [virtual]**

Implements [MM::NodeBehavior](#).

6.31.2.13 **MM::BOOLEAN DrainNodeBehavior::instanceof(MM::TID tid) [virtual]**

Reimplemented from [MM::NodeBehavior](#).

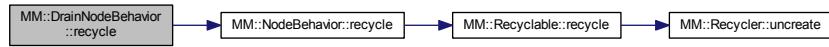
Here is the call graph for this function:



6.31.2.14 **MM::VOID DrainNodeBehavior::recycle(MM::Recycler * r) [virtual]**

Reimplemented from [MM::Recyclable](#).

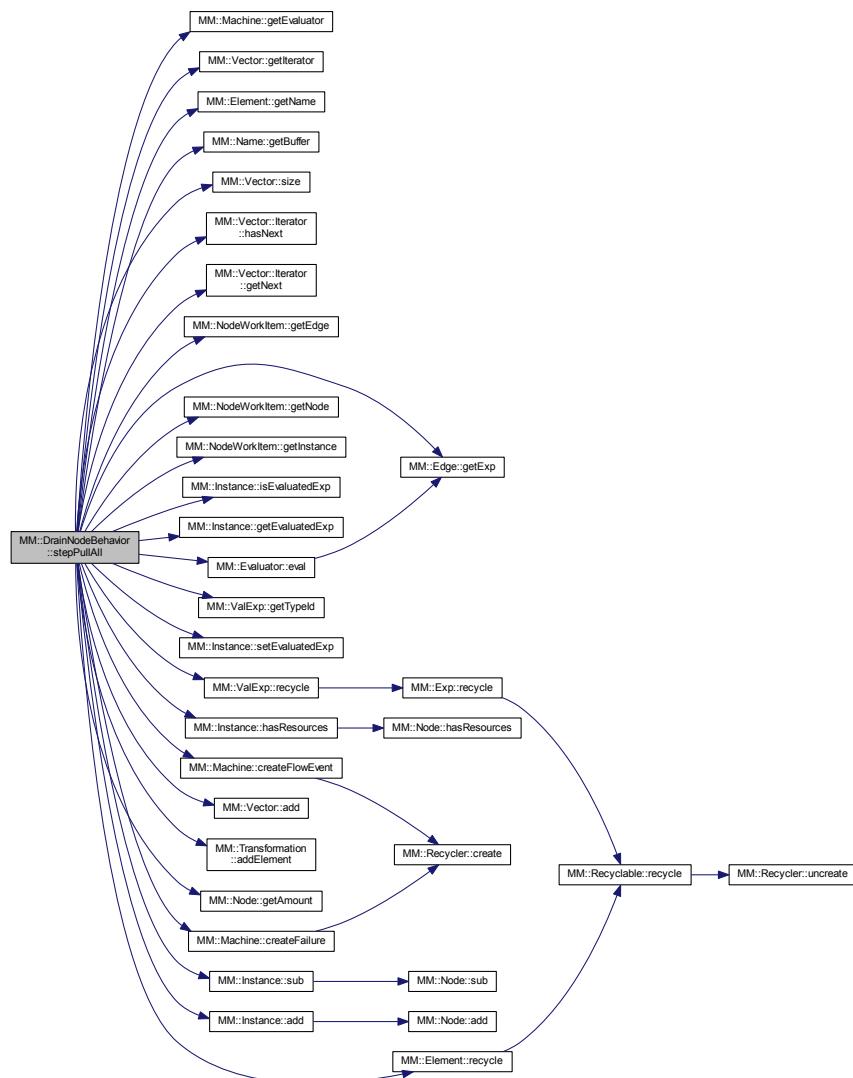
Here is the call graph for this function:



6.31.2.15 VOID DrainNodeBehavior::stepPullAll (MM::Node * node, MM::Instance * i, MM::Vector< MM::NodeWorkItem * > * work, MM::Machine * m, MM::Transition * tr) [virtual]

Implements [MM::NodeBehavior](#).

Here is the call graph for this function:



6.31.2.16 **MM::VOID DrainNodeBehavior::stepPushAll (MM::Node * node, MM::Instance * i, MM::Vector< MM::NodeWorkItem * > * work, MM::Machine * m, MM::Transition * tr) [virtual]**

Implements [MM::NodeBehavior](#).

6.31.2.17 **MM::VOID DrainNodeBehavior::stepPushAny (MM::Node * node, MM::Instance * i, MM::Vector< MM::NodeWorkItem * > * work, MM::Machine * m, MM::Transition * tr) [virtual]**

Reimplemented from [MM::NodeBehavior](#).

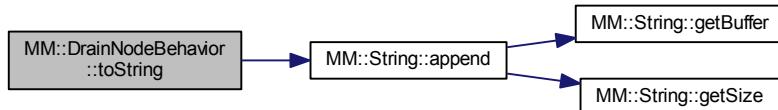
6.31.2.18 **MM::VOID DrainNodeBehavior::sub (MM::Instance * i, MM::Machine * m, MM::Node * n, MM::UINT32 amount) [virtual]**

Implements [MM::NodeBehavior](#).

6.31.2.19 **MM::VOID DrainNodeBehavior::toString (MM::String * buf) [virtual]**

Implements [MM::Recyclable](#).

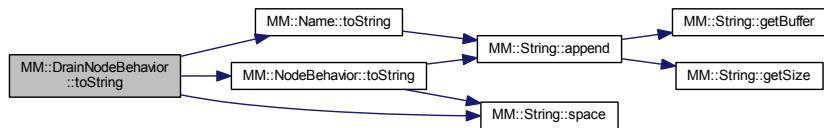
Here is the call graph for this function:



6.31.2.20 **MM::VOID DrainNodeBehavior::toString (MM::String * buf, MM::Name * name) [virtual]**

Reimplemented from [MM::NodeBehavior](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[DrainNodeBehavior.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[DrainNodeBehavior.cpp](#)

6.32 Edge Class Reference

The [Edge](#) abstraction is an abstract element connecting a source node and a target node.

```
#include <Edge.h>
```

6.32.1 Detailed Description

The [Edge](#) abstraction is an abstract element connecting a source node and a target node.

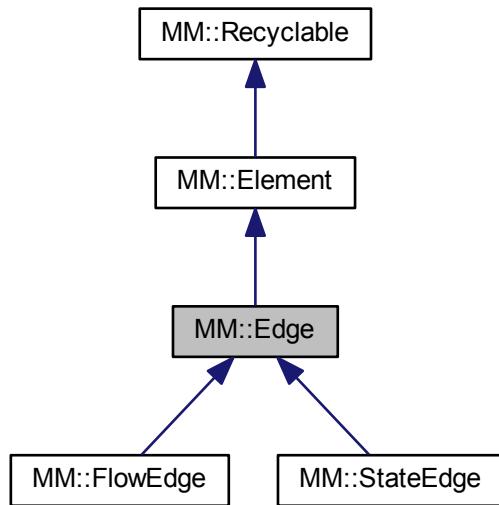
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Edge.h](#)

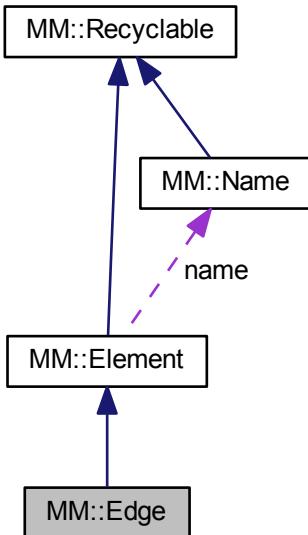
6.33 MM::Edge Class Reference

```
#include <Edge.h>
```

Inheritance diagram for MM::Edge:



Collaboration diagram for MM::Edge:



Public Member Functions

- `MM::VOID recycle (MM::Recycler *r)`
- `virtual MM::TID getTypeId ()`
- `virtual MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::Node * getSource ()`
- `MM::Node * getTarget ()`
- `MM::Exp * getExp ()`
- `MM::VOID setExp (MM::Exp *exp)`
- `MM::VOID setSource (MM::Node *src)`
- `MM::VOID setTarget (MM::Node *tgt)`
- `MM::Name * getSourceName ()`
- `MM::Name * getTargetName ()`
- `virtual MM::VOID toString (MM::String *buf)=0`
- `virtual MM::VOID toString (MM::String *buf, MM::UINT32 indent)`

Protected Member Functions

- `Edge (MM::Name *name, MM::Name *src, MM::Exp *exp, MM::Name *tgt)`
- `virtual ~Edge ()`

Additional Inherited Members

6.33.1 Constructor & Destructor Documentation

6.33.1.1 Edge::Edge (MM::Name * name, MM::Name * src, MM::Exp * exp, MM::Name * tgt) [protected]

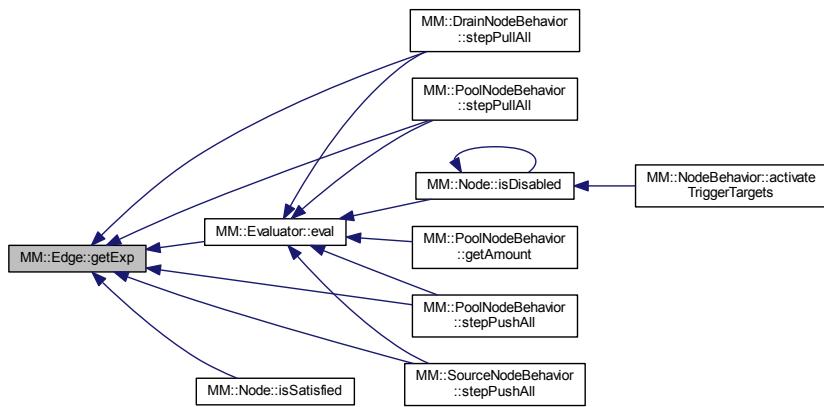
expression (parsed)

6.33.1.2 `Edge::~Edge()` [protected], [virtual]

6.33.2 Member Function Documentation

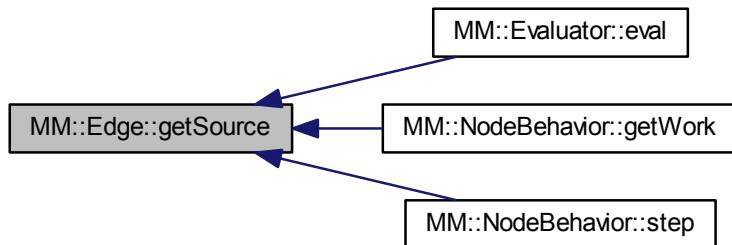
6.33.2.1 `MM::Exp * Edge::getExp()`

Here is the caller graph for this function:



6.33.2.2 `MM::Node * Edge::getSource()`

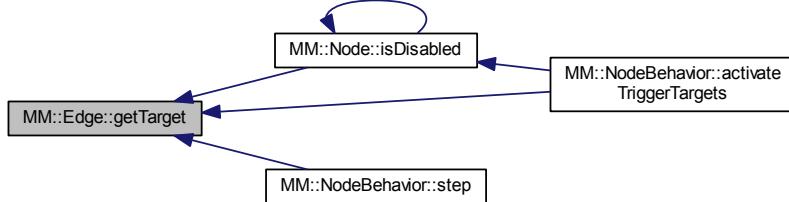
Here is the caller graph for this function:



6.33.2.3 `MM::Name * Edge::getSourceName()`

6.33.2.4 MM::Node * Edge::getTarget()

Here is the caller graph for this function:



6.33.2.5 MM::Name * Edge::getTargetName()

6.33.2.6 MM::TID Edge::getTypeld() [virtual]

Reimplemented from [MM::Element](#).

Reimplemented in [MM::FlowEdge](#), and [MM::StateEdge](#).

Here is the caller graph for this function:

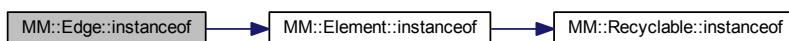


6.33.2.7 MM::BOOLEAN Edge::instanceof(MM::TID tid) [virtual]

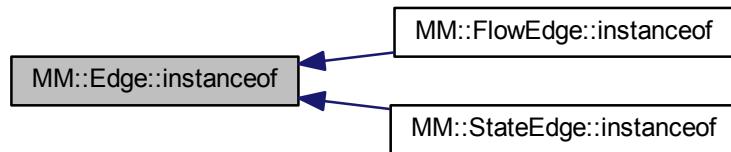
Reimplemented from [MM::Element](#).

Reimplemented in [MM::FlowEdge](#), and [MM::StateEdge](#).

Here is the call graph for this function:



Here is the caller graph for this function:

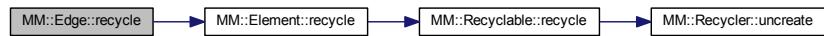


6.33.2.8 MM::VOID Edge::recycle(MM::Recycler * r) [virtual]

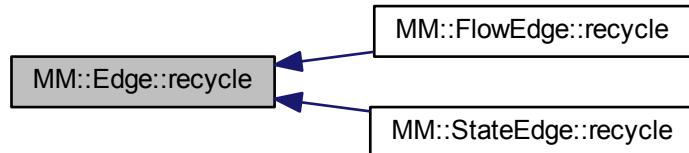
Reimplemented from [MM::Recyclable](#).

Reimplemented in [MM::FlowEdge](#), and [MM::StateEdge](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.33.2.9 MM::VOID Edge::setExp(MM::Exp * exp)

6.33.2.10 MM::VOID Edge::setSource (MM::Node * src)

Here is the caller graph for this function:



6.33.2.11 MM::VOID Edge::setTarget (MM::Node * tgt)

Here is the caller graph for this function:

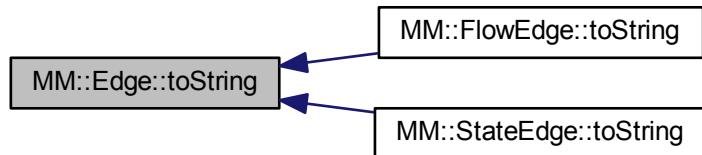


6.33.2.12 virtual MM::VOID MM::Edge::toString (MM::String * buf) [pure virtual]

Implements [MM::Element](#).

Implemented in [MM::StateEdge](#), and [MM::FlowEdge](#).

Here is the caller graph for this function:

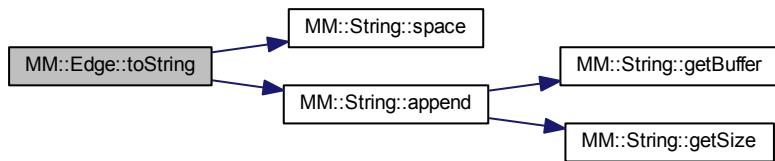


6.33.2.13 MM::VOID Edge::toString (MM::String * buf, MM::UINT32 indent) [virtual]

Implements [MM::Element](#).

Reimplemented in [MM::StateEdge](#), and [MM::FlowEdge](#).

Here is the call graph for this function:



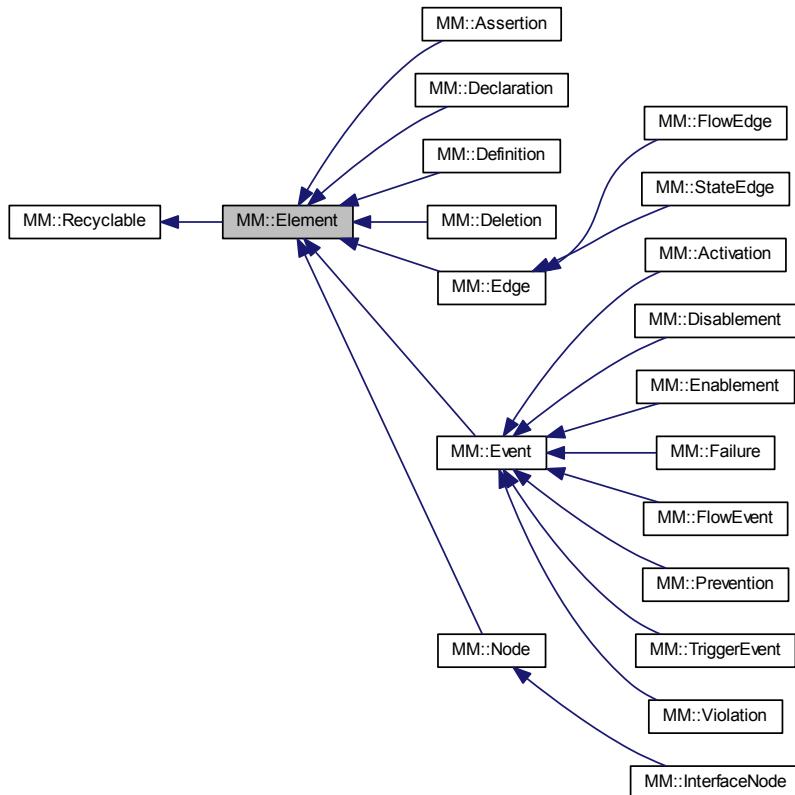
The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Edge.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Edge.cpp](#)

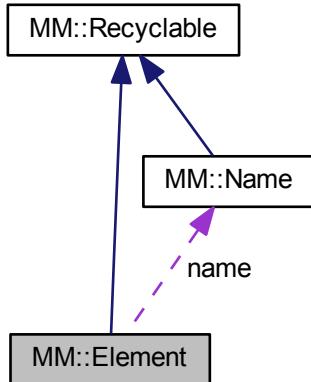
6.34 MM::Element Class Reference

```
#include <Element.h>
```

Inheritance diagram for MM::Element:



Collaboration diagram for MM::Element:



Public Member Functions

- virtual ~Element ()
- MM::VOID recycle (MM::Recycler *r)
- virtual MM::TID getTypeId ()
- virtual MM::BOOLEAN instanceof (MM::TID tid)
- MM::Name * getName ()
- MM::VOID setName (MM::Name *name)
- MM::BOOLEAN isVisible ()
- MM::VOID setVisible (MM::BOOLEAN visible)
- virtual MM::VOID begin (MM::Instance *i)
- virtual MM::VOID end (MM::Instance *i)
- virtual MM::VOID change (MM::Instance *i)
- virtual MM::VOID toString (MM::String *buf)=0
- virtual MM::VOID toString (MM::String *buf, MM::UINT32 indent)=0

Protected Member Functions

- Element (MM::Name *name)

Protected Attributes

- MM::Name * name
- MM::BOOLEAN visible

6.34.1 Constructor & Destructor Documentation

6.34.1.1 Element::Element (MM::Name * name) [protected]

6.34.1.2 Element::~Element () [virtual]

6.34.2 Member Function Documentation

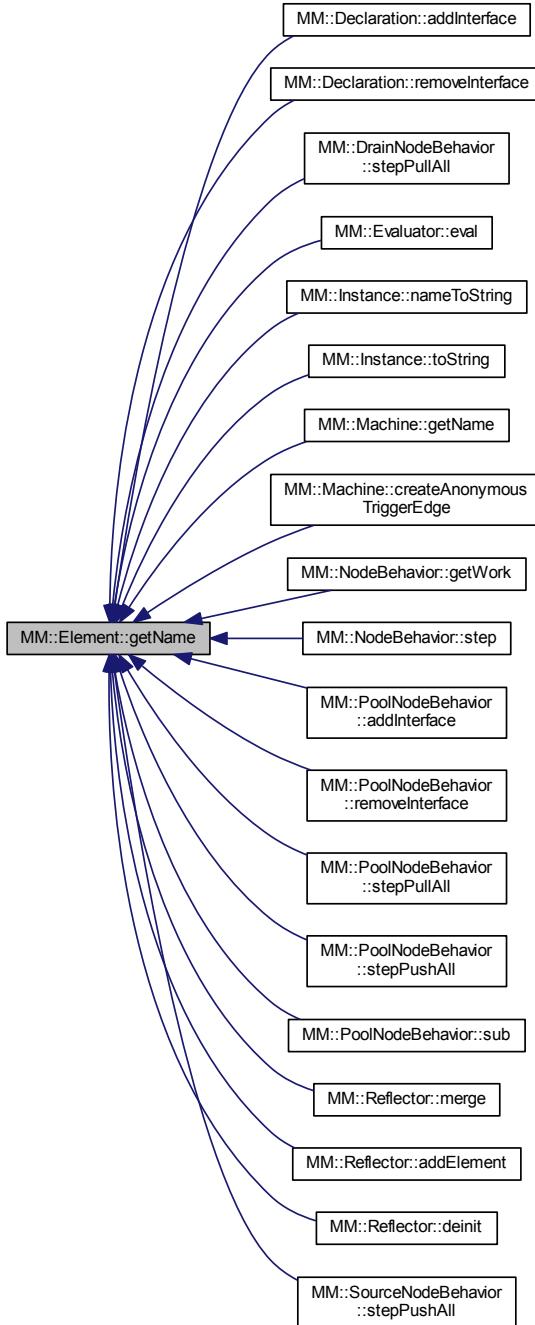
6.34.2.1 **MM::VOID Element::begin (MM::Instance * *i*) [virtual]**

6.34.2.2 **MM::VOID Element::change (MM::Instance * *i*) [virtual]**

6.34.2.3 **MM::VOID Element::end (MM::Instance * *i*) [virtual]**

6.34.2.4 MM::Name * Element::getName()

Here is the caller graph for this function:



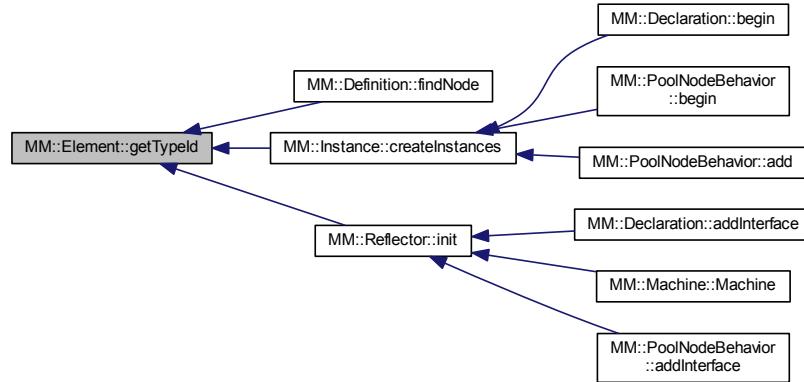
6.34.2.5 MM::TID Element::getTypeId() [virtual]

Reimplemented from [MM::Recyclable](#).

Reimplemented in [MM::Definition](#), [MM::FlowEvent](#), [MM::Declaration](#), [MM::Node](#), [MM::Assertion](#), [MM::Activation](#),

[MM::Disablement](#), [MM::Edge](#), [MM::Enablement](#), [MM::Failure](#), [MM::TriggerEvent](#), [MM::FlowEdge](#), [MM::Prevention](#), [MM::Violation](#), [MM::InterfaceNode](#), [MM::StateEdge](#), [MM::Deletion](#), and [MM::Event](#).

Here is the caller graph for this function:



6.34.2.6 MM::BOOLEAN Element::instanceof (MM::TID *tid*) [virtual]

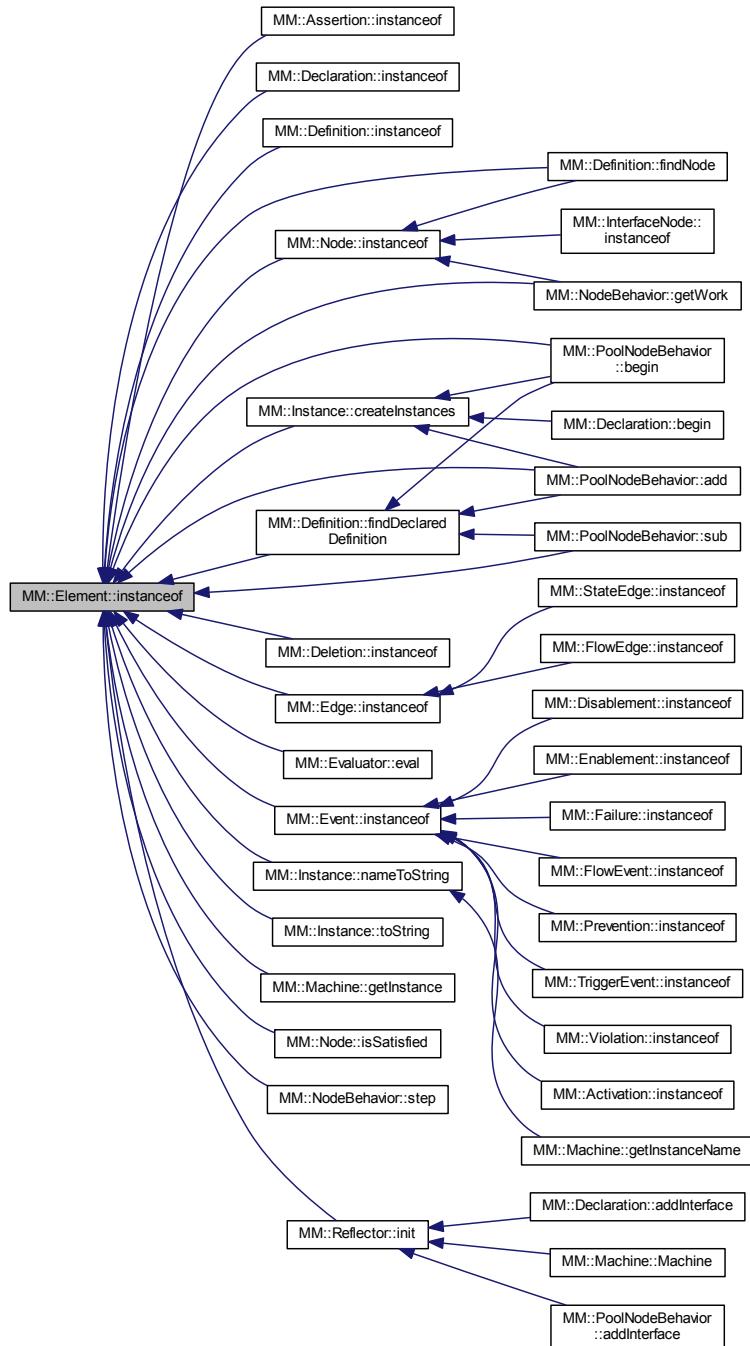
Reimplemented from [MM::Recyclable](#).

Reimplemented in [MM::Definition](#), [MM::FlowEvent](#), [MM::Declaration](#), [MM::Node](#), [MM::Assertion](#), [MM::Activation](#), [MM::Disablement](#), [MM::Edge](#), [MM::Enablement](#), [MM::Failure](#), [MM::TriggerEvent](#), [MM::FlowEdge](#), [MM::Prevention](#), [MM::Violation](#), [MM::InterfaceNode](#), [MM::StateEdge](#), [MM::Deletion](#), and [MM::Event](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.34.2.7 MM::BOOLEAN Element::isVisible()

Here is the caller graph for this function:

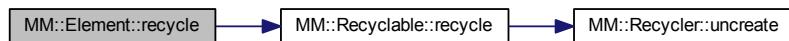


6.34.2.8 MM::VOID Element::recycle(MM::Recycler * r) [virtual]

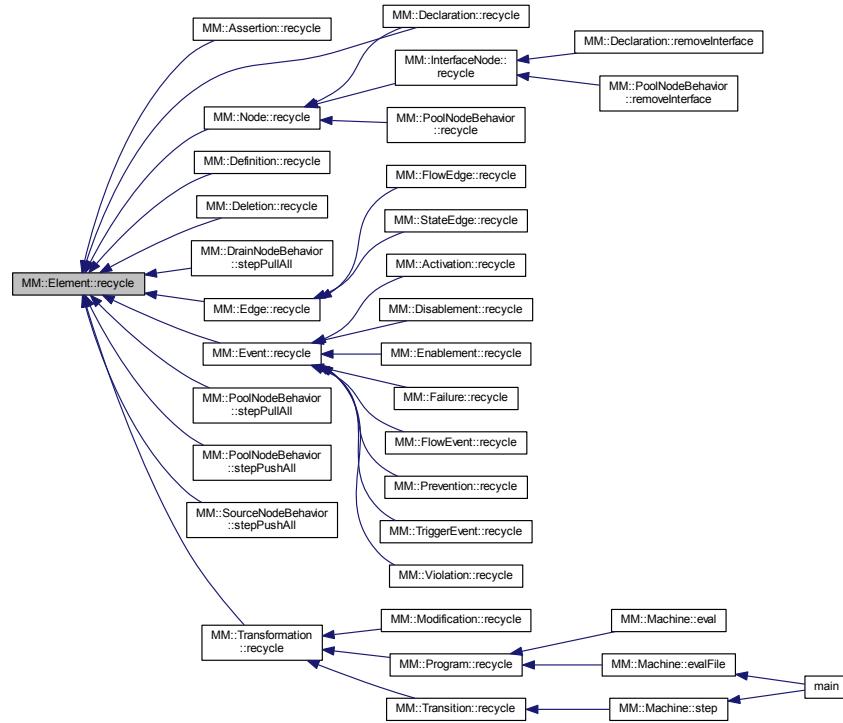
Reimplemented from [MM::Recyclable](#).

Reimplemented in [MM::FlowEvent](#), [MM::Node](#), [MM::Enablement](#), [MM::Failure](#), [MM::TriggerEvent](#), [MM::FlowEdge](#), [MM::Prevention](#), [MM::Violation](#), [MM::InterfaceNode](#), [MM::StateEdge](#), and [MM::Event](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.34.2.9 MM::VOID Element::setName (MM::Name * name)

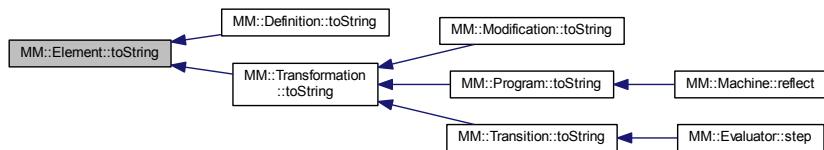
6.34.2.10 MM::VOID Element::setVisible (MM::BOOLEAN visible)

6.34.2.11 MM::VOID Element::toString (MM::String * buf) [pure virtual]

Implements [MM::Recyclable](#).

Implemented in [MM::Node](#), [MM::Definition](#), [MM::FlowEvent](#), [MM::Declaration](#), [MM::InterfaceNode](#), [MM::Edge](#), [MM::Assertion](#), [MM::Disablement](#), [MM::Enablement](#), [MM::Failure](#), [MM::TriggerEvent](#), [MM::Activation](#), [MM::Prevention](#), [MM::Event](#), [MM::Violation](#), [MM::StateEdge](#), [MM::FlowEdge](#), and [MM::Deletion](#).

Here is the caller graph for this function:



6.34.2.12 virtual MM::VOID MM::Element::toString(MM::String * buf, MM::UINT32 indent) [pure virtual]

Implemented in [MM::Node](#), [MM::Definition](#), [MM::FlowEvent](#), [MM::Declaration](#), [MM::Edge](#), [MM::Assertion](#), [MM::Disablement](#), [MM::Enablement](#), [MM::Failure](#), [MM::TriggerEvent](#), [MM::Activation](#), [MM::Prevention](#), [MM::Event](#), [MM::Violation](#), [MM::StateEdge](#), [MM::FlowEdge](#), and [MM::Deletion](#).

6.34.3 Member Data Documentation

6.34.3.1 MM::Name* MM::Element::name [protected]

6.34.3.2 MM::BOOLEAN MM::Element::visible [protected]

The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Element.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Element.cpp](#)

6.35 Element Class Reference

The [Element](#) abstraction is the abstract superclass of all program elements.

```
#include <Element.h>
```

6.35.1 Detailed Description

The [Element](#) abstraction is the abstract superclass of all program elements.

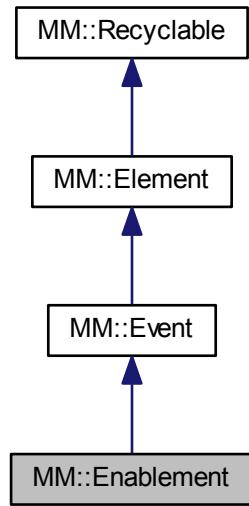
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Element.h](#)

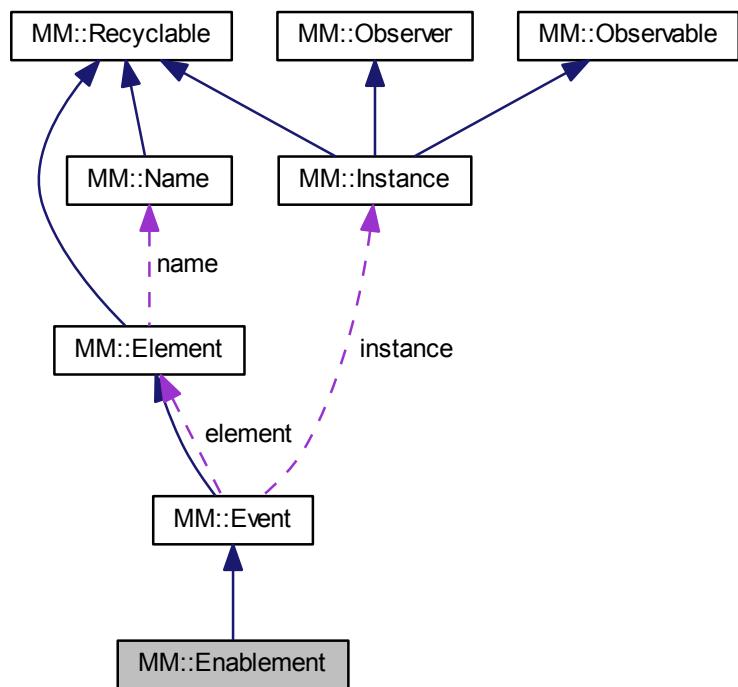
6.36 MM::Enablement Class Reference

```
#include <Enablement.h>
```

Inheritance diagram for MM::Enablement:



Collaboration diagram for MM::Enablement:



Public Member Functions

- `Enablement (MM::Name *name)`
- `Enablement (MM::Location *loc, MM::Name *name)`
- `Enablement (MM::Instance *instance, MM::Node *node)`
- `~Enablement ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::Location * getLocation ()`
- `MM::Name * getName ()`
- `MM::MESSAGE getMessage ()`
- `MM::VOID toString (MM::String *buf)`
- `MM::VOID toString (MM::String *buf, MM::UINT32 indent)`

Additional Inherited Members

6.36.1 Constructor & Destructor Documentation

6.36.1.1 Enablement::Enablement (`MM::Name * name`)

enable keyword location (parsed)

6.36.1.2 Enablement::Enablement (`MM::Location * loc, MM::Name * name`)

6.36.1.3 Enablement::Enablement (`MM::Instance * instance, MM::Node * node`)

6.36.1.4 Enablement::~Enablement ()

6.36.2 Member Function Documentation

6.36.2.1 MM::Location * Enablement::getLocation ()

6.36.2.2 MM::MESSAGE Enablement::getMessage () [virtual]

Implements [MM::Event](#).

6.36.2.3 MM::Name* MM::Enablement::getName ()

6.36.2.4 MM::TID Enablement::getTypeld () [virtual]

Reimplemented from [MM::Event](#).

6.36.2.5 MM::BOOLEAN Enablement::instanceof (`MM::TID tid`) [virtual]

Reimplemented from [MM::Event](#).

Here is the call graph for this function:



6.36.2.6 MM::VOID Enablement::recycle (MM::Recycler * r) [virtual]

Reimplemented from [MM::Element](#).

Here is the call graph for this function:



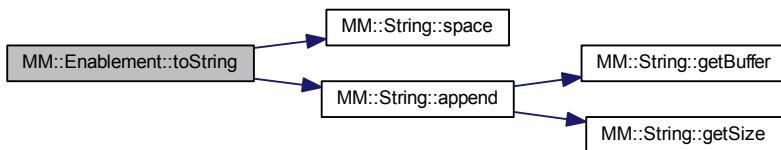
6.36.2.7 MM::VOID Enablement::toString (MM::String * buf) [virtual]

Implements [MM::Event](#).

6.36.2.8 MM::VOID Enablement::toString (MM::String * buf, MM::UINT32 indent) [virtual]

Implements [MM::Event](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Enablement.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Enablement.cpp](#)

6.37 Enablement Class Reference

The [Enablement](#) abstraction defines a node instance was enabled during a step because all of its conditions are true.

```
#include <Enablement.h>
```

6.37.1 Detailed Description

The [Enablement](#) abstraction defines a node instance was enabled during a step because all of its conditions are true.

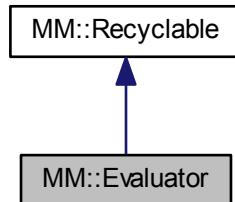
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Enablement.h](#)

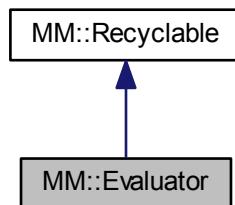
6.38 MM::Evaluator Class Reference

```
#include <Evaluator.h>
```

Inheritance diagram for MM::Evaluator:



Collaboration diagram for MM::Evaluator:



Public Member Functions

- `Evaluator (MM::Machine *m)`
- `~Evaluator ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::VOID toString (MM::String *buf)`
- `MM::VOID step (MM::Transition *tr)`
- `MM::ValExp * eval (MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::Exp *exp, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::OneExp *exp, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::ActiveExp *exp, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::DieExp *exp, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::OverrideExp *exp, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::VarExp *exp, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::AllExp *exp, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::BinExp *exp, MM::Instance *i, MM::Edge *e)`

- `MM::ValExp * eval (MM::UnExp *exp, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::BooleanValExp *exp, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::NumberValExp *exp, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::RangeValExp *exp, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::ValExp *e1, MM::Operator::OP op, MM::ValExp *e2, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::BooleanValExp *e1, MM::Operator::OP op, MM::BooleanValExp *e2, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::BooleanValExp *e1, MM::Operator::OP op, MM::NumberValExp *e2, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::BooleanValExp *e1, MM::Operator::OP op, MM::RangeValExp *e2, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::NumberValExp *e1, MM::Operator::OP op, MM::BooleanValExp *e2, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::NumberValExp *e1, MM::Operator::OP op, MM::NumberValExp *e2, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::NumberValExp *e1, MM::Operator::OP op, MM::RangeValExp *e2, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::RangeValExp *e1, MM::Operator::OP op, MM::BooleanValExp *e2, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::RangeValExp *e1, MM::Operator::OP op, MM::NumberValExp *e2, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::RangeValExp *e1, MM::Operator::OP op, MM::RangeValExp *e2, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::Operator::OP op, MM::ValExp *exp, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::Operator::OP op, MM::BooleanValExp *exp, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::Operator::OP op, MM::NumberValExp *exp, MM::Instance *i, MM::Edge *e)`
- `MM::ValExp * eval (MM::Operator::OP op, MM::RangeValExp *exp, MM::Instance *i, MM::Edge *e)`

6.38.1 Constructor & Destructor Documentation

6.38.1.1 `Evaluator::Evaluator (MM::Machine * m)`

6.38.1.2 `Evaluator::~Evaluator ()`

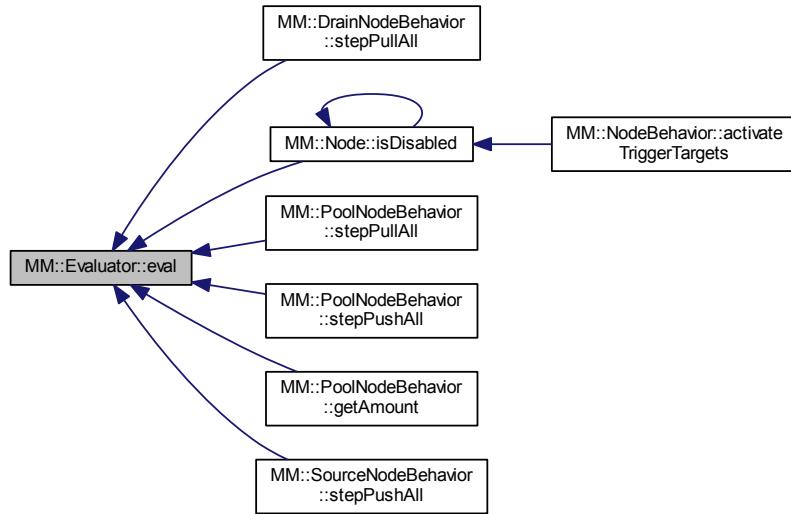
6.38.2 Member Function Documentation

6.38.2.1 `MM::ValExp * Evaluator::eval (MM::Instance * i, MM::Edge * e)`

Here is the call graph for this function:



Here is the caller graph for this function:



6.38.2.2 `MM::ValExp * Evaluator::eval (MM::Exp * exp, MM::Instance * i, MM::Edge * e)`

Here is the call graph for this function:



6.38.2.3 `MM::ValExp * Evaluator::eval (MM::OneExp * exp, MM::Instance * i, MM::Edge * e)`

6.38.2.4 `MM::ValExp * Evaluator::eval (MM::ActiveExp * exp, MM::Instance * i, MM::Edge * e)`

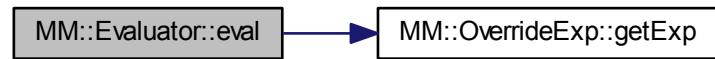
6.38.2.5 MM::ValExp * Evaluator::eval (MM::DieExp * exp, MM::Instance * i, MM::Edge * e)

Here is the call graph for this function:



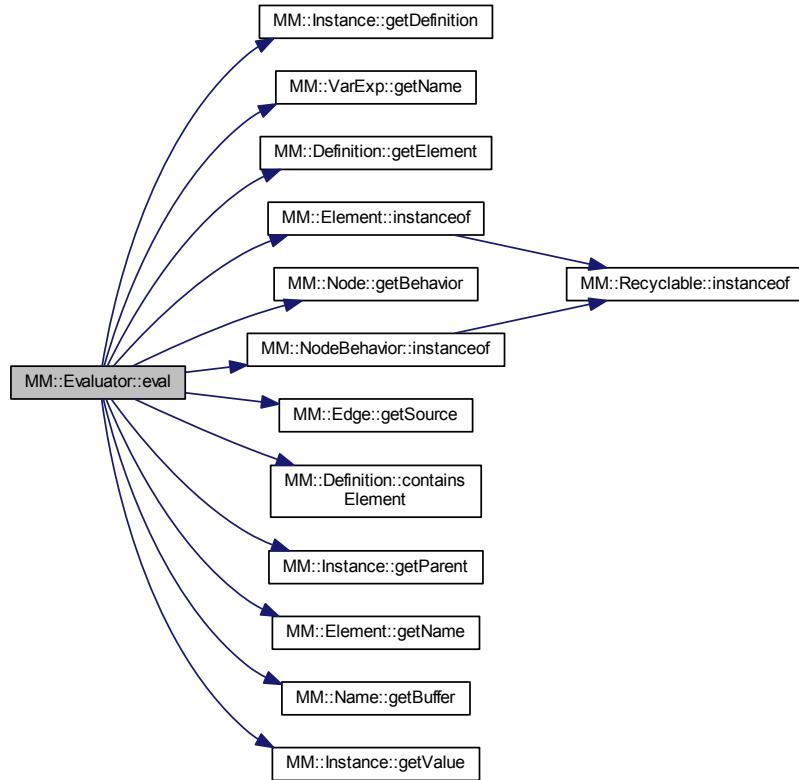
6.38.2.6 MM::ValExp * Evaluator::eval (MM::OverrideExp * exp, MM::Instance * i, MM::Edge * e)

Here is the call graph for this function:



6.38.2.7 MM::ValExp * Evaluator::eval (MM::VarExp * exp, MM::Instance * i, MM::Edge * e)

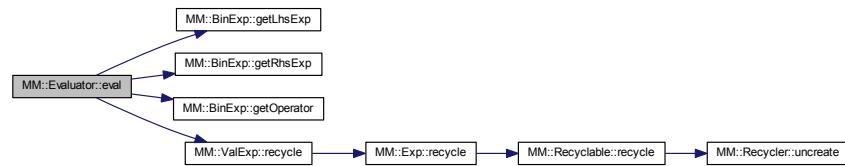
Here is the call graph for this function:



6.38.2.8 MM::ValExp * Evaluator::eval (MM::AllExp * exp, MM::Instance * i, MM::Edge * e)

6.38.2.9 MM::ValExp * Evaluator::eval (MM::BinExp * exp, MM::Instance * i, MM::Edge * e)

Here is the call graph for this function:



6.38.2.10 MM::ValExp * Evaluator::eval (MM::UnExp * exp, MM::Instance * i, MM::Edge * e)

Here is the call graph for this function:



6.38.2.11 MM::ValExp * Evaluator::eval (MM::BooleanValExp * exp, MM::Instance * i, MM::Edge * e)

Here is the call graph for this function:



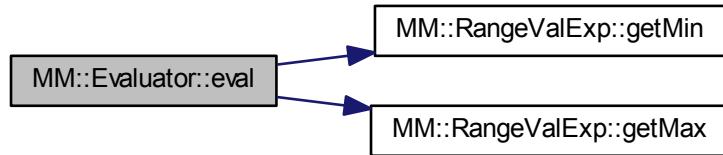
6.38.2.12 MM::ValExp * Evaluator::eval (MM::NumberValExp * exp, MM::Instance * i, MM::Edge * e)

Here is the call graph for this function:



6.38.2.13 `MM::ValExp * Evaluator::eval (MM::RangeValExp * exp, MM::Instance * i, MM::Edge * e)`

Here is the call graph for this function:



6.38.2.14 `MM::ValExp * Evaluator::eval (MM::ValExp * e1, MM::Operator::OP op, MM::ValExp * e2, MM::Instance * i, MM::Edge * e)`

Here is the call graph for this function:



6.38.2.15 `MM::ValExp * Evaluator::eval (MM::BooleanValExp * e1, MM::Operator::OP op, MM::BooleanValExp * e2, MM::Instance * i, MM::Edge * e)`

Here is the call graph for this function:



6.38.2.16 `MM::ValExp * Evaluator::eval (MM::BooleanValExp * e1, MM::Operator::OP op, MM::NumberValExp * e2, MM::Instance * i, MM::Edge * e)`

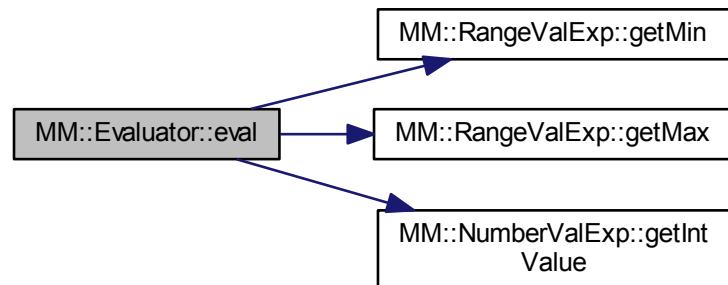
- 6.38.2.17 `MM::ValExp * Evaluator::eval (MM::BooleanValExp * e1, MM::Operator::OP op, MM::RangeValExp * e2, MM::Instance * i, MM::Edge * e)`
- 6.38.2.18 `MM::ValExp * Evaluator::eval (MM::NumberValExp * e1, MM::Operator::OP op, MM::BooleanValExp * e2, MM::Instance * i, MM::Edge * e)`
- 6.38.2.19 `MM::ValExp * Evaluator::eval (MM::NumberValExp * e1, MM::Operator::OP op, MM::NumberValExp * e2, MM::Instance * i, MM::Edge * e)`

Here is the call graph for this function:



- 6.38.2.20 `MM::ValExp * Evaluator::eval (MM::NumberValExp * e1, MM::Operator::OP op, MM::RangeValExp * e2, MM::Instance * i, MM::Edge * e)`

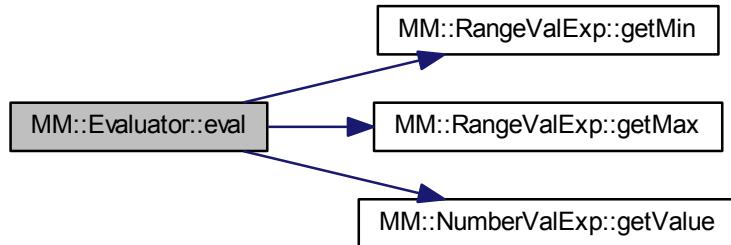
Here is the call graph for this function:



- 6.38.2.21 `MM::ValExp * Evaluator::eval (MM::RangeValExp * e1, MM::Operator::OP op, MM::BooleanValExp * e2, MM::Instance * i, MM::Edge * e)`

6.38.2.22 `MM::ValExp * Evaluator::eval (MM::RangeValExp * e1, MM::Operator::OP op, MM::NumberValExp * e2, MM::Instance * i, MM::Edge * e)`

Here is the call graph for this function:



6.38.2.23 `MM::ValExp * Evaluator::eval (MM::RangeValExp * e1, MM::Operator::OP op, MM::RangeValExp * e2, MM::Instance * i, MM::Edge * e)`

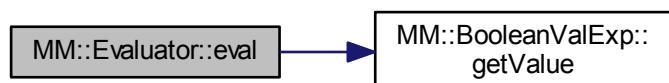
6.38.2.24 `MM::ValExp * Evaluator::eval (MM::Operator::OP op, MM::ValExp * exp, MM::Instance * i, MM::Edge * e)`

Here is the call graph for this function:



6.38.2.25 `MM::ValExp * Evaluator::eval (MM::Operator::OP op, MM::BooleanValExp * exp, MM::Instance * i, MM::Edge * e)`

Here is the call graph for this function:



6.38.2.26 **MM::ValExp * Evaluator::eval (MM::Operator::OP *op*, MM::NumberValExp * *exp*, MM::Instance * *i*, MM::Edge * *e*)**

Here is the call graph for this function:



6.38.2.27 **MM::ValExp * Evaluator::eval (MM::Operator::OP *op*, MM::RangeValExp * *exp*, MM::Instance * *i*, MM::Edge * *e*)**

6.38.2.28 **MM::TID Evaluator::getTypeld () [virtual]**

Reimplemented from [MM::Recyclable](#).

6.38.2.29 **MM::BOOLEAN Evaluator::instanceof (MM::TID *tid*) [virtual]**

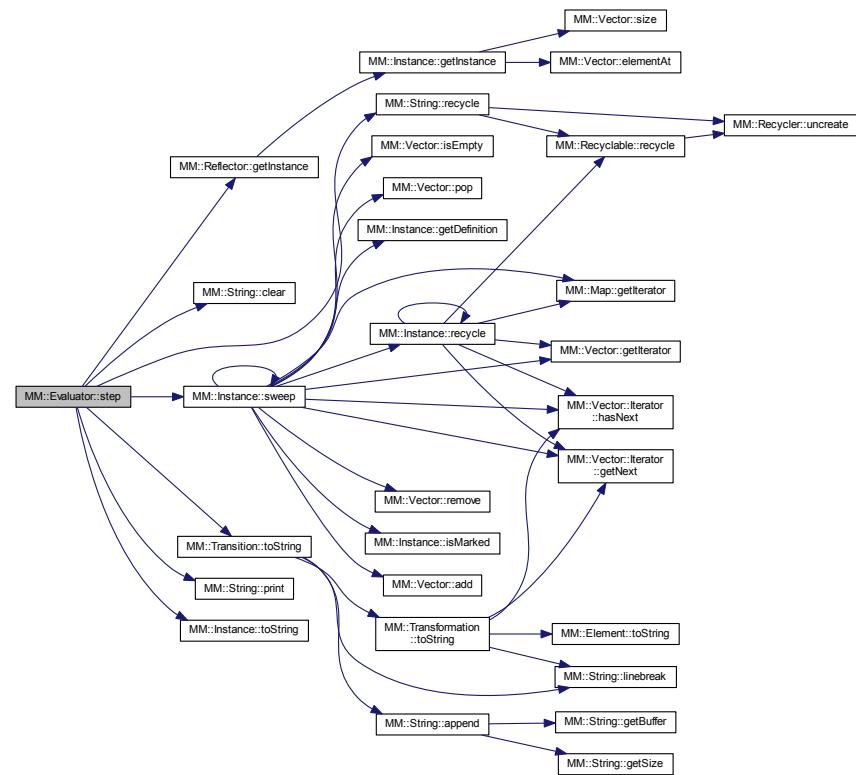
Reimplemented from [MM::Recyclable](#).

6.38.2.30 **MM::VOID Evaluator::recycle (MM::Recycler * *r*) [virtual]**

Reimplemented from [MM::Recyclable](#).

6.38.2.31 MM::VOID Evaluator::step (MM::Transition * tr)

Here is the call graph for this function:



6.38.2.32 MM::VOID Evaluator::toString (MM::String * buf) [virtual]

Implements [MM::Recyclable](#).

The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Evaluator.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Evaluator.cpp](#)

6.39 Evaluator Class Reference

The [Evaluator](#) evaluates program steps and the resulting changes to instances.

```
#include <Evaluator.h>
```

6.39.1 Detailed Description

The [Evaluator](#) evaluates program steps and the resulting changes to instances.

Note

Expressions are visited whereas nodes are interpreted.

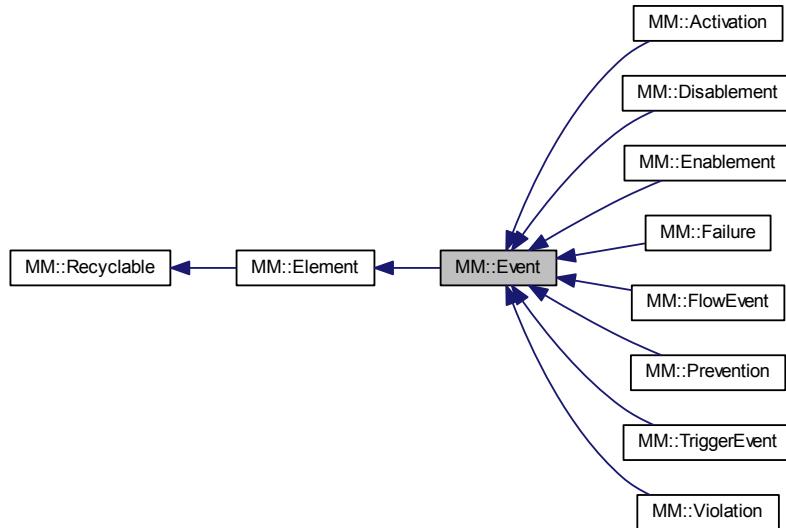
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Evaluator.h](#)

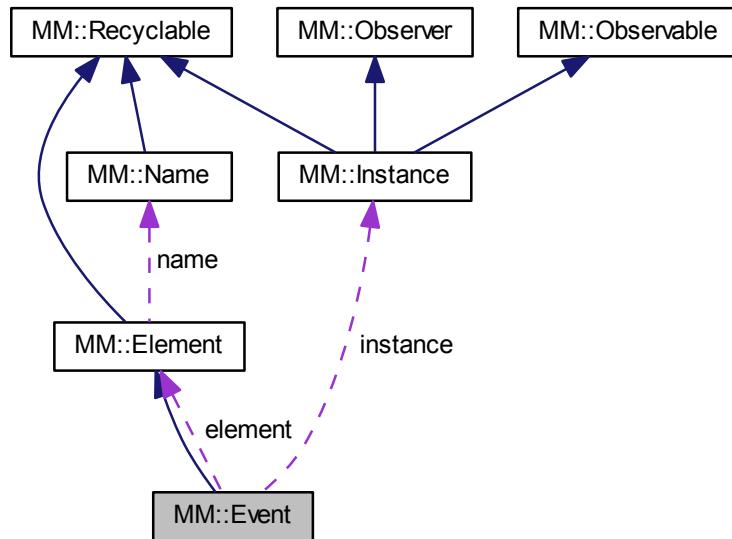
6.40 MM::Event Class Reference

```
#include <Event.h>
```

Inheritance diagram for MM::Event:



Collaboration diagram for MM::Event:



Public Member Functions

- virtual ~Event ()
- MM::VOID recycle (MM::Recycler *r)
- virtual MM::TID getTypeId ()
- virtual MM::BOOLEAN instanceof (MM::TID tid)
- MM::Element * getElement ()
- MM::VOID setElement (MM::Element *element)
- MM::Instance * getInstance ()
- MM::VOID setInstance (MM::Instance *instance)
- virtual MM::MESSAGE getMessage ()=0
- virtual MM::VOID toString (MM::String *buf)=0
- virtual MM::VOID toString (MM::String *buf, MM::UINT32 indent)=0

Protected Member Functions

- Event (MM::Name *name, MM::Instance *instance, MM::Element *element)

Protected Attributes

- MM::Element * element
- MM::Instance * instance

6.40.1 Constructor & Destructor Documentation

6.40.1.1 Event::Event (MM::Name * name, MM::Instance * instance, MM::Element * element) [protected]

instance in which the acting element lives

6.40.1.2 `Event::~Event() [virtual]`

6.40.2 Member Function Documentation

6.40.2.1 `MM::Element * Event::getElement()`

Here is the caller graph for this function:



6.40.2.2 `MM::Instance * Event::getInstance()`

Here is the caller graph for this function:



6.40.2.3 `virtual MM::MESSAGE MM::Event::getMessage() [pure virtual]`

Implemented in [MM::FlowEvent](#), [MM::Disablement](#), [MM::Enablement](#), [MM::Failure](#), [MM::TriggerEvent](#), [MM::Activation](#), [MM::Prevention](#), and [MM::Violation](#).

6.40.2.4 `MM::TID Event::getTypeId() [virtual]`

Reimplemented from [MM::Element](#).

Reimplemented in [MM::FlowEvent](#), [MM::Activation](#), [MM::Disablement](#), [MM::Enablement](#), [MM::Failure](#), [MM::TriggerEvent](#), [MM::Prevention](#), and [MM::Violation](#).

6.40.2.5 `MM::BOOLEAN Event::instanceof(MM::TID tid) [virtual]`

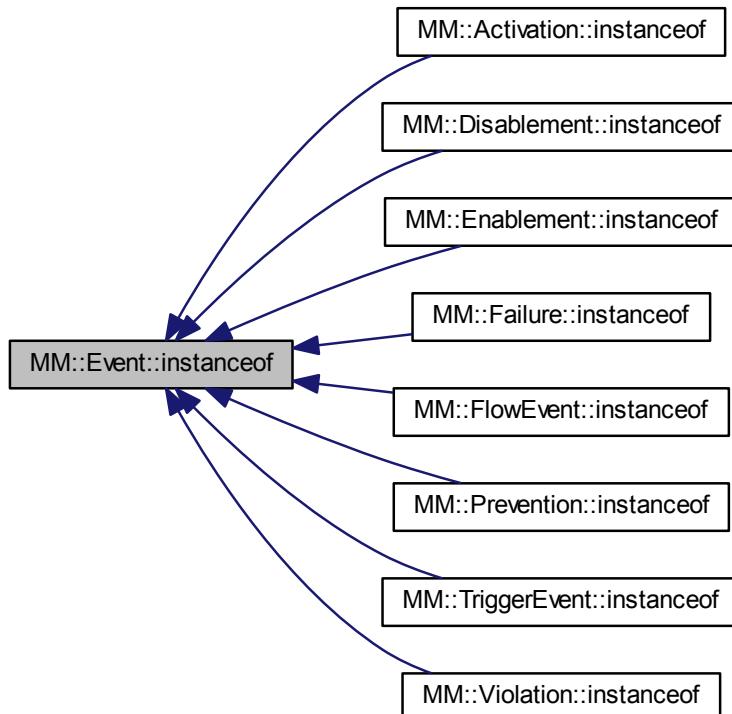
Reimplemented from [MM::Element](#).

Reimplemented in [MM::FlowEvent](#), [MM::Activation](#), [MM::Disablement](#), [MM::Enablement](#), [MM::Failure](#), [MM::TriggerEvent](#), [MM::Prevention](#), and [MM::Violation](#).

Here is the call graph for this function:



Here is the caller graph for this function:

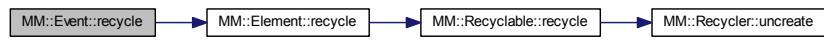


6.40.2.6 MM::VOID Event::recycle (MM::Recycler * r) [virtual]

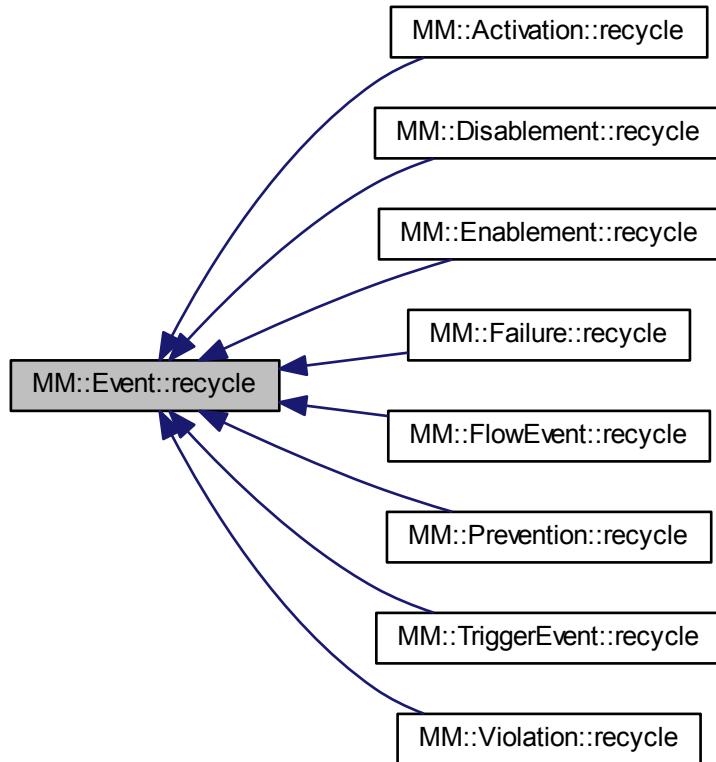
Reimplemented from [MM::Element](#).

Reimplemented in [MM::FlowEvent](#), [MM::Failure](#), [MM::TriggerEvent](#), [MM::Prevention](#), and [MM::Violation](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.40.2.7 **MM::VOID Event::setElement (*MM::Element * element*)**

6.40.2.8 **MM::VOID Event::setInstance (*MM::Instance * instance*)**

6.40.2.9 **virtual MM::VOID MM::Event::toString (*MM::String * buf*) [pure virtual]**

Implements [MM::Element](#).

Implemented in [MM::FlowEvent](#), [MM::Disablement](#), [MM::Enablement](#), [MM::Failure](#), [MM::TriggerEvent](#), [MM::Activation](#), [MM::Prevention](#), and [MM::Violation](#).

6.40.2.10 **virtual MM::VOID MM::Event::toString (*MM::String * buf*, *MM::UINT32 indent*) [pure virtual]**

Implements [MM::Element](#).

Implemented in [MM::FlowEvent](#), [MM::Disablement](#), [MM::Enablement](#), [MM::Failure](#), [MM::TriggerEvent](#), [MM::Activation](#), [MM::Prevention](#), and [MM::Violation](#).

6.40.3 Member Data Documentation

6.40.3.1 **MM::Element* MM::Event::element [protected]**

6.40.3.2 MM::Instance* MM::Event::instance [protected]

element causing the event

The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Event.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Event.cpp](#)

6.41 Event Class Reference

The [Event](#) abstraction is the abstract superclass of all transition elements.

```
#include <Event.h>
```

6.41.1 Detailed Description

The [Event](#) abstraction is the abstract superclass of all transition elements.

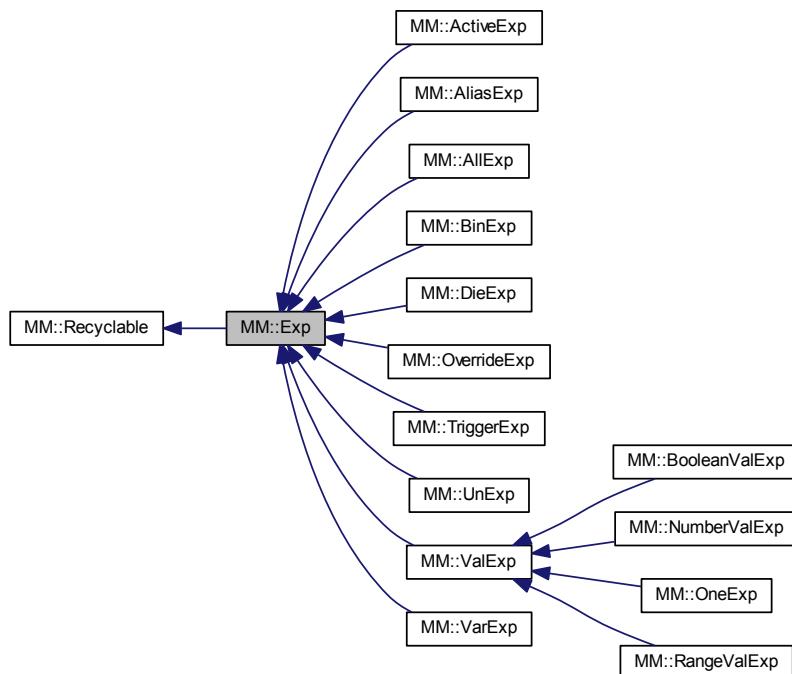
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Event.h](#)

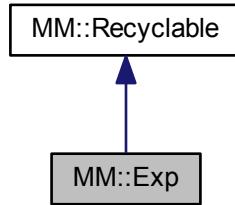
6.42 MM::Exp Class Reference

```
#include <Exp.h>
```

Inheritance diagram for MM::Exp:



Collaboration diagram for MM::Exp:



Public Member Functions

- virtual `~Exp ()=0`
- virtual `MM::VOID recycle (MM::Recycler *r)=0`
- virtual `MM::TID getTypeId ()`

Retrieves the type id of a `Exp` object.

- virtual `MM::BOOLEAN instanceof (MM::TID tid)=0`

Assesses if an object is an instance of a type tid.

- virtual `MM::VOID toString (MM::String *buf)=0`

Protected Member Functions

- `Exp ()`

6.42.1 Constructor & Destructor Documentation

6.42.1.1 `Exp::Exp () [protected]`

Constructs an `Exp` object.

Returns

`new Exp object`

Constructs a `TriggerExp` object.

Returns

`new TriggerExp object`

6.42.1.2 `Exp::~Exp () [pure virtual]`

Destructs an `Exp` object.

6.42.2 Member Function Documentation

6.42.2.1 MM::TID Exp::getTypeId() [virtual]

Retrieves the type id of a [Exp](#) object.

Retrieves the type id of a [TriggerExp](#) object.

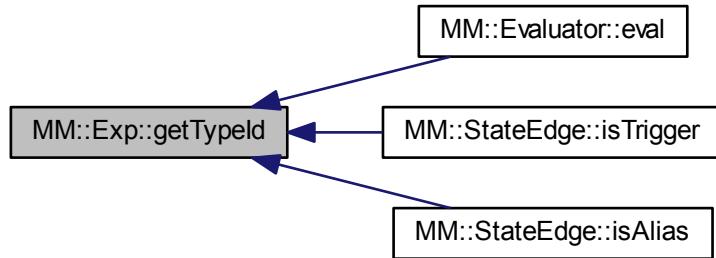
Returns

type id

Reimplemented from [MM::Recyclable](#).

Reimplemented in [MM::DieExp](#), [MM::RangeValExp](#), [MM::BinExp](#), [MM::UnExp](#), [MM::ActiveExp](#), [MM::OverrideExp](#), [MM::TriggerExp](#), [MM::OneExp](#), [MM::AllExp](#), [MM::NumberValExp](#), [MM::BooleanValExp](#), [MM::VarExp](#), [MM::ValExp](#), and [MM::AliasExp](#).

Here is the caller graph for this function:



6.42.2.2 MM::BOOLEAN Exp::instanceof(MM::TID *tid*) [virtual]

Assesses if an object is an instance of a type tid.

Parameters

<i>tid</i>	type id
------------	---------

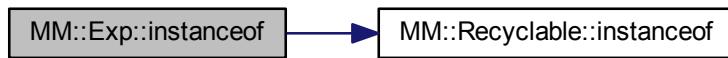
Returns

`MM_TRUE` if this object is instance of `tid`, `MM_FALSE` otherwise

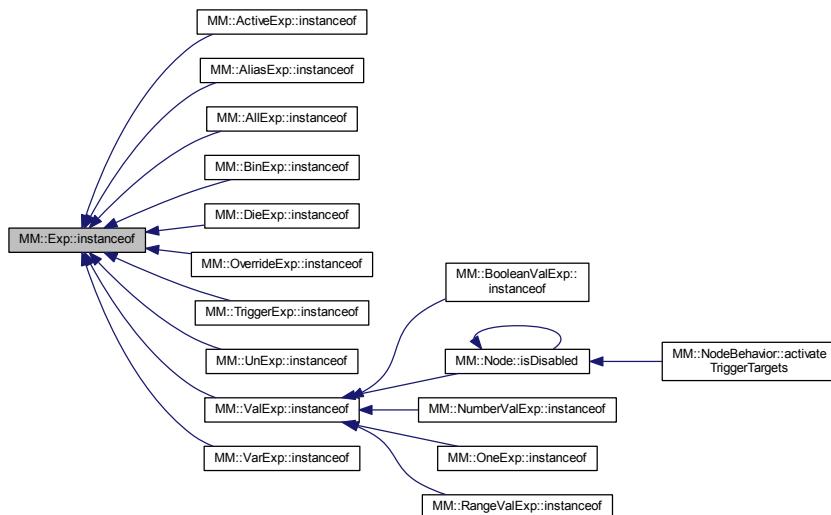
Reimplemented from [MM::Recyclable](#).

Reimplemented in [MM::DieExp](#), [MM::RangeValExp](#), [MM::BinExp](#), [MM::UnExp](#), [MM::ActiveExp](#), [MM::OverrideExp](#), [MM::TriggerExp](#), [MM::OneExp](#), [MM::AllExp](#), [MM::NumberValExp](#), [MM::BooleanValExp](#), [MM::VarExp](#), [MM::ValExp](#), and [MM::AliasExp](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.42.2.3 `MM::VOID Exp::recycle(MM::Recycler * r) [pure virtual]`

Recycles an [Exp](#) object in a [Recycler](#).

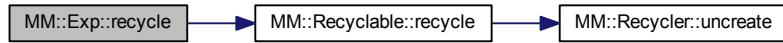
Parameters

<code>r</code>	Recycler object
----------------	---------------------------------

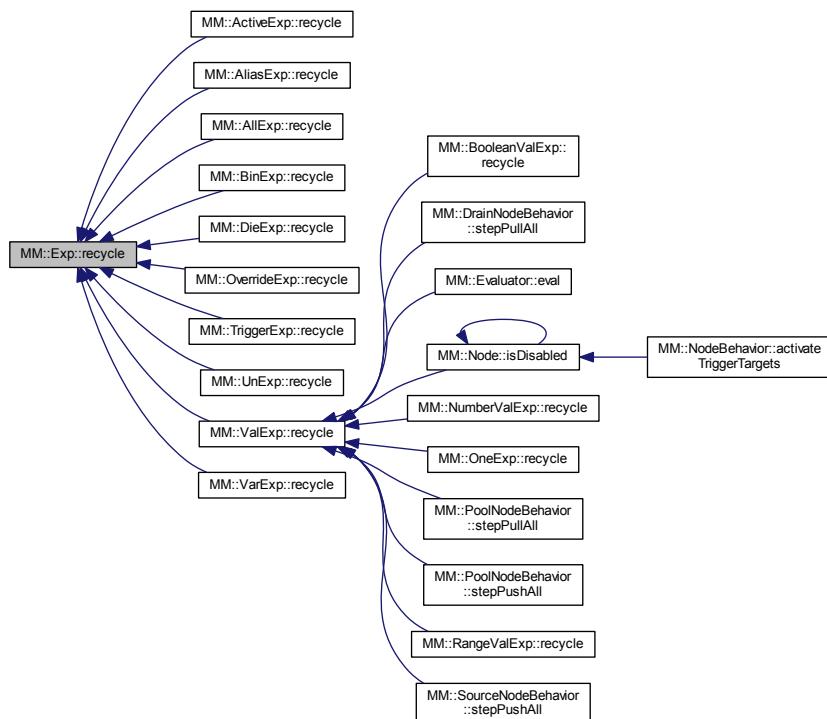
Reimplemented from [MM::Recyclable](#).

Implemented in [MM::DieExp](#), [MM::RangeValExp](#), [MM::BinExp](#), [MM::UnExp](#), [MM::ActiveExp](#), [MM::BooleanValExp](#), [MM::OverrideExp](#), [MM::TriggerExp](#), [MM::OneExp](#), [MM::AllExp](#), [MM::NumberValExp](#), [MM::VarExp](#), [MM::ValExp](#), and [MM::AliasExp](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.42.2.4 virtual MM::VOID MM::Exp::toString (MM::String * buf) [pure virtual]

Implements [MM::Recyclable](#).

Implemented in [MM::BinExp](#), [MM::RangeValExp](#), [MM::DieExp](#), [MM::UnExp](#), [MM::NumberValExp](#), [MM::ActiveExp](#), [MM::BooleanValExp](#), [MM::OverrideExp](#), [MM::OneExp](#), [MM::TriggerExp](#), [MM::AllExp](#), [MM::VarExp](#), [MM::ValExp](#), and [MM::AliasExp](#).

The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Exp.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Exp.cpp](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[TriggerExp.cpp](#)

6.43 Exp Class Reference

The [Exp](#) abstraction is the abstract super class of all expressions.

```
#include <Exp.h>
```

6.43.1 Detailed Description

The [Exp](#) abstraction is the abstract super class of all expressions.

Note

Expressions are visited by the [Evaluator](#).

The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Exp.h](#)

6.44 FailEvent Class Reference

The Failure abstraction defines a node instance failed during a step.

```
#include <Failure.h>
```

6.44.1 Detailed Description

The Failure abstraction defines a node instance failed during a step.

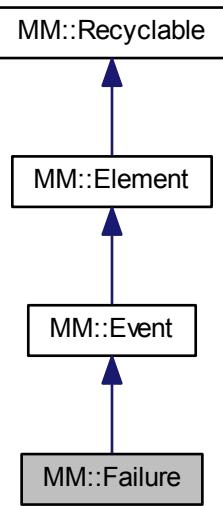
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Failure.h](#)

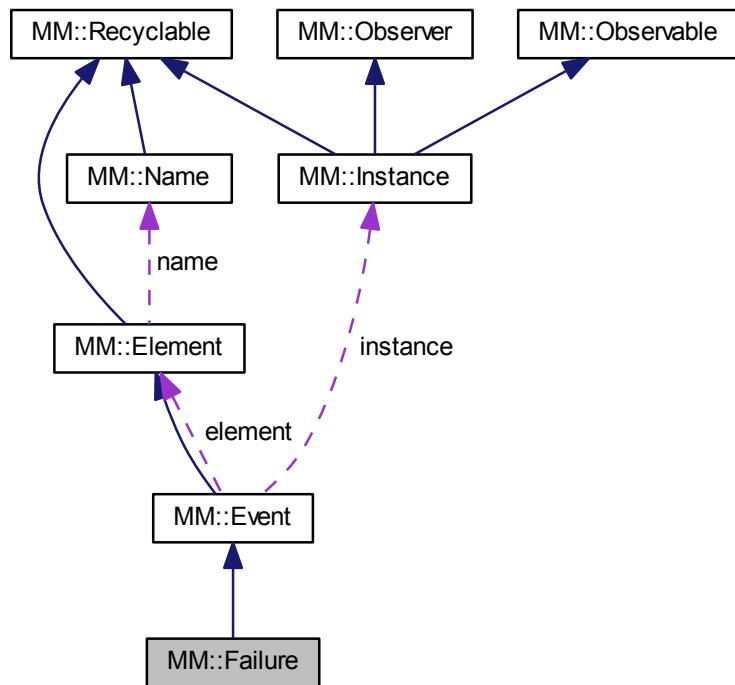
6.45 MM::Failure Class Reference

```
#include <Failure.h>
```

Inheritance diagram for MM::Failure:



Collaboration diagram for MM::Failure:



Public Member Functions

- [Failure \(MM::Instance *instance, MM::Node *node\)](#)
- [Failure \(MM::Location *loc, MM::Name *name\)](#)
- [~Failure \(\)](#)
- [MM::VOID recycle \(MM::Recycler *r\)](#)
- [MM::TID getTypeld \(\)](#)
- [MM::BOOLEAN instanceof \(MM::TID tid\)](#)
- [MM::Location * getLocation \(\)](#)
- [MM::MESSAGE getMessage \(\)](#)
- [MM::VOID toString \(MM::String *buf\)](#)
- [MM::VOID toString \(MM::String *buf, MM::UINT32 indent\)](#)

Additional Inherited Members

6.45.1 Constructor & Destructor Documentation

6.45.1.1 MM::Failure::Failure (MM::Instance * *instance*, MM::Node * *node*)

fail source location

6.45.1.2 MM::Failure::Failure (MM::Location * *loc*, MM::Name * *name*)

6.45.1.3 MM::Failure::~Failure ()

6.45.2 Member Function Documentation

6.45.2.1 MM::Location * MM::Failure::getLocation ()

6.45.2.2 MM::MESSAGE MM::Failure::getMessage () [virtual]

Implements [MM::Event](#).

6.45.2.3 MM::TID MM::Failure::getTypeld () [virtual]

Reimplemented from [MM::Event](#).

6.45.2.4 MM::BOOLEAN MM::Failure::instanceof (MM::TID *tid*) [virtual]

Reimplemented from [MM::Event](#).

Here is the call graph for this function:



6.45.2.5 MM::VOID MM::Failure::recycle (MM::Recycler * r) [virtual]

Reimplemented from [MM::Event](#).

Here is the call graph for this function:



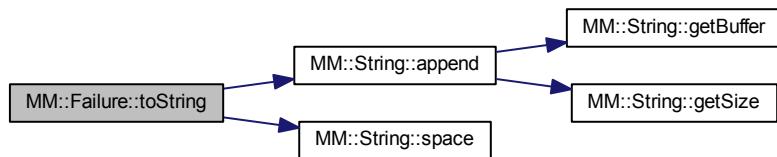
6.45.2.6 MM::VOID MM::Failure::toString (MM::String * buf) [virtual]

Implements [MM::Event](#).

6.45.2.7 MM::VOID MM::Failure::toString (MM::String * buf, MM::UINT32 indent) [virtual]

Implements [MM::Event](#).

Here is the call graph for this function:



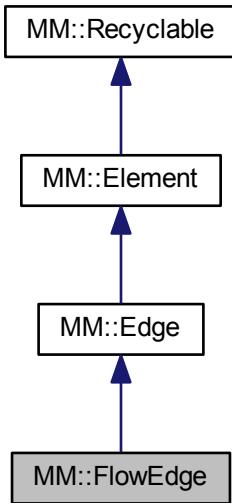
The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Failure.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Failure.cpp](#)

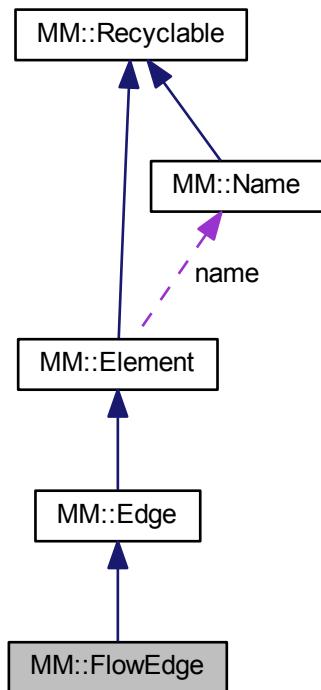
6.46 MM::FlowEdge Class Reference

```
#include <FlowEdge.h>
```

Inheritance diagram for MM::FlowEdge:



Collaboration diagram for MM::FlowEdge:



Public Member Functions

- `FlowEdge (MM::Name *name, MM::Name *src, MM::Exp *exp, MM::Name *tgt)`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypId ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::VOID toString (MM::String *buf)`
- `MM::VOID toString (MM::String *buf, MM::UINT32 indent)`

Protected Member Functions

- `~FlowEdge ()`

Additional Inherited Members

6.46.1 Constructor & Destructor Documentation

6.46.1.1 FlowEdge::~FlowEdge () [protected]

end arrow keyword length

6.46.1.2 FlowEdge::FlowEdge (MM::Name * name, MM::Name * src, MM::Exp * exp, MM::Name * tgt)

6.46.2 Member Function Documentation

6.46.2.1 MM::TID FlowEdge::getTypId () [virtual]

Reimplemented from [MM::Edge](#).

6.46.2.2 MM::BOOLEAN FlowEdge::instanceof (MM::TID tid) [virtual]

Reimplemented from [MM::Edge](#).

Here is the call graph for this function:



6.46.2.3 MM::VOID FlowEdge::recycle (MM::Recycler * r) [virtual]

Reimplemented from [MM::Edge](#).

Here is the call graph for this function:



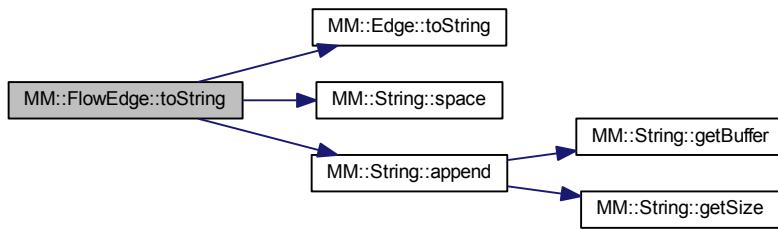
6.46.2.4 MM::VOID FlowEdge::toString (MM::String * buf) [virtual]

Implements [MM::Edge](#).

6.46.2.5 MM::VOID FlowEdge::toString (MM::String * buf, MM::UINT32 indent) [virtual]

Reimplemented from [MM::Edge](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[FlowEdge.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[FlowEdge.cpp](#)

6.47 FlowEdge Class Reference

The [FlowEdge](#) abstraction is a program element that defines resource connections, where the amount of resources that may or must flow between source and target node instances is defined by its expression.

```
#include <FlowEdge.h>
```

6.47.1 Detailed Description

The [FlowEdge](#) abstraction is a program element that defines resource connections, where the amount of resources that may or must flow between source and target node instances is defined by its expression.

The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[FlowEdge.h](#)

6.48 FlowEvent Class Reference

The [FlowEvent](#) abstraction defines the amount of resources that flows between source and target node instances during a step.

```
#include <FlowEvent.h>
```

6.48.1 Detailed Description

The [FlowEvent](#) abstraction defines the amount of resources that flows between source and target node instances during a step.

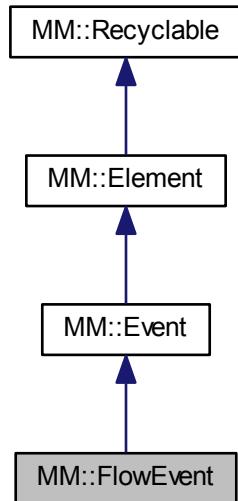
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[FlowEvent.h](#)

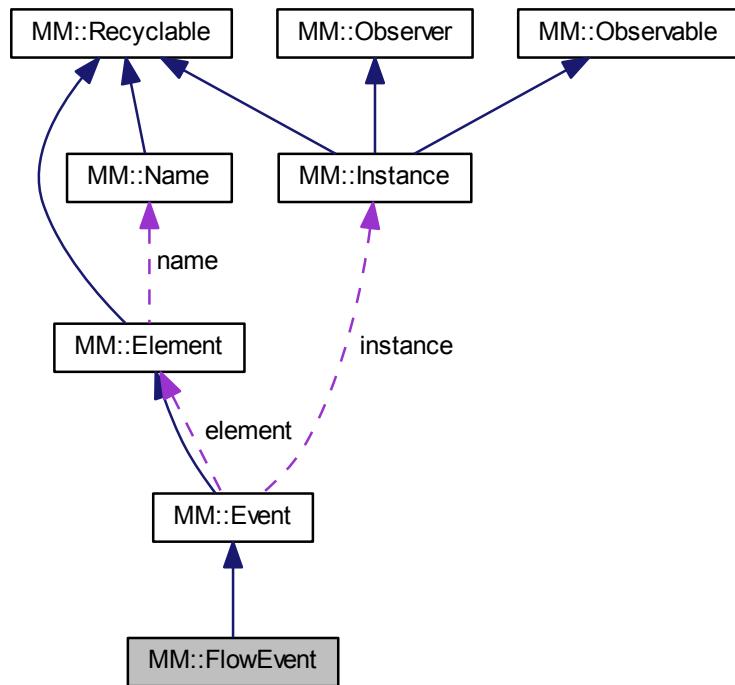
6.49 MM::FlowEvent Class Reference

```
#include <FlowEvent.h>
```

Inheritance diagram for MM::FlowEvent:



Collaboration diagram for MM::FlowEvent:



Public Member Functions

- `FlowEvent (MM::Instance *actInstance, MM::Node *actNode, MM::Edge *actEdge, MM::Instance *srcInstance, MM::Node *srcNode, MM::UINT32 amount, MM::Instance *tgtInstance, MM::Node *tgtNode)`
- `~FlowEvent ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::Node * getSourceNode ()`
- `MM::UINT32 getAmount ()`
- `MM::Node * getTargetNode ()`
- `MM::Instance * getActInstance ()`
- `MM::Node * getActNode ()`
- `MM::Edge * getActEdge ()`
- `MM::Instance * getSourceInstance ()`
- `MM::Instance * getTargetInstance ()`
- `MM::VOID setSourceInstance (MM::Instance *instance)`
- `MM::VOID setTargetInstance (MM::Instance *instance)`
- `MM::MESSAGE getMessage ()`
- `MM::VOID toString (MM::String *buf)`
- `MM::VOID toString (MM::String *buf, MM::UINT32 indent)`

Additional Inherited Members

6.49.1 Constructor & Destructor Documentation

6.49.1.1 `FlowEvent::FlowEvent (MM::Instance * actInstance, MM::Node * actNode, MM::Edge * actEdge, MM::Instance * srcInstance, MM::Node * srcNode, MM::UINT32 amount, MM::Instance * tgtInstance, MM::Node * tgtNode)`

amount of resources that flowed from source to target

6.49.1.2 `FlowEvent::~FlowEvent ()`

6.49.2 Member Function Documentation

6.49.2.1 `MM::Edge * FlowEvent::getActEdge ()`

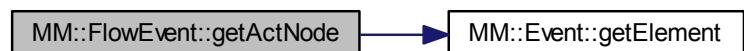
6.49.2.2 `MM::Instance * FlowEvent::getActInstance ()`

Here is the call graph for this function:



6.49.2.3 `MM::Node * FlowEvent::getActNode ()`

Here is the call graph for this function:



6.49.2.4 `MM::UINT32 FlowEvent::getAmount ()`

6.49.2.5 `MM::MESSAGE FlowEvent::getMessage () [virtual]`

Implements [MM::Event](#).

6.49.2.6 `MM::Instance * FlowEvent::getSourceInstance ()`

6.49.2.7 **MM::Node *** FlowEvent::getSourceNode ()

6.49.2.8 **MM::Instance *** FlowEvent::getTargetInstance ()

6.49.2.9 **MM::Node *** FlowEvent::getTargetNode ()

6.49.2.10 **MM::TID** FlowEvent::getTypeId () [virtual]

Reimplemented from [MM::Event](#).

6.49.2.11 **MM::BOOLEAN** FlowEvent::instanceof (**MM::TID** *tid*) [virtual]

Reimplemented from [MM::Event](#).

Here is the call graph for this function:



6.49.2.12 **MM::VOID** FlowEvent::recycle (**MM::Recycler** * *r*) [virtual]

Reimplemented from [MM::Event](#).

Here is the call graph for this function:



6.49.2.13 **MM::VOID** MM::FlowEvent::setSourceInstance (**MM::Instance** * *instance*)

6.49.2.14 **MM::VOID** MM::FlowEvent::setTargetInstance (**MM::Instance** * *instance*)

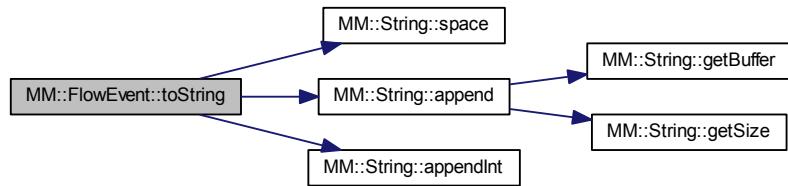
6.49.2.15 **MM::VOID** FlowEvent::toString (**MM::String** * *buf*) [virtual]

Implements [MM::Event](#).

6.49.2.16 **MM::VOID** FlowEvent::toString (**MM::String** * *buf*, **MM::UINT32** *indent*) [virtual]

Implements [MM::Event](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[FlowEvent.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[FlowEvent.cpp](#)

6.50 GateNodeBehavior Class Reference

The [GateNodeBehavior](#) abstraction defines the behavior of gate nodes.

```
#include <GateNodeBehavior.h>
```

6.50.1 Detailed Description

The [GateNodeBehavior](#) abstraction defines the behavior of gate nodes.

Note

Strategy

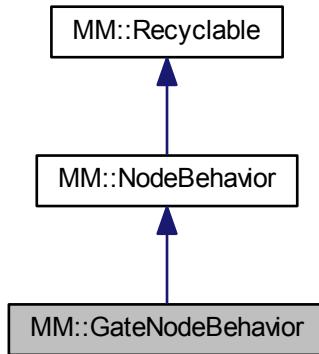
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[GateNodeBehavior.h](#)

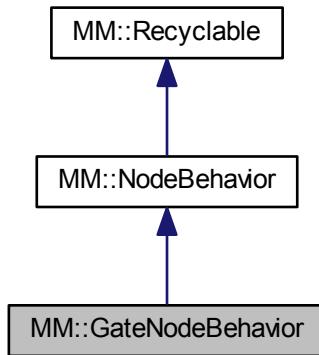
6.51 MM::GateNodeBehavior Class Reference

```
#include <GateNodeBehavior.h>
```

Inheritance diagram for MM::GateNodeBehavior:



Collaboration diagram for MM::GateNodeBehavior:



Public Member Functions

- `GateNodeBehavior (MM::NodeBehavior::IO io, MM::NodeBehavior::When when, MM::NodeBehavior::Act act, MM::NodeBehavior::How how)`
- `~GateNodeBehavior ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::UINT32 getCreateMessage ()`
- `MM::UINT32 getUpdateMessage ()`
- `MM::UINT32 getDeleteMessage ()`
- `MM::VOID stepPullAll (MM::Node *node, MM::Instance *i, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)`

- MM::VOID stepPushAll (MM::Node *node, MM::Instance *i, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)
- MM::VOID begin (MM::Instance *i, MM::Machine *m, MM::Node *n)
- MM::VOID end (MM::Instance *i, MM::Machine *m, MM::Node *n)
- MM::VOID change (MM::Instance *i, MM::Machine *m, MM::Node *n)
- MM::VOID add (MM::Instance *i, MM::Machine *m, MM::Node *n, MM::UINT32 amount)
- MM::VOID sub (MM::Instance *i, MM::Machine *m, MM::Node *n, MM::UINT32 amount)
- MM::UINT32 getCapacity (MM::Instance *i, MM::Node *n)
- MM::UINT32 getResources (MM::Instance *i, MM::Node *n)
- MM::BOOLEAN hasCapacity (MM::Instance *i, MM::Node *n, MM::UINT32 amount)
- MM::BOOLEAN hasResources (MM::Instance *i, MM::Node *n, MM::UINT32 amount)
- MM::VOID toString (MM::String *buf)
- MM::VOID toString (MM::String *buf, MM::Name *name)

Additional Inherited Members

6.51.1 Constructor & Destructor Documentation

6.51.1.1 GateNodeBehavior::GateNodeBehavior (MM::NodeBehavior::IO *io*, MM::NodeBehavior::When *when*, MM::NodeBehavior::Act *act*, MM::NodeBehavior::How *how*)

gate keyword length

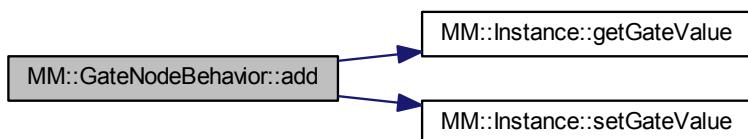
6.51.1.2 GateNodeBehavior::~GateNodeBehavior ()

6.51.2 Member Function Documentation

6.51.2.1 MM::VOID GateNodeBehavior::add (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*, MM::UINT32 *amount*) [virtual]

Implements [MM::NodeBehavior](#).

Here is the call graph for this function:



6.51.2.2 MM::VOID GateNodeBehavior::begin (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*) [virtual]

Implements [MM::NodeBehavior](#).

6.51.2.3 **MM::VOID** `GateNodeBehavior::change (MM::Instance * i, MM::Machine * m, MM::Node * n)`
[virtual]

Implements [MM::NodeBehavior](#).

6.51.2.4 **MM::VOID** `GateNodeBehavior::end (MM::Instance * i, MM::Machine * m, MM::Node * n)` [virtual]

Implements [MM::NodeBehavior](#).

6.51.2.5 **MM::UINT32** `GateNodeBehavior::getCapacity (MM::Instance * i, MM::Node * n)` [virtual]

Implements [MM::NodeBehavior](#).

6.51.2.6 **MM::UINT32** `GateNodeBehavior::getCreateMessage ()` [virtual]

Implements [MM::NodeBehavior](#).

6.51.2.7 **MM::UINT32** `GateNodeBehavior::getDeleteMessage ()` [virtual]

Implements [MM::NodeBehavior](#).

6.51.2.8 **MM::UINT32** `GateNodeBehavior::getResources (MM::Instance * i, MM::Node * n)` [virtual]

Implements [MM::NodeBehavior](#).

6.51.2.9 **MM::TID** `GateNodeBehavior::getTypeld ()` [virtual]

Reimplemented from [MM::NodeBehavior](#).

6.51.2.10 **MM::UINT32** `GateNodeBehavior::getUpdateMessage ()` [virtual]

Implements [MM::NodeBehavior](#).

6.51.2.11 **MM::BOOLEAN** `GateNodeBehavior::hasCapacity (MM::Instance * i, MM::Node * n, MM::UINT32 amount)` [virtual]

Implements [MM::NodeBehavior](#).

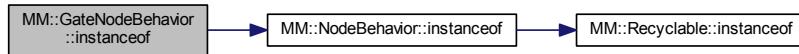
6.51.2.12 **MM::BOOLEAN** `GateNodeBehavior::hasResources (MM::Instance * i, MM::Node * n, MM::UINT32 amount)` [virtual]

Implements [MM::NodeBehavior](#).

6.51.2.13 **MM::BOOLEAN** `GateNodeBehavior::instanceof (MM::TID tid)` [virtual]

Reimplemented from [MM::NodeBehavior](#).

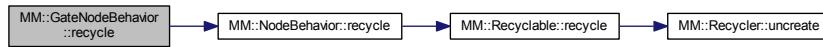
Here is the call graph for this function:



6.51.2.14 MM::VOID GateNodeBehavior::recycle (MM::Recycler * r) [virtual]

Reimplemented from [MM::Recyclable](#).

Here is the call graph for this function:



6.51.2.15 MM::VOID GateNodeBehavior::stepPullAll (MM::Node * node, MM::Instance * i, MM::Vector< MM::NodeWorkItem * > * work, MM::Machine * m, MM::Transition * tr) [virtual]

Implements [MM::NodeBehavior](#).

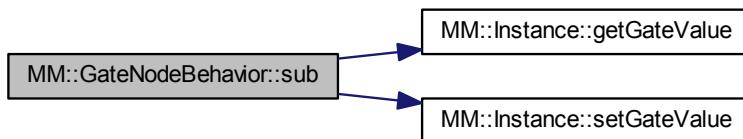
6.51.2.16 MM::VOID GateNodeBehavior::stepPushAll (MM::Node * node, MM::Instance * i, MM::Vector< MM::NodeWorkItem * > * work, MM::Machine * m, MM::Transition * tr) [virtual]

Implements [MM::NodeBehavior](#).

6.51.2.17 MM::VOID GateNodeBehavior::sub (MM::Instance * i, MM::Machine * m, MM::Node * n, MM::UINT32 amount) [virtual]

Implements [MM::NodeBehavior](#).

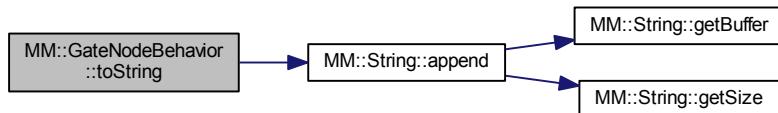
Here is the call graph for this function:



6.51.2.18 MM::VOID GateNodeBehavior::toString (MM::String * buf) [virtual]

Implements [MM::Recyclable](#).

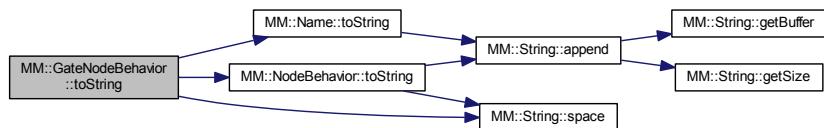
Here is the call graph for this function:



6.51.2.19 MM::VOID GateNodeBehavior::toString (MM::String * buf, MM::Name * name) [virtual]

Reimplemented from [MM::NodeBehavior](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[GateNodeBehavior.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[GateNodeBehavior.cpp](#)

6.52 Instance Class Reference

The instance abstraction defines instance data that is defined by its definition, and is manipulated by the evaluator during steps.

```
#include <Instance.h>
```

6.52.1 Detailed Description

The instance abstraction defines instance data that is defined by its definition, and is manipulated by the evaluator during steps.

Note

[Observable](#) by components external to [MM Lib](#).

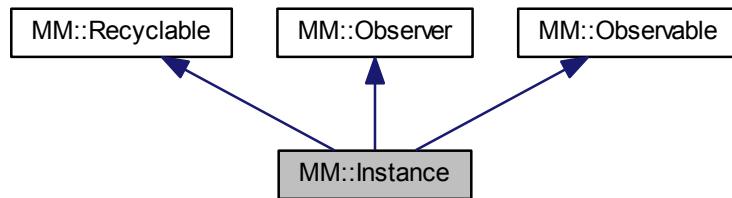
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Instance.h](#)

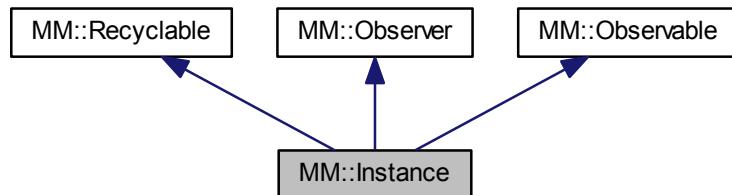
6.53 MM::Instance Class Reference

```
#include <Instance.h>
```

Inheritance diagram for MM::Instance:



Collaboration diagram for MM::Instance:



Public Member Functions

- `Instance (MM::Instance *parent, MM::Definition *def, MM::Element *decl)`
Instantiates an `Instance`.
- `~Instance ()`
Destructs an `Instance` object.
- `MM::VOID recycle (MM::Recycler *r)`
Recycles an `Instance` object in a `Recycler`.
- `MM::TID getTypeId ()`
Retrieves the type id of an `Instance` object.
- `MM::BOOLEAN instanceof (MM::TID tid)`
Assesses if an object is an instance of a type tid.
- `MM::Definition * getDefinition ()`
Retrieves the definition of an `Instance` object.
- `MM::Instance * getParent ()`
- `MM::Element * getDeclaration ()`
- `MM::VOID mark ()`
- `MM::BOOLEAN isMarked ()`

- MM::VOID sweep (MM::Machine *m)
- MM::Map< MM::Element *, MM::Vector< MM::Instance * > * > * getInstances ()
- MM::Vector< MM::Instance * > * getInstances (MM::Element *element)
- MM::Instance * getInstance (MM::Declaration *decl)

Retrieves an [Instance](#) object for a node.
- MM::INT32 getIndex (MM::Element *element, MM::Instance *i)
- MM::UINT32 getValue (MM::Node *node)
- MM::UINT32 getNewValue (MM::Node *node)
- MM::UINT32 getOldValue (MM::Node *node)
- MM::UINT32 getGateValue (MM::Node *node)
- MM::VOID deleteValue (MM::Node *node)
- MM::VOID setValue (MM::Node *node, MM::UINT32 value)
- MM::VOID setNewValue (MM::Node *node, MM::UINT32 value)
- MM::VOID setOldValue (MM::Node *node, MM::UINT32 value)
- MM::VOID setGateValue (MM::Node *node, MM::UINT32 value)
- MM::BOOLEAN isEvaluatedExp (MM::Exp *exp)
- MM::VOID setEvaluatedExp (MM::Exp *exp, MM::INT32 val)
- MM::INT32 getEvaluatedExp (MM::Exp *exp)
- MM::BOOLEAN isActive (MM::Node *node)
- MM::VOID setActive (MM::Node *node)
- MM::VOID setNextActive (MM::Node *node)
- MM::BOOLEAN isDisabled (MM::Node *node)
- MM::VOID setDisabled (MM::Node *node)
- MM::VOID update (MM::Observable *observable, MM::VOID *aux, MM::UINT32 message, MM::VOID *object)

Updates an [Observer](#).
- MM::VOID createInstances (MM::Element *element, MM::Machine *m, MM::Definition *unitDef, MM::UINT32 amount)
- MM::VOID destroyInstances (MM::Element *element, MM::Machine *m, MM::UINT32 amount)
- MM::VOID destroyAllInstances (MM::Element *element, MM::Machine *m)
- MM::VOID destroyInstance (MM::Element *element, MM::Machine *m, MM::Instance *i)
- MM::VOID begin ()
- MM::VOID finalize ()

Commits the new values and purges the old values. finalize step.
- MM::VOID clearActive ()
- MM::VOID clearDisabled ()
- MM::UINT32 getCapacity (MM::Node *node)
- MM::UINT32 getResources (MM::Node *node)
- MM::BOOLEAN hasResources (MM::Node *node, MM::UINT32 amount)
- MM::BOOLEAN hasCapacity (MM::Node *node, MM::UINT32 amount)
- MM::VOID sub (MM::Node *node, MM::Machine *m, MM::UINT32 amount)
- MM::VOID add (MM::Node *node, MM::Machine *m, MM::UINT32 amount)
- MM::VOID nameToString (MM::String *buf)
- MM::VOID nameToString (MM::Element *element, MM::String *buf)
- MM::VOID toString (MM::String *buf)
- MM::VOID toString (MM::String *buf, MM::UINT32 indent)

6.53.1 Constructor & Destructor Documentation

6.53.1.1 Instance::Instance (MM::Instance * parent, MM::Definition * def, MM::Element * decl)

Instantiates an [Instance](#).

Parameters

<code>type</code>	Definition that defines the instance
-------------------	--

Returns

new [Instance](#) object

6.53.1.2 Instance::~Instance()

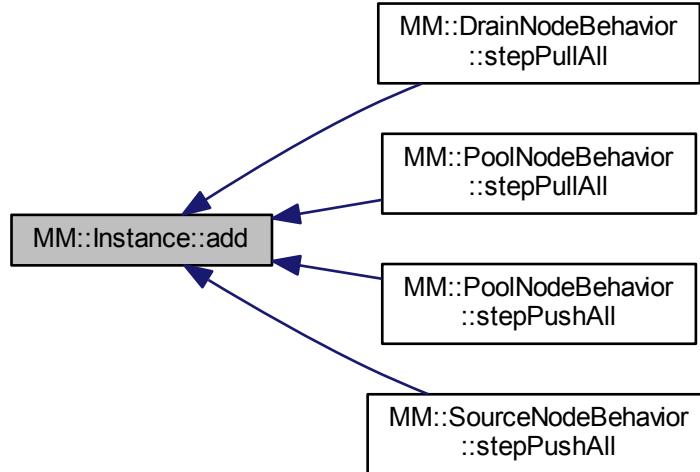
Destructs an [Instance](#) object.

6.53.2 Member Function Documentation**6.53.2.1 MM::VOID Instance::add (MM::Node * node, MM::Machine * m, MM::UINT32 amount)**

Here is the call graph for this function:



Here is the caller graph for this function:



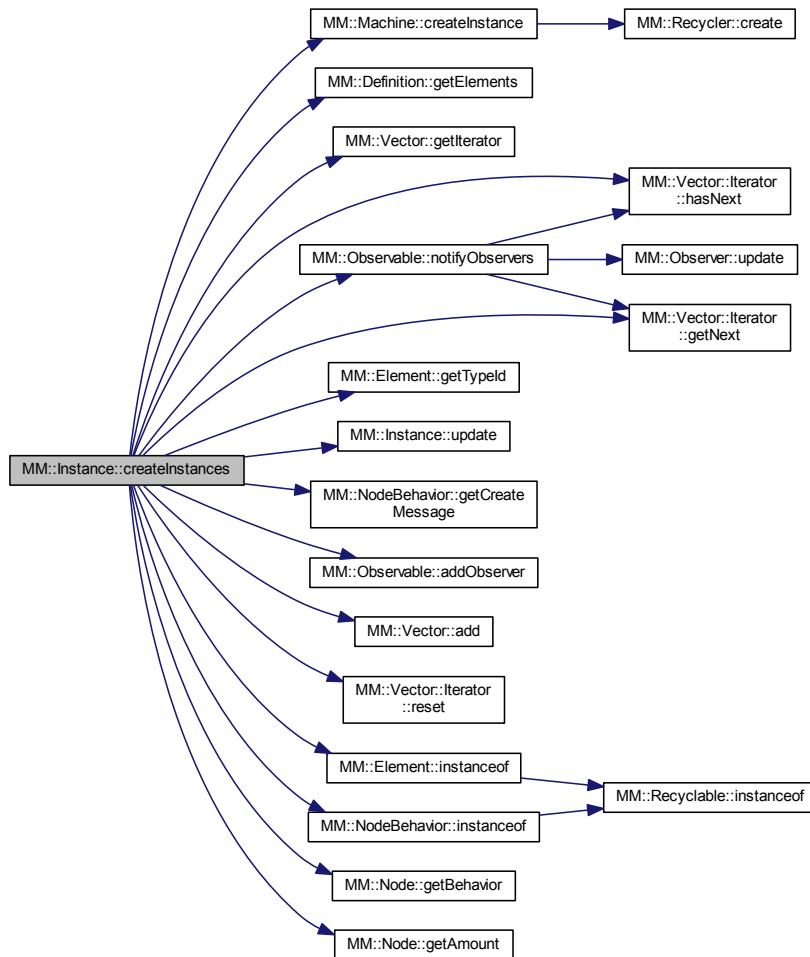
6.53.2.2 **MM::VOID Instance::begin ()**

6.53.2.3 **MM::VOID MM::Instance::clearActive ()**

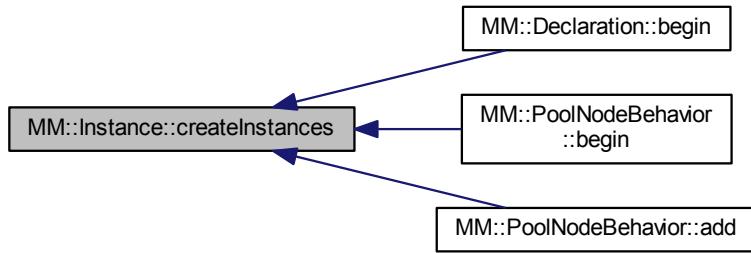
6.53.2.4 **MM::VOID MM::Instance::clearDisabled ()**

6.53.2.5 **MM::VOID Instance::createInstances (MM::Element * element, MM::Machine * m, MM::Definition * unitDef, MM::UINT32 amount)**

Here is the call graph for this function:



Here is the caller graph for this function:



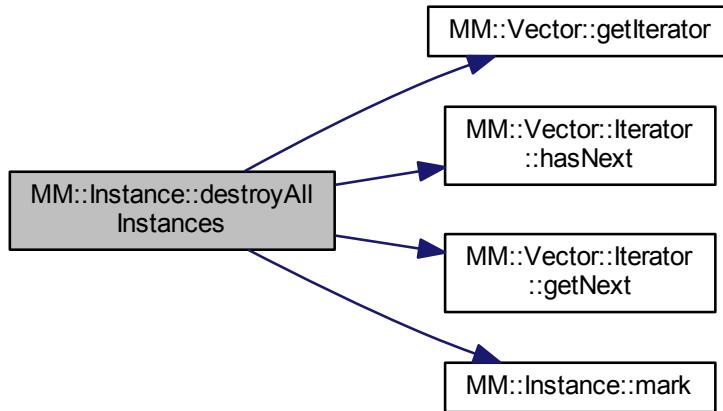
6.53.2.6 MM::VOID Instance::deleteValue (`MM::Node * node`)

Here is the caller graph for this function:

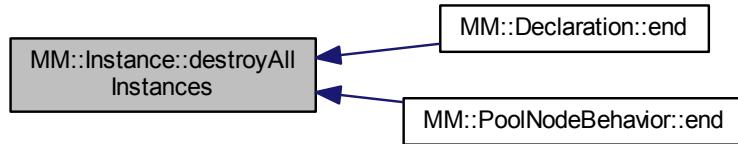


6.53.2.7 MM::VOID Instance::destroyAllInstances (`MM::Element * element`, `MM::Machine * m`)

Here is the call graph for this function:



Here is the caller graph for this function:



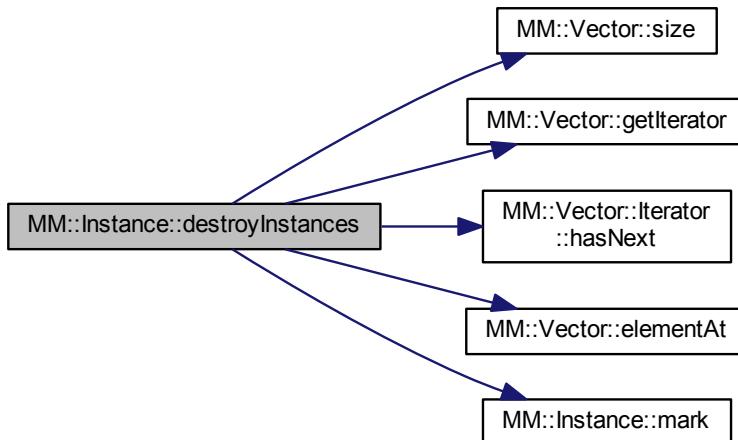
6.53.2.8 MM::VOID Instance::destroyInstance (MM::Element * element, MM::Machine * m, MM::Instance * i)

Here is the call graph for this function:

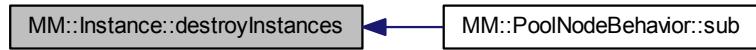


6.53.2.9 MM::VOID Instance::destroyInstances (MM::Element * element, MM::Machine * m, MM::UINT32 amount)

Here is the call graph for this function:



Here is the caller graph for this function:



6.53.2.10 MM::VOID Instance::finalize()

Commits the new values and purges the old values. finalize step.

Here is the call graph for this function:

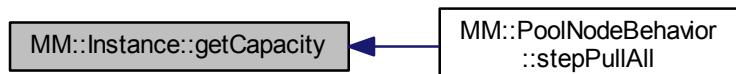


6.53.2.11 MM::UINT32 Instance::getCapacity(MM::Node * node)

Here is the call graph for this function:

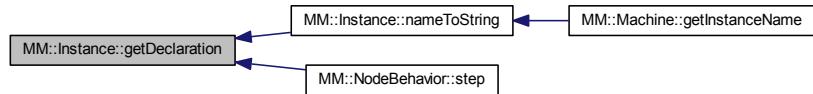


Here is the caller graph for this function:



6.53.2.12 MM::Element * Instance::getDeclaration()

Here is the caller graph for this function:



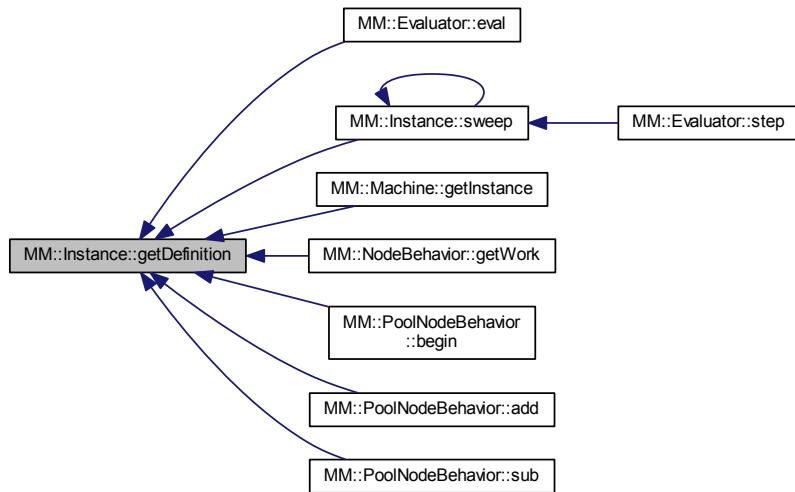
6.53.2.13 MM::Definition * Instance::getDefinition()

Retrieves the definition of an [Instance](#) object.

Returns

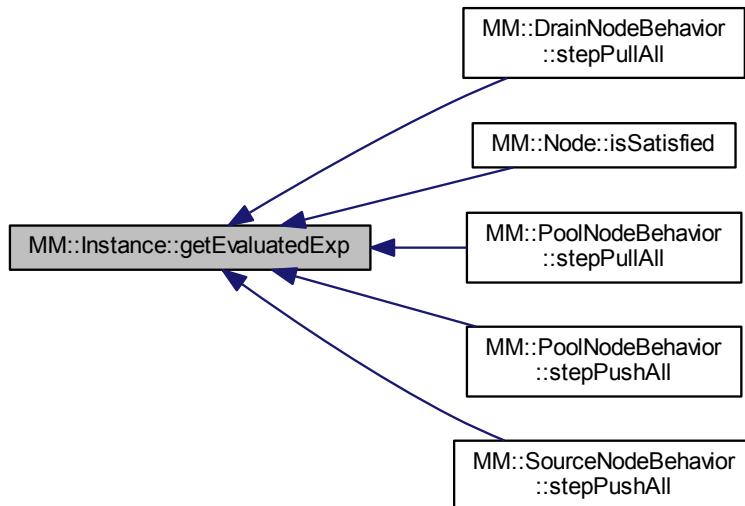
[Definition](#) object

Here is the caller graph for this function:



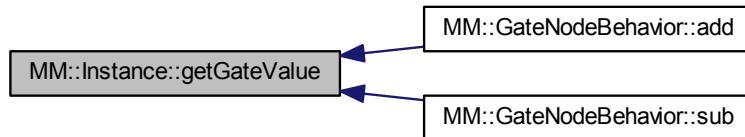
6.53.2.14 MM::INT32 Instance::getEvaluatedExp (MM::Exp * exp)

Here is the caller graph for this function:



6.53.2.15 MM::UINT32 Instance::getGateValue (MM::Node * node)

Here is the caller graph for this function:

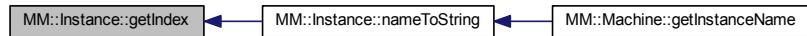


6.53.2.16 MM::INT32 Instance::getIndex (MM::Element * element, MM::Instance * i)

Here is the call graph for this function:



Here is the caller graph for this function:



6.53.2.17 MM::Instance * Instance::getInstance (MM::Declaration * decl)

Retrieves an [Instance](#) object for a node.

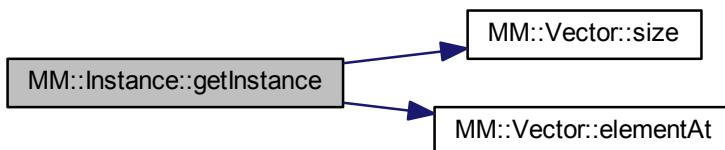
Parameters

<i>decl</i>	declaration id
-------------	----------------

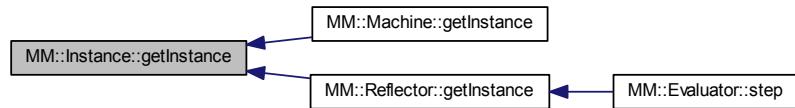
Returns

[Instance](#) object

Here is the call graph for this function:



Here is the caller graph for this function:



6.53.2.18 `MM::Map< MM::Element *, MM::Vector< MM::Instance * > * > * Instance::getInstances()`

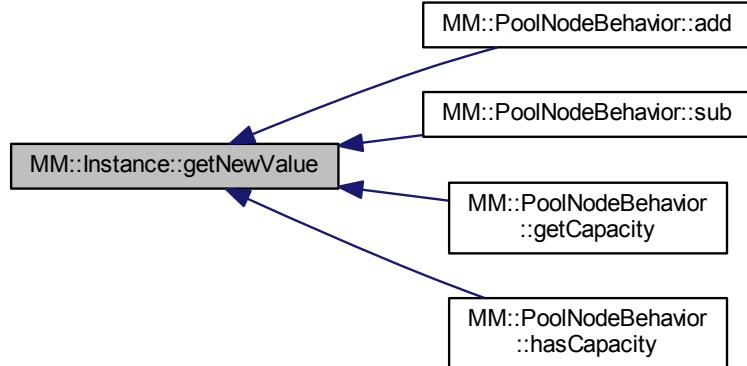
Here is the caller graph for this function:



6.53.2.19 `MM::Vector< MM::Instance * > * Instance::getInstances(MM::Element * element)`

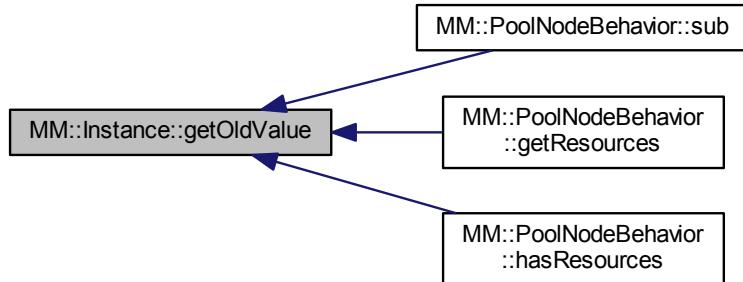
6.53.2.20 `MM::UINT32 Instance::getNewValue(MM::Node * node)`

Here is the caller graph for this function:



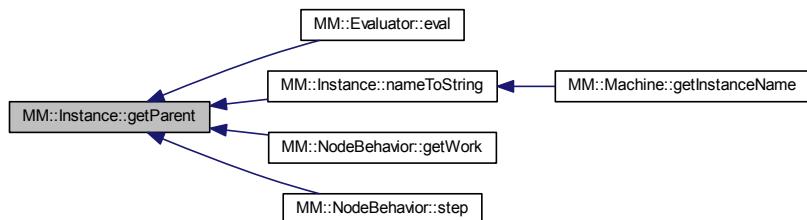
6.53.2.21 MM::UINT32 Instance::getOldValue (MM::Node * node)

Here is the caller graph for this function:



6.53.2.22 MM::Instance * Instance::getParent ()

Here is the caller graph for this function:



6.53.2.23 MM::UINT32 Instance::getResources (MM::Node * node)

Here is the call graph for this function:



6.53.2.24 MM::TID Instance::getTypeId() [virtual]

Retrieves the type id of an [Instance](#) object.

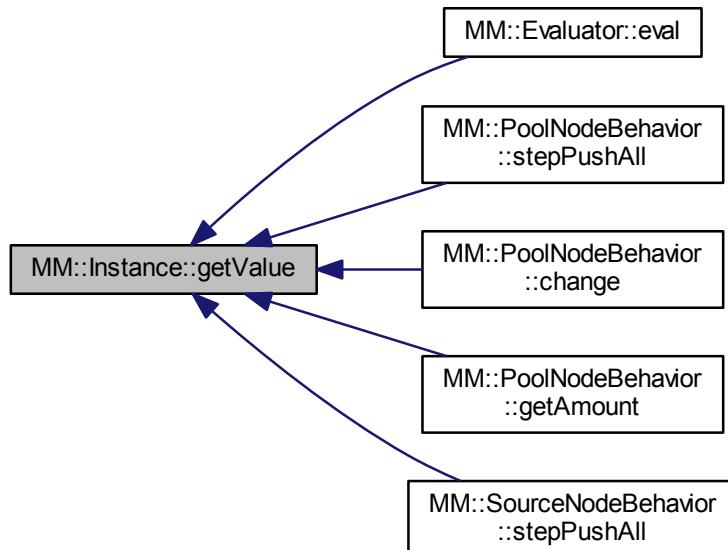
Returns

type id

Reimplemented from [MM::Observable](#).

6.53.2.25 MM::UINT32 Instance::getValue(MM::Node * node)

Here is the caller graph for this function:

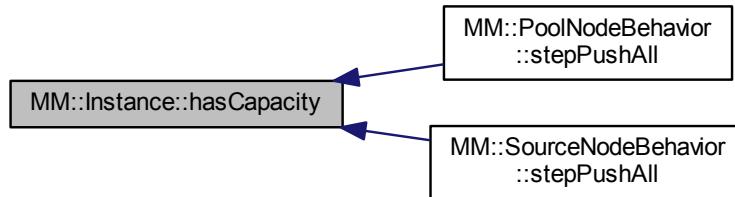


6.53.2.26 MM::BOOLEAN Instance::hasCapacity(MM::Node * node, MM::UINT32 amount)

Here is the call graph for this function:



Here is the caller graph for this function:

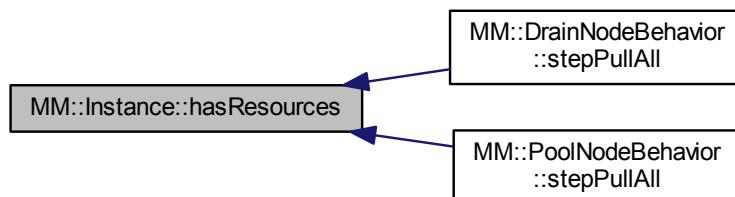


6.53.2.27 MM::BOOLEAN Instance::hasResources (MM::Node * node, MM::UINT32 amount)

Here is the call graph for this function:



Here is the caller graph for this function:



6.53.2.28 MM::BOOLEAN Instance::instanceof (MM::TID tid) [virtual]

Assesses if an object is an instance of a type tid.

Parameters

<i>tid</i>	type id
------------	---------

Returns

MM_TRUE if this object is instance of tid, MM_FALSE otherwise

Reimplemented from [MM::Observable](#).

Here is the call graph for this function:

**6.53.2.29 MM::BOOLEAN Instance::isActive (MM::Node * node)**

Here is the caller graph for this function:

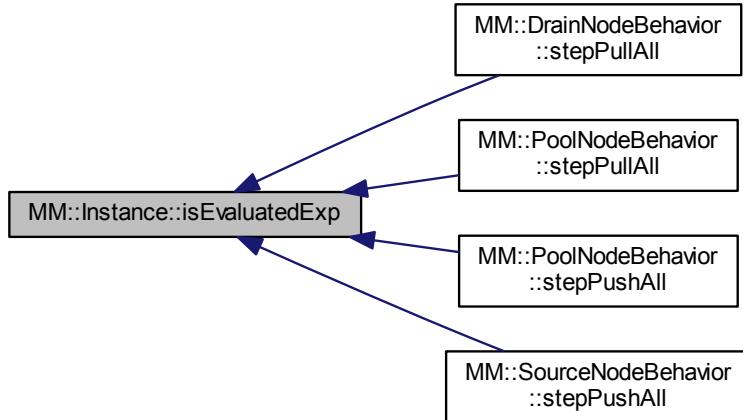
**6.53.2.30 MM::BOOLEAN Instance::isEnabled (MM::Node * node)**

Here is the caller graph for this function:



6.53.2.31 MM::BOOLEAN Instance::isEvaluatedExp (MM::Exp * exp)

Here is the caller graph for this function:



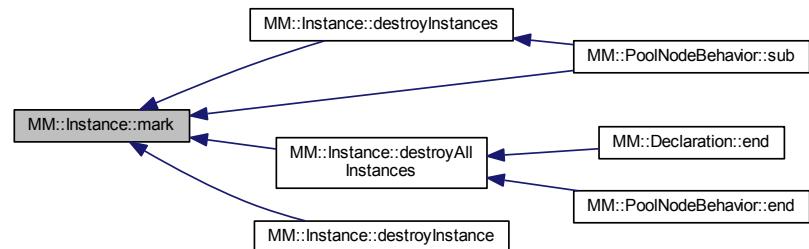
6.53.2.32 MM::BOOLEAN Instance::isMarked ()

Here is the caller graph for this function:



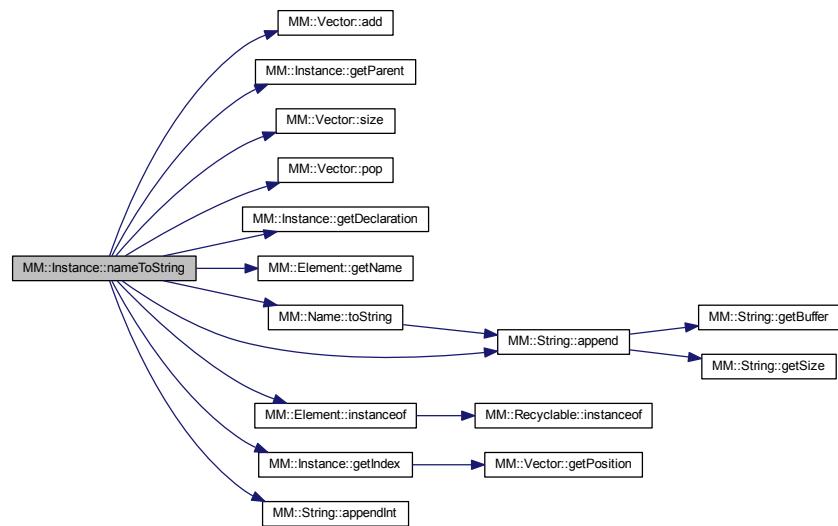
6.53.2.33 MM::VOID Instance::mark ()

Here is the caller graph for this function:



6.53.2.34 MM::VOID Instance::nameToString (MM::String * buf)

Here is the call graph for this function:

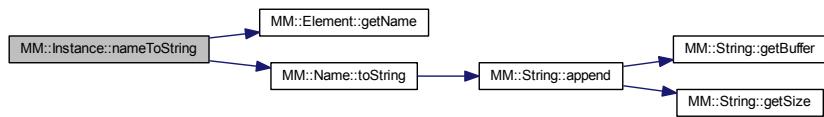


Here is the caller graph for this function:



6.53.2.35 MM::VOID Instance::nameToString (MM::Element * element, MM::String * buf)

Here is the call graph for this function:



6.53.2.36 MM::VOID Instance::recycle (MM::Recycler * r) [virtual]

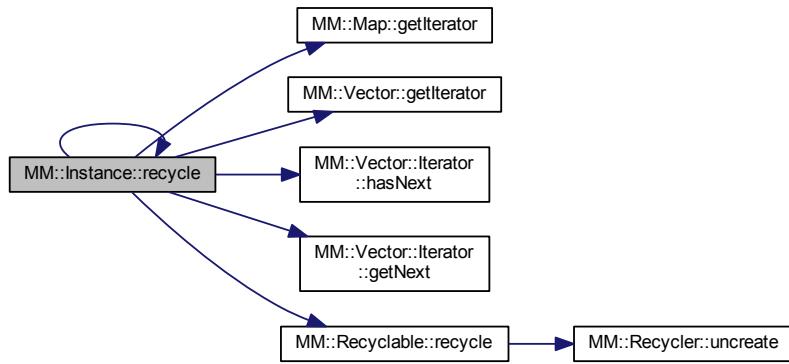
Recycles an [Instance](#) object in a [Recycler](#).

Parameters

<i>r</i>	Recycler object
----------	-----------------

Reimplemented from [MM::Recyclable](#).

Here is the call graph for this function:

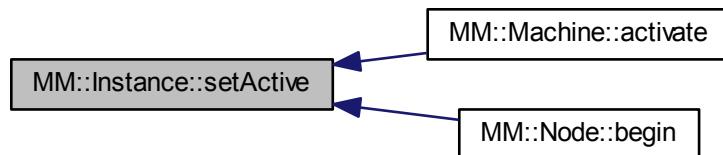


Here is the caller graph for this function:



6.53.2.37 MM::VOID Instance::setActive (MM::Node * node)

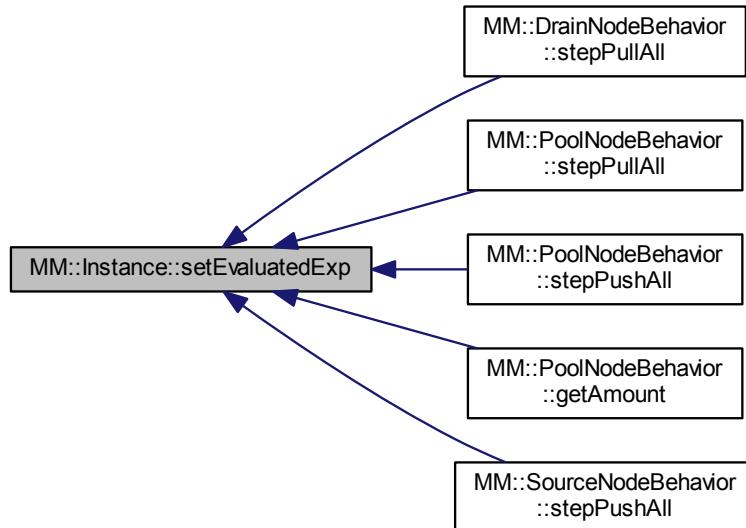
Here is the caller graph for this function:



6.53.2.38 MM::VOID Instance::setDisabled (MM::Node * node)

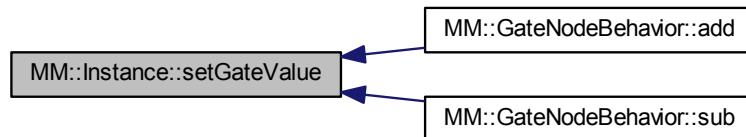
6.53.2.39 MM::VOID Instance::setEvaluatedExp (MM::Exp * exp, MM::INT32 val)

Here is the caller graph for this function:



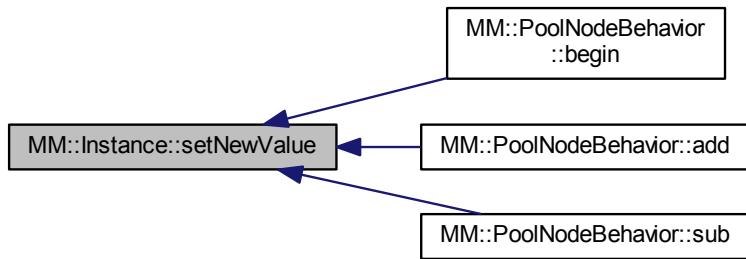
6.53.2.40 MM::VOID Instance::setGateValue (MM::Node * node, MM::UINT32 value)

Here is the caller graph for this function:

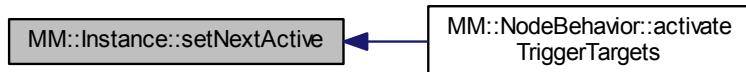


6.53.2.41 MM::VOID Instance::setNewValue (MM::Node * node, MM::UINT32 value)

Here is the caller graph for this function:

**6.53.2.42 MM::VOID Instance::setNextActive (MM::Node * node)**

Here is the caller graph for this function:

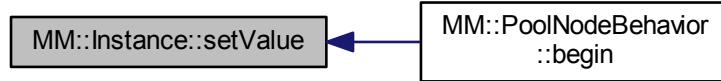
**6.53.2.43 MM::VOID Instance::setOldValue (MM::Node * node, MM::UINT32 value)**

Here is the caller graph for this function:



6.53.2.44 MM::VOID Instance::setValue (MM::Node * node, MM::UINT32 value)

Here is the caller graph for this function:

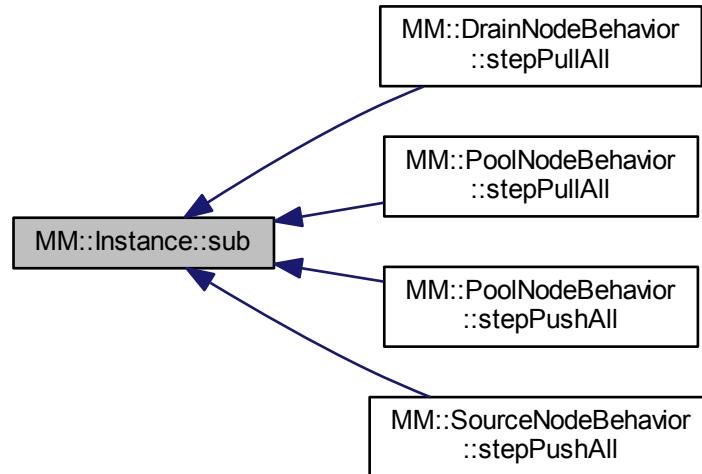


6.53.2.45 MM::VOID Instance::sub (MM::Node * node, MM::Machine * m, MM::UINT32 amount)

Here is the call graph for this function:

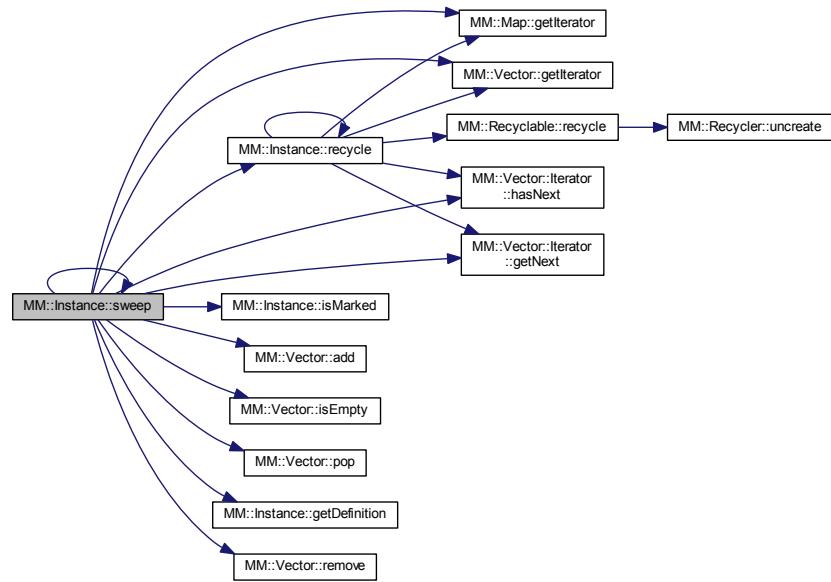


Here is the caller graph for this function:



6.53.2.46 MM::VOID Instance::sweep (MM::Machine * m)

Here is the call graph for this function:



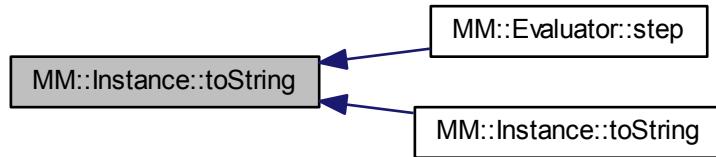
Here is the caller graph for this function:



6.53.2.47 MM::VOID Instance::toString (MM::String * buf) [virtual]

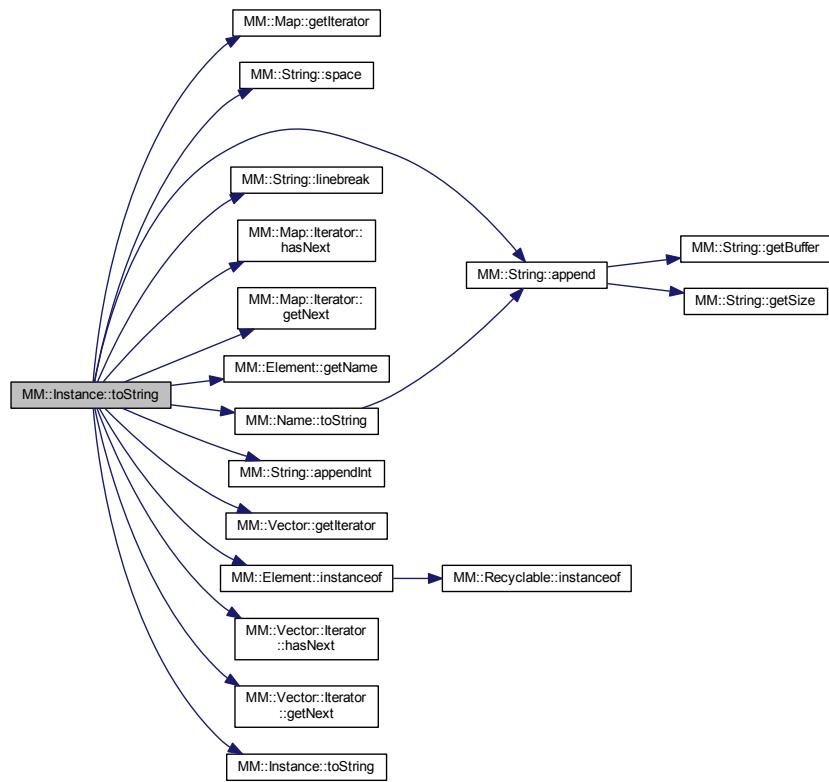
Implements [MM::Recyclable](#).

Here is the caller graph for this function:



6.53.2.48 MM::VOID Instance::toString (MM::String * buf, MM::UINT32 indent)

Here is the call graph for this function:



6.53.2.49 MM::VOID Instance::update (MM::Observable * observable, MM::VOID * aux, MM::UINT32 message, MM::VOID * object) [virtual]

Updates an [Observer](#).

Parameters

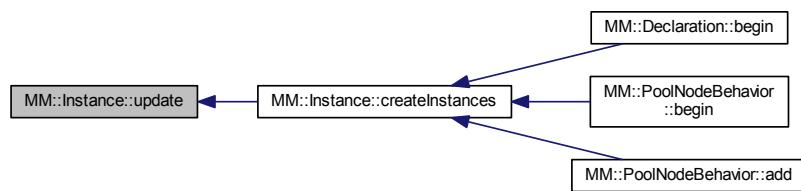
<i>observable</i>	Observable object
<i>aux</i>	Auxiliary argument
<i>message</i>	Message to specify what changed
<i>object</i>	Object that changed with respect to observable

Note

changes to instances are made using the observer pattern, the message redistribution looks clunky because it might just as well just use NEW DEL and UPD messages which requires less code.

Implements [MM::Observer](#).

Here is the caller graph for this function:



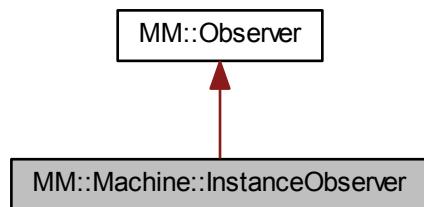
The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Instance.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Instance.cpp](#)

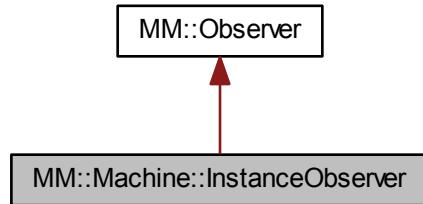
6.54 MM::Machine::InstanceObserver Class Reference

```
#include <Machine.h>
```

Inheritance diagram for MM::Machine::InstanceObserver:



Collaboration diagram for MM::Machine::InstanceObserver:



Public Member Functions

- `InstanceObserver (MM::UINT32 caller, MM::Instance *instance, MM::CALLBACK callback)`
- `~InstanceObserver ()`
- `MM::TID getTypeld ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::VOID update (MM::Observable *observable, MM::VOID *aux, MM::UINT32 message, MM::VOID *object)`

6.54.1 Detailed Description

log buffer size

6.54.2 Constructor & Destructor Documentation

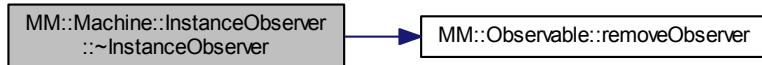
6.54.2.1 `MM::Machine::InstanceObserver::InstanceObserver (MM::UINT32 caller, MM::Instance * instance, MM::CALLBACK callback) [inline]`

Here is the call graph for this function:



6.54.2.2 MM::Machine::InstanceObserver::~InstanceObserver() [inline]

Here is the call graph for this function:



6.54.3 Member Function Documentation

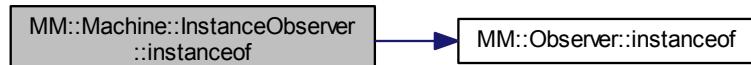
6.54.3.1 MM::TID MM::Machine::InstanceObserver::getTypId() [inline], [virtual]

Reimplemented from [MM::Observer](#).

6.54.3.2 MM::BOOLEAN MM::Machine::InstanceObserver::instanceof(MM::TID tid) [inline], [virtual]

Reimplemented from [MM::Observer](#).

Here is the call graph for this function:



6.54.3.3 MM::VOID MM::Machine::InstanceObserver::update(MM::Observable * observable, MM::VOID * aux, MM::UINT32 message, MM::VOID * object) [inline], [virtual]

Implements [MM::Observer](#).

The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Machine.h](#)

6.55 InterfaceNode Class Reference

The [InterfaceNode](#) abstraction defines interface nodes on declarations and instance pools that result from observable nodes inside their respective definitions.

```
#include <InterfaceNode.h>
```

6.55.1 Detailed Description

The [InterfaceNode](#) abstraction defines interface nodes on declarations and instance pools that result from observable nodes inside their respective definitions.

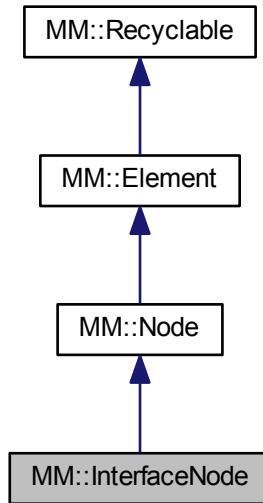
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[InterfaceNode.h](#)

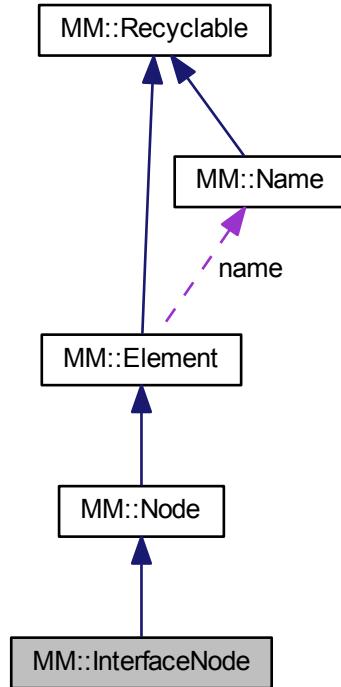
6.56 MM::InterfaceNode Class Reference

```
#include <InterfaceNode.h>
```

Inheritance diagram for MM::InterfaceNode:



Collaboration diagram for MM::InterfaceNode:



Public Member Functions

- `InterfaceNode (MM::Name *name, MM::Element *decl, MM::Node *ref)`
- `~InterfaceNode ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::NodeBehavior * getBehavior ()`
- `MM::Element * getDeclaration ()`
- `MM::Node * getNode ()`
- `MM::VOID add (MM::Instance *i, MM::UINT32 amount)`
- `MM::VOID sub (MM::Instance *i, MM::UINT32 amount)`
- `MM::UINT32 getCapacity (MM::Instance *i)`
- `MM::UINT32 getResources (MM::Instance *i)`
- `MM::BOOLEAN hasCapacity (MM::Instance *i, MM::UINT32 amount)`
- `MM::BOOLEAN hasResources (MM::Instance *i, MM::UINT32 amount)`
- `MM::VOID activateTriggerTargets (MM::Instance *i, MM::Machine *m)`
- `MM::VOID toString (MM::String *buf)`

Additional Inherited Members

6.56.1 Constructor & Destructor Documentation

6.56.1.1 `InterfaceNode::InterfaceNode (MM::Name * name, MM::Element * decl, MM::Node * ref)`

6.56.1.2 `InterfaceNode::~InterfaceNode ()`

6.56.2 Member Function Documentation

6.56.2.1 `MM::VOID InterfaceNode::activateTriggerTargets (MM::Instance * i, MM::Machine * m) [virtual]`

Reimplemented from [MM::Node](#).

6.56.2.2 `MM::VOID InterfaceNode::add (MM::Instance * i, MM::UINT32 amount)`

6.56.2.3 `MM::NodeBehavior * InterfaceNode::getBehavior () [virtual]`

Reimplemented from [MM::Node](#).

6.56.2.4 `MM::UINT32 InterfaceNode::getCapacity (MM::Instance * i) [virtual]`

Reimplemented from [MM::Node](#).

6.56.2.5 `MM::Element * InterfaceNode::getDeclaration ()`

Here is the caller graph for this function:



6.56.2.6 `MM::Node * InterfaceNode::getNode ()`

6.56.2.7 `MM::UINT32 InterfaceNode::getResources (MM::Instance * i) [virtual]`

Reimplemented from [MM::Node](#).

6.56.2.8 `MM::TID InterfaceNode::getTypeld () [virtual]`

Reimplemented from [MM::Node](#).

6.56.2.9 `MM::BOOLEAN InterfaceNode::hasCapacity (MM::Instance * i, MM::UINT32 amount) [virtual]`

Reimplemented from [MM::Node](#).

6.56.2.10 `MM::BOOLEAN InterfaceNode::hasResources (MM::Instance * i, MM::UINT32 amount) [virtual]`

Reimplemented from [MM::Node](#).

6.56.2.11 MM::BOOLEAN InterfaceNode::instanceof(MM::TID *tid*) [virtual]

Reimplemented from [MM::Node](#).

Here is the call graph for this function:



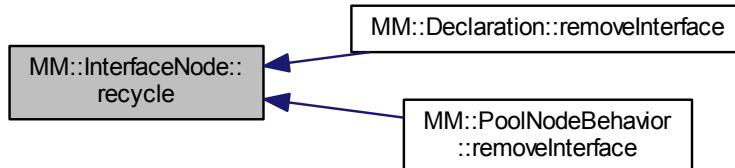
6.56.2.12 MM::VOID InterfaceNode::recycle(MM::Recycler * *r*) [virtual]

Reimplemented from [MM::Element](#).

Here is the call graph for this function:



Here is the caller graph for this function:

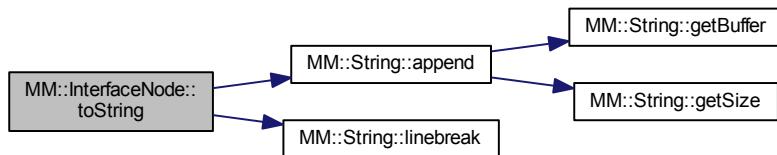


6.56.2.13 MM::VOID InterfaceNode::sub(MM::Instance * *i*, MM::UINT32 *amount*)

6.56.2.14 MM::VOID InterfaceNode::toString(MM::String * *buf*) [virtual]

Implements [MM::Element](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[InterfaceNode.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[InterfaceNode.cpp](#)

6.57 MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::Iterator Class Reference

```
#include <Map.h>
```

Public Member Functions

- [Iterator](#) (typename std::map< MAP_KEY, MAP_VALUE, COMPARE >::iterator begin, typename std::map< MAP_KEY, MAP_VALUE, COMPARE >::iterator end)
- [~Iterator](#) ()
- [MM::BOOLEAN hasNext](#) ()
- [MAP_VALUE getNext](#) ()
- [MAP_VALUE getNext](#) (MAP_KEY *key)
- [MM::VOID reset](#) ()

6.57.1 Constructor & Destructor Documentation

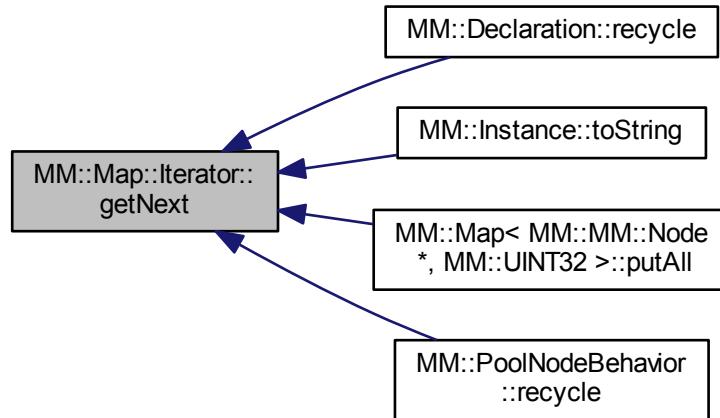
6.57.1.1 template<class MAP_KEY, class MAP_VALUE, class COMPARE = std::less<MAP_KEY>> **MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::Iterator**::[Iterator](#) (typename std::map< MAP_KEY, MAP_VALUE, COMPARE >::iterator begin, typename std::map< MAP_KEY, MAP_VALUE, COMPARE >::iterator end) [inline]

6.57.1.2 template<class MAP_KEY, class MAP_VALUE, class COMPARE = std::less<MAP_KEY>> **MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::Iterator**::[~Iterator](#) () [inline]

6.57.2 Member Function Documentation

6.57.2.1 `template<class MAP_KEY, class MAP_VALUE, class COMPARE = std::less<MAP_KEY>> MAP_VALUE MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::Iterator::getNext() [inline]`

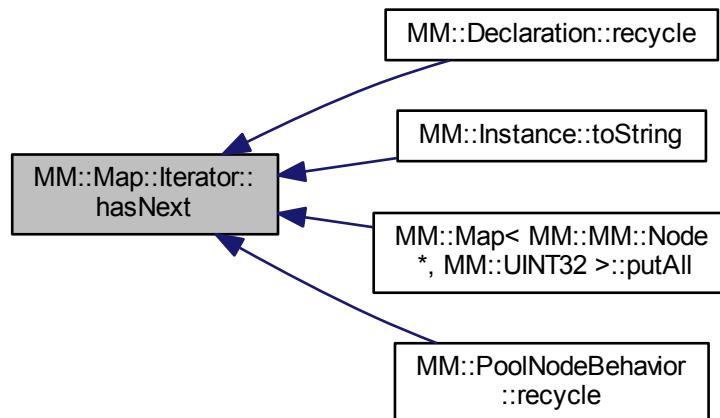
Here is the caller graph for this function:



6.57.2.2 `template<class MAP_KEY, class MAP_VALUE, class COMPARE = std::less<MAP_KEY>> MAP_VALUE MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::Iterator::getNext(MAP_KEY * key) [inline]`

6.57.2.3 `template<class MAP_KEY, class MAP_VALUE, class COMPARE = std::less<MAP_KEY>> MM::BOOLEAN MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::Iterator::hasNext() [inline]`

Here is the caller graph for this function:



6.57.2.4 template<class MAP_KEY, class MAP_VALUE, class COMPARE = std::less<MAP_KEY>> MM::VOID MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::Iterator::reset() [inline]

The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Map.h

6.58 MM::Vector< T >::Iterator Class Reference

```
#include <Vector.h>
```

Public Member Functions

- [Iterator](#) (typename std::vector< T >::iterator begin, typename std::vector< T >::iterator end)
- [~Iterator](#) ()
- [MM::BOOLEAN hasNext](#) ()
- [T getNext](#) ()
- [MM::VOID reset](#) ()

6.58.1 Constructor & Destructor Documentation

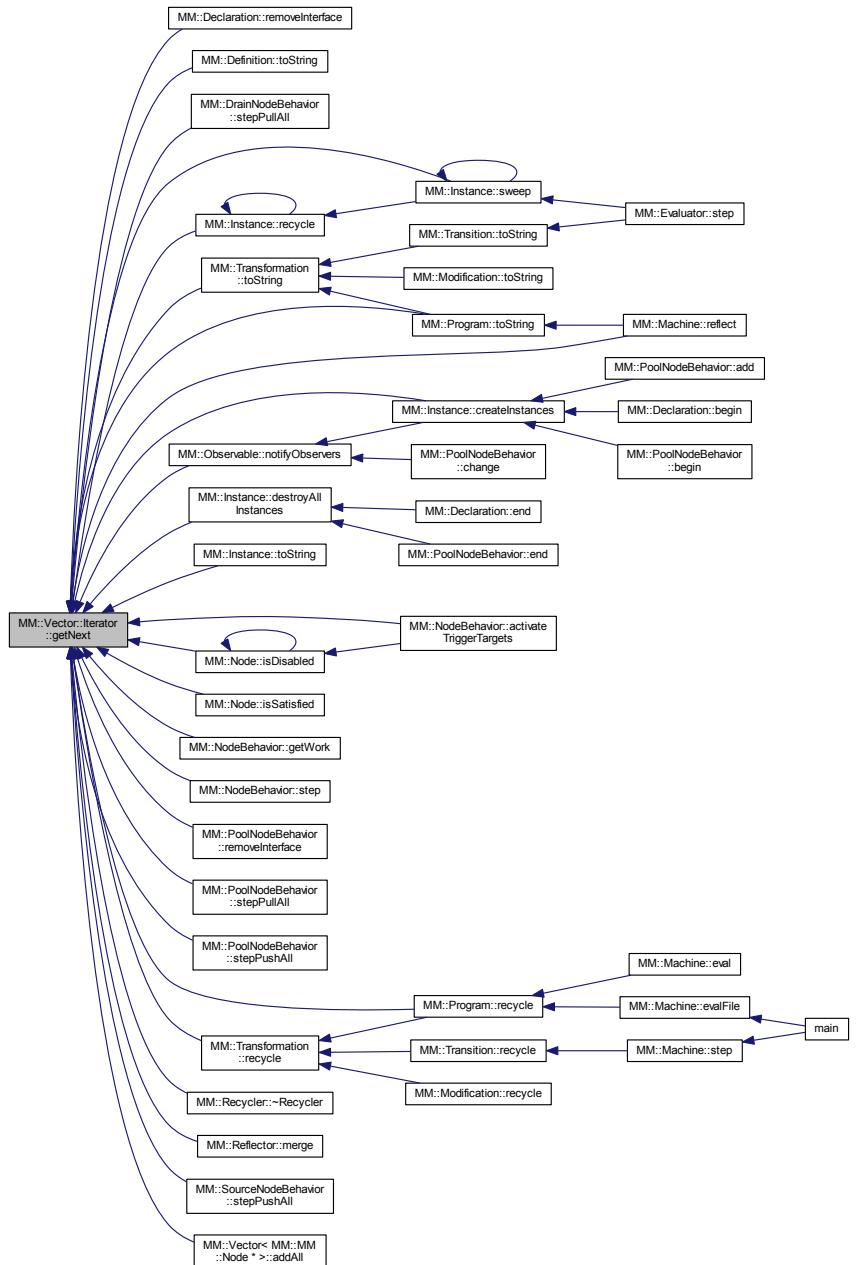
6.58.1.1 template<class T> MM::Vector< T >::Iterator::Iterator (typename std::vector< T >::iterator *begin*, typename std::vector< T >::iterator *end*) [inline]

6.58.1.2 template<class T> MM::Vector< T >::Iterator::~Iterator () [inline]

6.58.2 Member Function Documentation

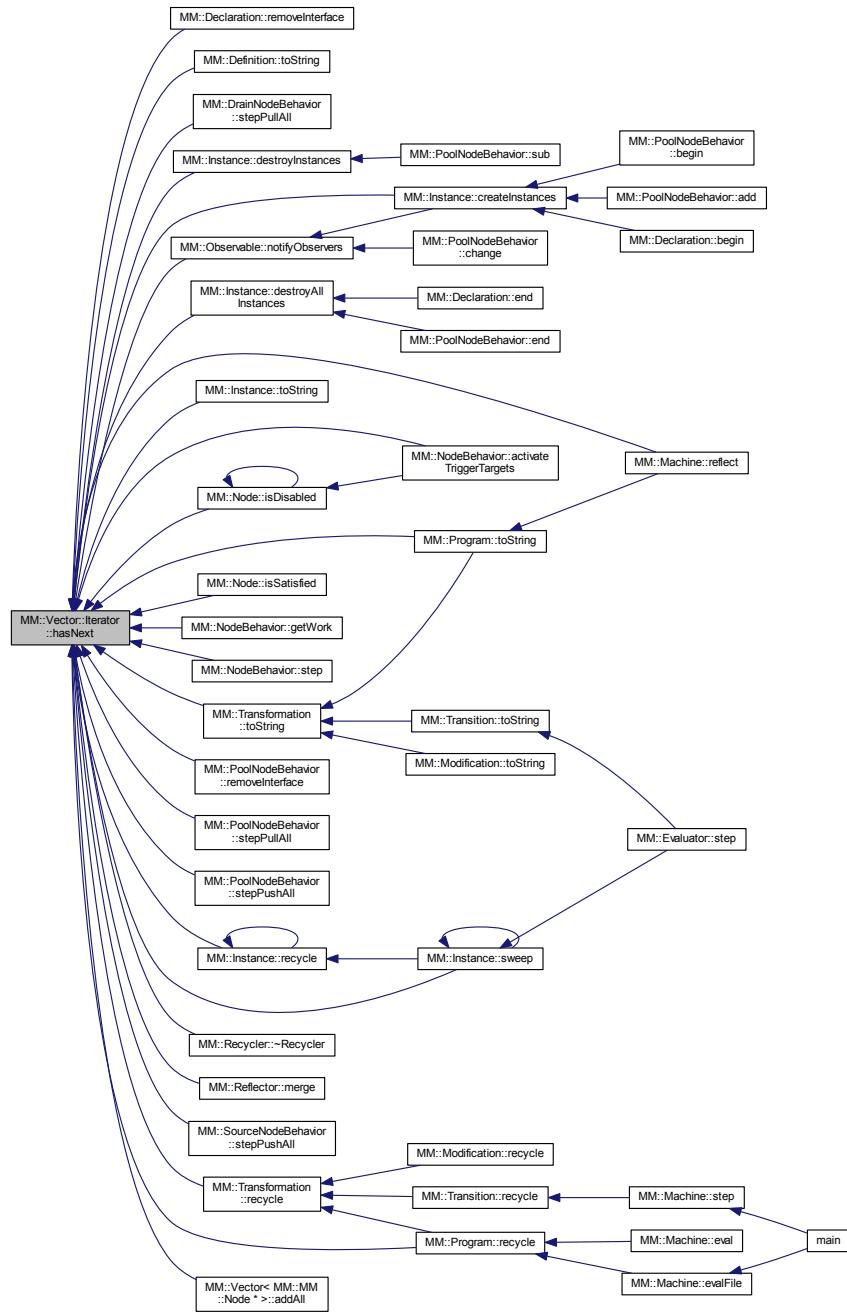
6.58.2.1 template<class T> T MM::Vector< T >::Iterator::getNext() [inline]

Here is the caller graph for this function:



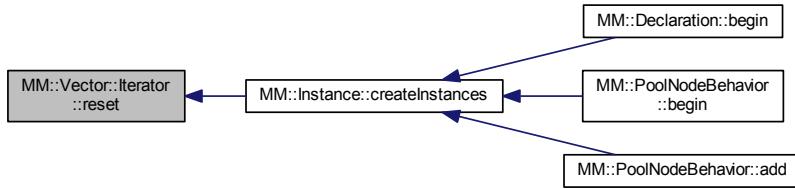
6.58.2.2 template<class T> MM::BOOLEAN MM::Vector< T >::Iterator::hasNext() [inline]

Here is the caller graph for this function:



6.58.2.3 template<class T> MM::VOID MM::Vector< T >::Iterator::reset() [inline]

Here is the caller graph for this function:



The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Vector.h](#)

6.59 Location Class Reference

The [Location](#) abstraction defines textual source locations.

```
#include <Location.h>
```

6.59.1 Detailed Description

The [Location](#) abstraction defines textual source locations.

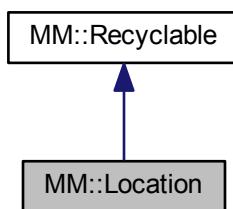
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Location.h](#)

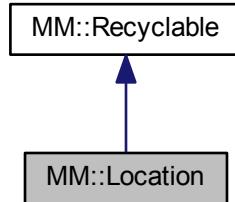
6.60 MM::Location Class Reference

```
#include <Location.h>
```

Inheritance diagram for MM::Location:



Collaboration diagram for MM::Location:



Public Member Functions

- [Location \(MM::UINT32 beginLine, MM::UINT32 beginCol, MM::UINT32 endLine, MM::UINT32 endCol\)](#)
Constructs a [Location](#) object.
- [~Location \(\)](#)
Destructs a [Location](#).
- [MM::VOID recycle \(MM::Recycler *r\)](#)
Recycles a [Location](#) object in a [Recycler](#).
- [MM::TID getTypeId \(\)](#)
Retrieves the type id of a [Location](#) object.
- [MM::BOOLEAN instanceof \(MM::TID tid\)](#)
Assesses if an object is an instance of a type tid.
- [MM::UINT32 getLine \(\)](#)
Retrieves the begin line number of a [Location](#) object.
- [MM::UINT32 getColumn \(\)](#)
Retrieves the begin column number of a [Location](#) object.
- [MM::UINT32 getLength \(\)](#)
Retrieves the length of a [Location](#) object.
- [MM::VOID toString \(MM::String *buf\)](#)
Serializes a [Location](#) object into a [String](#) buffer.

6.60.1 Constructor & Destructor Documentation

6.60.1.1 [Location::Location \(MM::UINT32 beginLine, MM::UINT32 beginCol, MM::UINT32 endLine, MM::UINT32 endCol \)](#)

Constructs a [Location](#) object.

end column number

Parameters

<i>beginLine</i>	line on which the location starts
<i>beginCol</i>	column on which the location starts
<i>endLine</i>	line on which the location ends
<i>endCol</i>	column on which the location ends

Returns

new [Location](#) object

6.60.1.2 Location::~Location()

Destructs a [Location](#).

6.60.2 Member Function Documentation**6.60.2.1 MM::UINT32 Location::getColumn()**

Retrieves the begin column number of a [Location](#) object.

Returns

column number

6.60.2.2 MM::UINT32 Location::getLength()

Retrieves the length of a [Location](#) object.

Returns

column number

6.60.2.3 MM::UINT32 Location::getLine()

Retrieves the begin line number of a [Location](#) object.

Returns

line number

6.60.2.4 MM::TID Location::getTypeld() [virtual]

Retrieves the type id of a [Location](#) object.

Returns

type id

Reimplemented from [MM::Recyclable](#).

6.60.2.5 MM::BOOLEAN Location::instanceof(MM::TID tid) [virtual]

Assesses if an object is an instance of a type tid.

Parameters

<i>tid</i>	type id
------------	---------

Returns

MM_TRUE if this object is instance of tid, MM_FALSE otherwise

Reimplemented from [MM::Recyclable](#).

Here is the call graph for this function:



6.60.2.6 MM::VOID Location::recycle(MM::Recycler * r) [virtual]

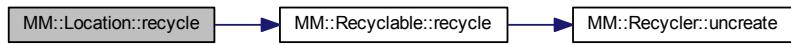
Recycles a [Location](#) object in a [Recycler](#).

Parameters

<i>r</i>	Recycler object
----------	-----------------

Reimplemented from [MM::Recyclable](#).

Here is the call graph for this function:



6.60.2.7 MM::VOID Location::toString(MM::String * buf) [virtual]

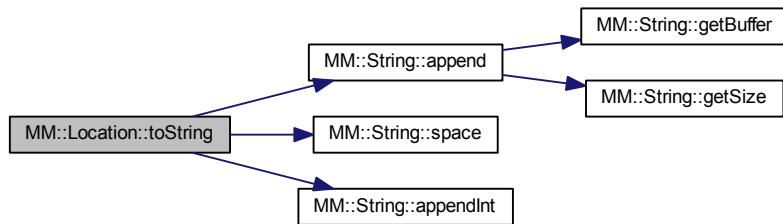
Serializes a [Location](#) object into a [String](#) buffer.

Parameters

<i>buf</i>	String buffer to serialize this object into
------------	---

Implements [MM::Recyclable](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Location.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Location.cpp](#)

6.61 Machine Class Reference

The [Machine](#) abstraction manages the creation of model elements, such that it can recycle and reuse them.

```
#include <Machine.h>
```

6.61.1 Detailed Description

The [Machine](#) abstraction manages the creation of model elements, such that it can recycle and reuse them.

Note

Factory

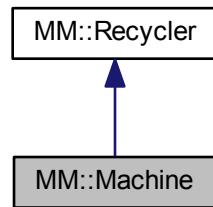
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Machine.h](#)

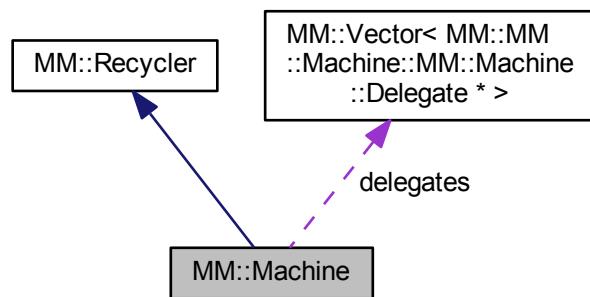
6.62 MM::Machine Class Reference

```
#include <Machine.h>
```

Inheritance diagram for MM::Machine:



Collaboration diagram for MM::Machine:



Classes

- class [Delegate](#)
- class [InstanceObserver](#)

Public Member Functions

- [Machine \(\)](#)
- [~Machine \(\)](#)
- [MM::TID getTypeld \(\)](#)
- [MM::BOOLEAN instanceof \(MM::TID tid\)](#)
- [MM::Reflector * getReflector \(\)](#)
- [MM::Evaluator * getEvaluator \(\)](#)
- [MM::Definition * getDefinition \(\)](#)
- [MM::Instance * getInstance \(\)](#)
- [MM::String * getLog \(\)](#)
- [MM::VOID setDefinition \(MM::Definition *def\)](#)
- [MM::VOID setInstance \(MM::Instance *inst\)](#)
- [MM::VOID eval \(const MM::CHAR *input\)](#)

- MM::VOID evalFile (const MM::CHAR *file)
- MM::VOID reflect (MM::Program *program)
- MM::VOID step ()
- MM::VOID step (MM::CHAR *buf, MM::UINT32 size)
- MM::VOID step (MM::UINT32 instance, MM::CHAR *buf, MM::UINT32 size)
- MM::VOID reset ()
- MM::VOID reset (MM::UINT32 instance)
- MM::VOID activate (MM::UINT32 node, MM::UINT32 instance)
- MM::UINT32 getInstance (MM::UINT32 instance, MM::CHAR *name)
- MM::VOID getName (MM::UINT32 element, MM::CHAR *buffer, MM::UINT32 bufferSize)
- MM::VOID getInstanceName (MM::UINT32 instance, MM::CHAR *buffer, MM::UINT32 bufferSize)
- MM::UINT32 addInstanceObserver (MM::UINT32 instance, MM::UINT32 caller, MM::CALLBACK callback)
- MM::UINT32 addDefinitionObserver (MM::UINT32 definition, MM::UINT32 caller, MM::CALLBACK callback)
- MM::VOID removeObserver (MM::UINT32 observer)
- MM::Vector< MM::Transformation * > * createTransformationVector ()
- MM::Vector< MM::Element * > * createElementVector ()
- MM::Vector< MM::Node * > * createNodeVector ()
- MM::Vector< MM::Edge * > * createEdgeVector ()
- MM::Map< MM::Name
*, MM::Element
*, MM::Name::Compare > * createName2ElementMap ()
- MM::Map< MM::Name *, MM::Node
*, MM::Name::Compare > * createName2NodeMap ()
- MM::String * createString (MM::UINT32 size)
- MM::Location * createLocation (YYLTYPE *loc)
- MM::Name * createName (MM::Name *n1, MM::CHAR *str, YYLTYPE *strLoc)
- MM::Name * createName (MM::CHAR *str, MM::UINT32 *len, MM::UINT32 *start, MM::UINT32 *end)
- MM::VOID eatWhiteSpace (MM::CHAR *str, MM::UINT32 *start, MM::UINT32 *end)
- MM::Name * createName (MM::CHAR *str, YYLTYPE *strLoc)
- MM::Name * createName (MM::Name *name)
- MM::Name * createName (MM::CHAR *buf)
- MM::Program * createProgram ()
- MM::Program * createProgram (MM::Vector< MM::Transformation * > *transformations)
- MM::Modification * createModification ()
- MM::Modification * createModification (MM::Vector< MM::Element * > *transformations)
- MM::Modification * createModification (MM::Vector< MM::Element * > *transformations, YYLTYPE *modifyLoc)
- MM::Transition * createTransition ()
- MM::Transition * createTransition (MM::Vector< MM::Element * > *elements)
- MM::Transition * createTransition (MM::Vector< MM::Element * > *elements, YYLTYPE *stepLoc)
- MM::FlowEvent * createFlowEvent (MM::Instance *actInstance, MM::Node *actNode, MM::Edge *actEdge, MM::Instance *srcInstance, MM::Node *srcNode, MM::UINT32 amount, MM::Instance *tgtInstance, MM::Node *tgtNode)
- MM::TriggerEvent * createTriggerEvent (MM::Instance *instance, MM::Edge *edge)
- MM::TriggerEvent * createTriggerEvent (YYLTYPE *failLoc, MM::Name *name)
- MM::Prevention * createPrevention (MM::Instance *instance, MM::Edge *edge)
- MM::Prevention * createPrevention (YYLTYPE *preventLoc, MM::Name *name)
- MM::Failure * createFailure (MM::Instance *instance, MM::Node *node)
- MM::Failure * createFailure (YYLTYPE *failLoc, MM::Name *name)
- MM::Activation * createActivation (MM::Instance *instance, MM::Node *node)
- MM::Activation * createActivation (YYLTYPE *activateLoc, MM::Name *name)
- MM::Enablement * createEnablement (MM::Instance *instance, MM::Node *node)
- MM::Enablement * createEnablement (YYLTYPE *enableLoc, MM::Name *name)
- MM::Disablement * createDisablement (MM::Instance *instance, MM::Node *node)
- MM::Disablement * createDisablement (YYLTYPE *disableLoc, MM::Name *name)

- MM::Violation * createViolation (MM::Instance *instance, MM::Assertion *assertion)
- MM::Violation * createViolation (YYLTYPE *deleteLoc, MM::Name *name)
- MM::Node * createSourceNode (MM::NodeBehavior::IO io, MM::NodeBehavior::When when, MM::Name *name)
- MM::Node * createDrainNode (MM::NodeBehavior::IO io, MM::NodeBehavior::When when, MM::NodeBehavior::How how, MM::Name *name)
- MM::Node * createGateNode (MM::NodeBehavior::IO io, MM::NodeBehavior::When when, MM::NodeBehavior::Act act, MM::NodeBehavior::How how, MM::Name *name)
- MM::Node * createPoolNode (MM::NodeBehavior::IO io, MM::NodeBehavior::When when, MM::NodeBehavior::Act act, MM::NodeBehavior::How how, MM::Name *name, MM::Name *of, MM::UINT32 at, MM::UINT32 max, MM::Exp *exp)
- MM::Node * createConverterNode (MM::NodeBehavior::IO io, MM::NodeBehavior::When when, MM::Name *name, MM::Name *from, MM::Name *to)
- MM::Node * createRefNode (MM::NodeBehavior::IO io, MM::Name *name)
- MM::InterfaceNode * createInterfaceNode (MM::Name *name, MM::Element *parent, MM::Node *ref)
- MM::StateEdge * createStateEdge (MM::Name *name, MM::Name *src, MM::Exp *exp, MM::Name *tgt)
- MM::StateEdge * createAnonymousTriggerEdge (MM::Node *src, MM::Node *tgt)
- MM::FlowEdge * createFlowEdge (MM::Name *name, MM::Name *src, MM::Exp *exp, MM::Name *tgt)
- MM::Definition * createDefinition ()
- MM::Definition * createDefinition (MM::Name *name, MM::Vector< Element * > *elements)
- MM::Declaration * createDeclaration (MM::Name *type, MM::Name *name)
- MM::Assertion * createAssertion (YYLTYPE *assertLoc, MM::Name *name, MM::Exp *exp, MM::CHAR *msg)
- MM::Assertion * createAssertion (MM::Name *name, MM::Exp *exp, MM::CHAR *msg)
- MM::Deletion * createDeletion (MM::Name *name)
- MM::Deletion * createDeletion (YYLTYPE *deleteLoc, MM::Name *name)
- MM::UnExp * createUnExp (MM::Operator::OP op, YYLTYPE *opLoc, MM::Exp *exp)
- MM::UnExp * createUnExp (MM::Operator::OP op, MM::Exp *exp)
- MM::BinExp * createBinExp (MM::Exp *e1, MM::Operator::OP op, YYLTYPE *opLoc, MM::Exp *e2)
- MM::BinExp * createBinExp (MM::Exp *e1, MM::Operator::OP op, MM::Exp *e2)
- MM::OverrideExp * createOverrideExp (YYLTYPE *lparenLoc, MM::Exp *e, YYLTYPE *rparenLoc)
- MM::OverrideExp * createOverrideExp (MM::Exp *e)
- MM::RangeValExp * createRangeValExp (MM::INT32 v1, YYLTYPE *v1Loc, YYLTYPE *rangeLoc, MM::INT32 v2, YYLTYPE *v2Loc)
- MM::RangeValExp * createRangeValExp (MM::INT32 v1, MM::INT32 v2)
- MM::NumberValExp * createNumberValExp (MM::INT32 val, YYLTYPE *valLoc)
- MM::NumberValExp * createNumberValExp (MM::INT32 val)
- MM::BooleanValExp * createBooleanValExp (MM::BOOLEAN val, YYLTYPE *valLoc)
- MM::BooleanValExp * createBooleanValExp (MM::BOOLEAN val)
- MM::AllExp * createAllExp (YYLTYPE *allLoc)
- MM::ActiveExp * createActiveExp (YYLTYPE *activeLoc, MM::Name *name)
- MM::ActiveExp * createActiveExp (MM::Name *name)
- MM::AliasExp * createAliasExp (YYLTYPE *aliasLoc)
- MM::AliasExp * createAliasExp ()
- MM::OneExp * createOneExp (YYLTYPE *epsilonLoc)
- MM::OneExp * createOneExp ()
- MM::VarExp * createVarExp (MM::Name *name)
- MM::Instance * createInstance (MM::Instance *parent, MM::Definition *def, MM::Element *decl)

Public Attributes

- MM::Vector
< MM::Machine::Delegate * > * delegates

Static Public Attributes

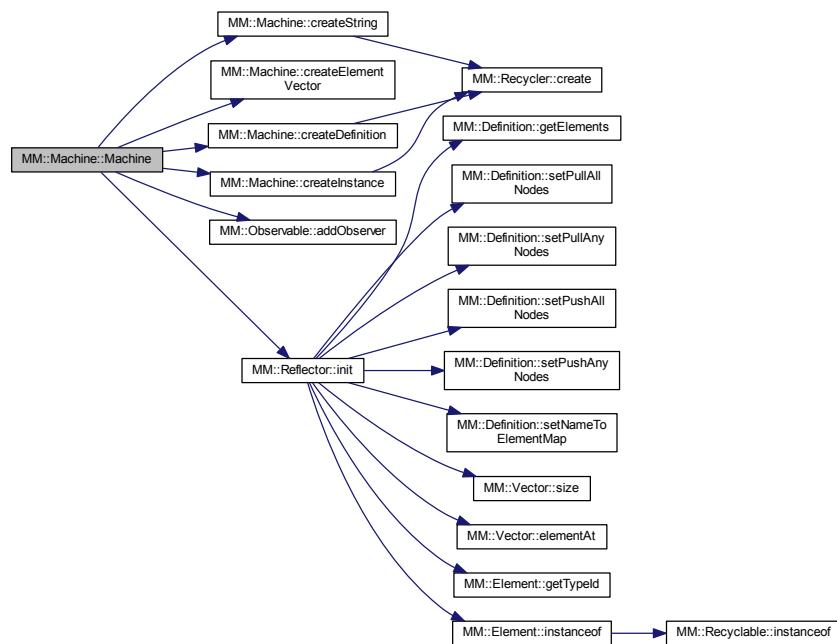
- static const `MM::UINT32 LOG_SIZE = 1024 * 32 * 8`

Additional Inherited Members

6.62.1 Constructor & Destructor Documentation

6.62.1.1 `Machine::Machine()`

Here is the call graph for this function:



6.62.1.2 `Machine::~Machine()`

6.62.2 Member Function Documentation

6.62.2.1 `MM::VOID Machine::activate(MM::UINT32 node, MM::UINT32 instance)`

Here is the call graph for this function:



6.62.2.2 **MM::UINT32 Machine::addDefinitionObserver (MM::UINT32 *definition*, MM::UINT32 *caller*, MM::CALLBACK *callback*)**

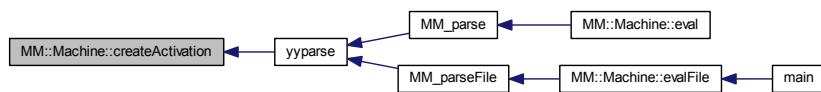
6.62.2.3 **MM::UINT32 Machine::addInstanceObserver (MM::UINT32 *instance*, MM::UINT32 *caller*, MM::CALLBACK *callback*)**

6.62.2.4 **MM::Activation * Machine::createActivation (MM::Instance * *instance*, MM::Node * *node*)**

Here is the call graph for this function:

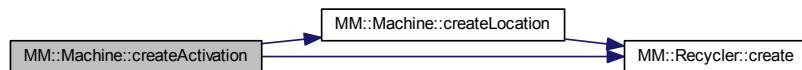


Here is the caller graph for this function:



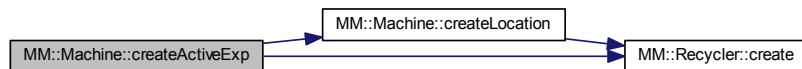
6.62.2.5 **MM::Activation * Machine::createActivation (YYLTYPE * *activateLoc*, MM::Name * *name*)**

Here is the call graph for this function:

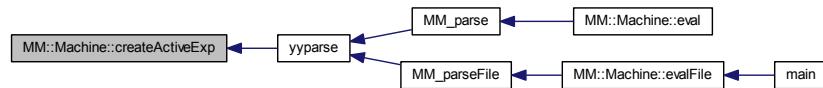


6.62.2.6 **MM::ActiveExp * Machine::createActiveExp (YYLTYPE * *activeLoc*, MM::Name * *name*)**

Here is the call graph for this function:



Here is the caller graph for this function:



6.62.2.7 MM::ActiveExp* MM::Machine::createActiveExp (MM::Name * name)

6.62.2.8 MM::AliasExp * Machine::createAliasExp (YYLTYPE * aliasLoc)

Here is the call graph for this function:



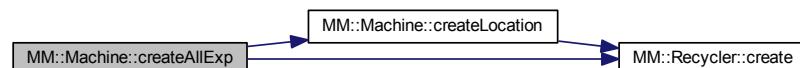
Here is the caller graph for this function:



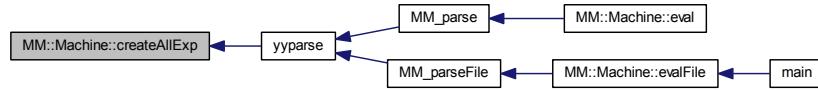
6.62.2.9 MM::AliasExp* MM::Machine::createAliasExp ()

6.62.2.10 MM::AllExp * Machine::createAllExp (YYLTYPE * allLoc)

Here is the call graph for this function:

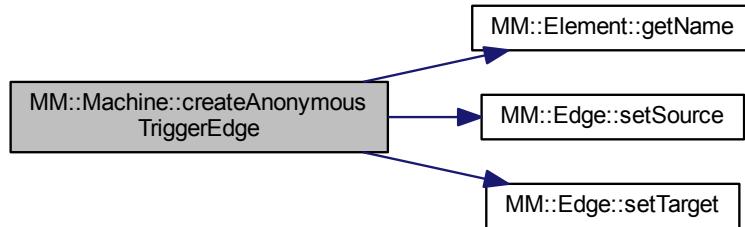


Here is the caller graph for this function:



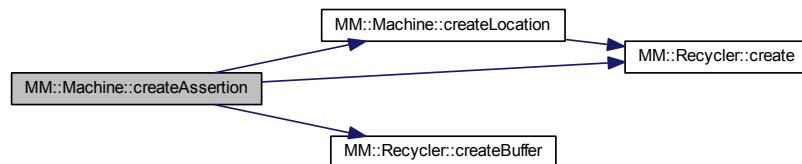
6.62.2.11 MM::StateEdge * Machine::createAnonymousTriggerEdge (MM::Node * src, MM::Node * tgt)

Here is the call graph for this function:



6.62.2.12 MM::Assertion * Machine::createAssertion (YYLTYPE * assertLoc, MM::Name * name, MM::Exp * exp, MM::CHAR * msg)

Here is the call graph for this function:

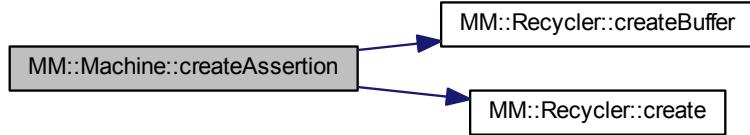


Here is the caller graph for this function:



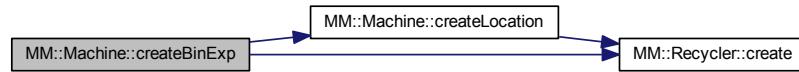
6.62.2.13 MM::Assertion * Machine::createAssertion (MM::Name * name, MM::Exp * exp, MM::CHAR * msg)

Here is the call graph for this function:

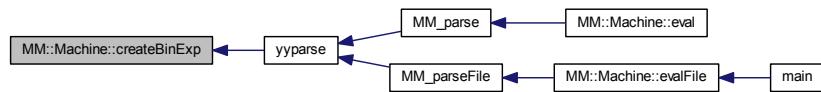


6.62.2.14 MM::BinExp * Machine::createBinExp (MM::Exp * e1, MM::Operator::OP op, YYLTYPE * opLoc, MM::Exp * e2)

Here is the call graph for this function:



Here is the caller graph for this function:



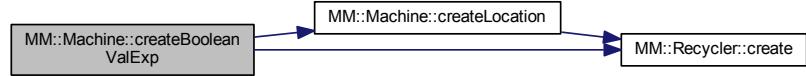
6.62.2.15 MM::BinExp * Machine::createBinExp (MM::Exp * e1, MM::Operator::OP op, MM::Exp * e2)

Here is the call graph for this function:

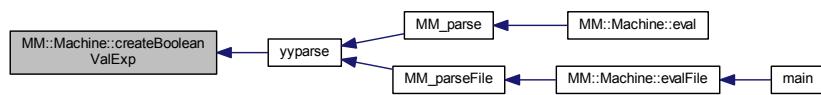


6.62.2.16 MM::BooleanValExp * Machine::createBooleanValExp (MM::BOOLEAN val, YYLTYPE * valLoc)

Here is the call graph for this function:

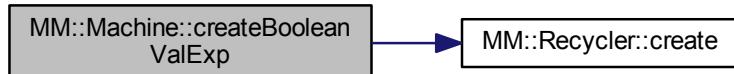


Here is the caller graph for this function:



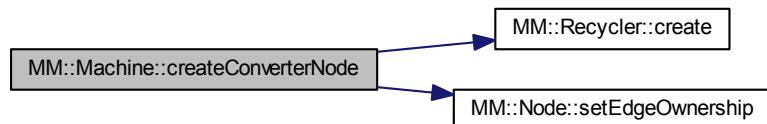
6.62.2.17 MM::BooleanValExp * Machine::createBooleanValExp (MM::BOOLEAN val)

Here is the call graph for this function:

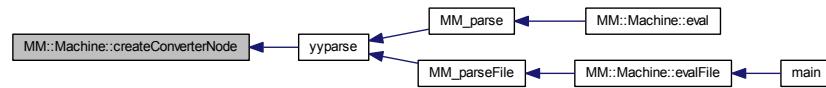


6.62.2.18 MM::Node * Machine::createConverterNode (MM::NodeBehavior::IO io, MM::NodeBehavior::When when, MM::Name * name, MM::Name * from, MM::Name * to)

Here is the call graph for this function:

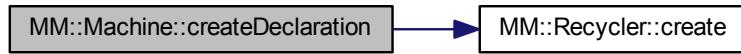


Here is the caller graph for this function:

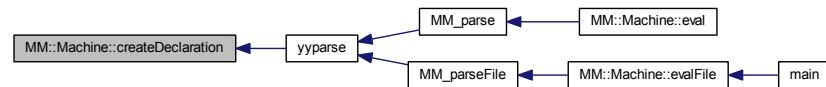


6.62.2.19 MM::Declaration * Machine::createDeclaration (MM::Name * type, MM::Name * name)

Here is the call graph for this function:



Here is the caller graph for this function:

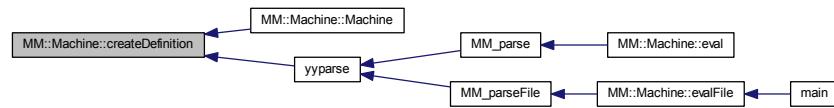


6.62.2.20 MM::Definition * Machine::createDefinition ()

Here is the call graph for this function:

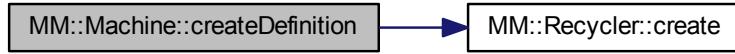


Here is the caller graph for this function:



6.62.2.21 MM::Definition * Machine::createDefinition (**MM::Name * name, MM::Vector< Element * > * elements**)

Here is the call graph for this function:

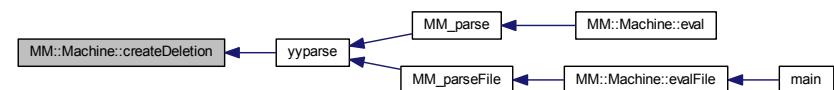


6.62.2.22 MM::Deletion * Machine::createDeletion (**MM::Name * name**)

Here is the call graph for this function:

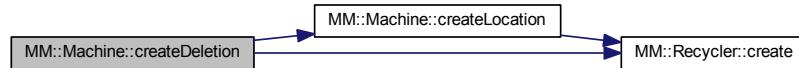


Here is the caller graph for this function:



6.62.2.23 MM::Deletion * Machine::createDeletion (YYLTYPE * deleteLoc, MM::Name * name)

Here is the call graph for this function:

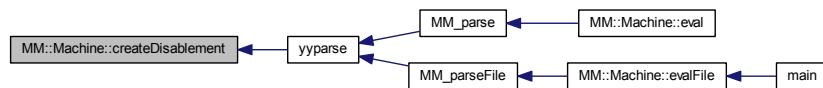


6.62.2.24 MM::Disablement * Machine::createDisablement (MM::Instance * instance, MM::Node * node)

Here is the call graph for this function:

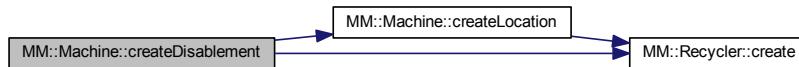


Here is the caller graph for this function:



6.62.2.25 MM::Disablement * Machine::createDisablement (YYLTYPE * disableLoc, MM::Name * name)

Here is the call graph for this function:

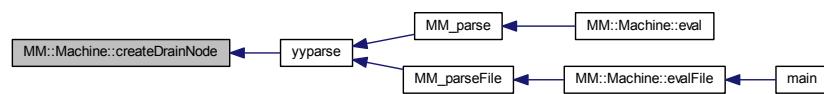


6.62.2.26 `MM::Node * Machine::createDrainNode (MM::NodeBehavior::IO io, MM::NodeBehavior::When when, MM::NodeBehavior::How how, MM::Name * name)`

Here is the call graph for this function:



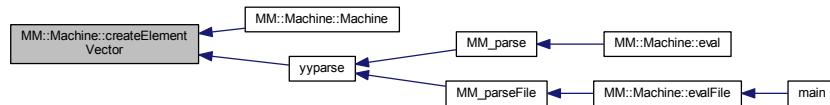
Here is the caller graph for this function:



6.62.2.27 `MM::Vector< MM::Edge * > * Machine::createEdgeVector ()`

6.62.2.28 `MM::Vector< MM::Element * > * Machine::createElementVector ()`

Here is the caller graph for this function:

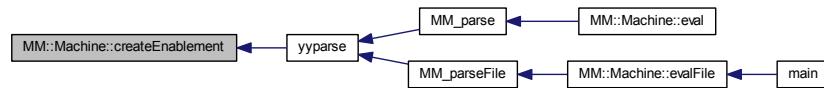


6.62.2.29 `MM::Enablement * Machine::createEnablement (MM::Instance * instance, MM::Node * node)`

Here is the call graph for this function:

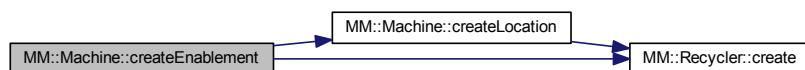


Here is the caller graph for this function:



6.62.2.30 MM::Enablement * Machine::createEnablement (YYLTYPE * disableLoc, MM::Name * name)

Here is the call graph for this function:

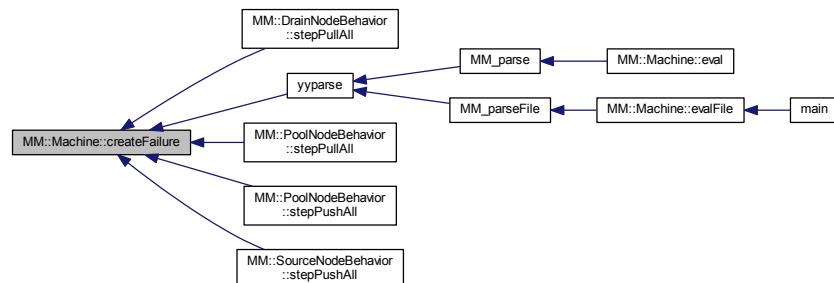


6.62.2.31 MM::Failure * Machine::createFailure (MM::Instance * instance, MM::Node * node)

Here is the call graph for this function:

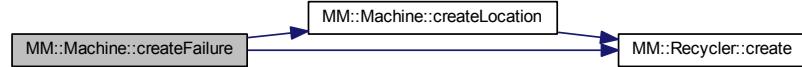


Here is the caller graph for this function:



6.62.2.32 MM::Failure * Machine::createFailure (YYLTYPE * failLoc, MM::Name * name)

Here is the call graph for this function:

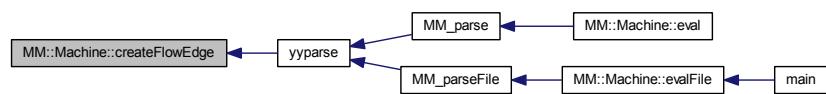


6.62.2.33 MM::FlowEdge * Machine::createFlowEdge (MM::Name * name, MM::Name * src, MM::Exp * exp, MM::Name * tgt)

Here is the call graph for this function:



Here is the caller graph for this function:

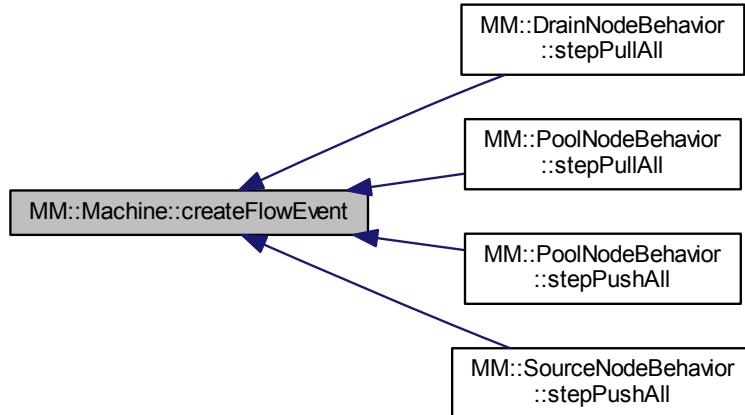


6.62.2.34 MM::FlowEvent * Machine::createFlowEvent (MM::Instance * actInstance, MM::Node * actNode, MM::Edge * actEdge, MM::Instance * srcInstance, MM::Node * srcNode, MM::UINT32 amount, MM::Instance * tgtInstance, MM::Node * tgtNode)

Here is the call graph for this function:



Here is the caller graph for this function:

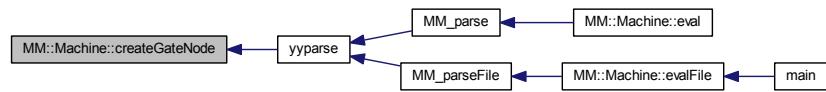


6.62.2.35 `MM::Node * Machine::createGateNode (MM::NodeBehavior::IO io, MM::NodeBehavior::When when, MM::NodeBehavior::Act act, MM::NodeBehavior::How how, MM::Name * name)`

Here is the call graph for this function:



Here is the caller graph for this function:

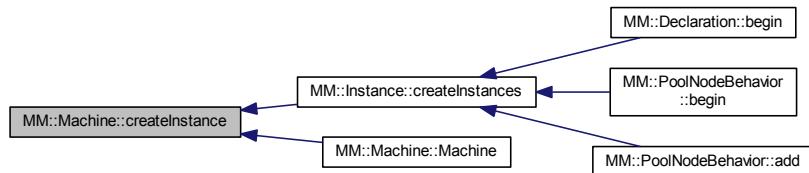


6.62.2.36 `MM::Instance * Machine::createInstance (MM::Instance * parent, MM::Definition * def, MM::Element * decl)`

Here is the call graph for this function:



Here is the caller graph for this function:

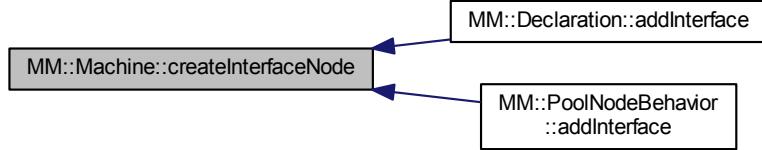


6.62.2.37 `MM::InterfaceNode * Machine::createInterfaceNode (MM::Name * name, MM::Element * parent, MM::Node * ref)`

Here is the call graph for this function:

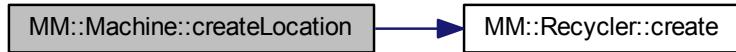


Here is the caller graph for this function:

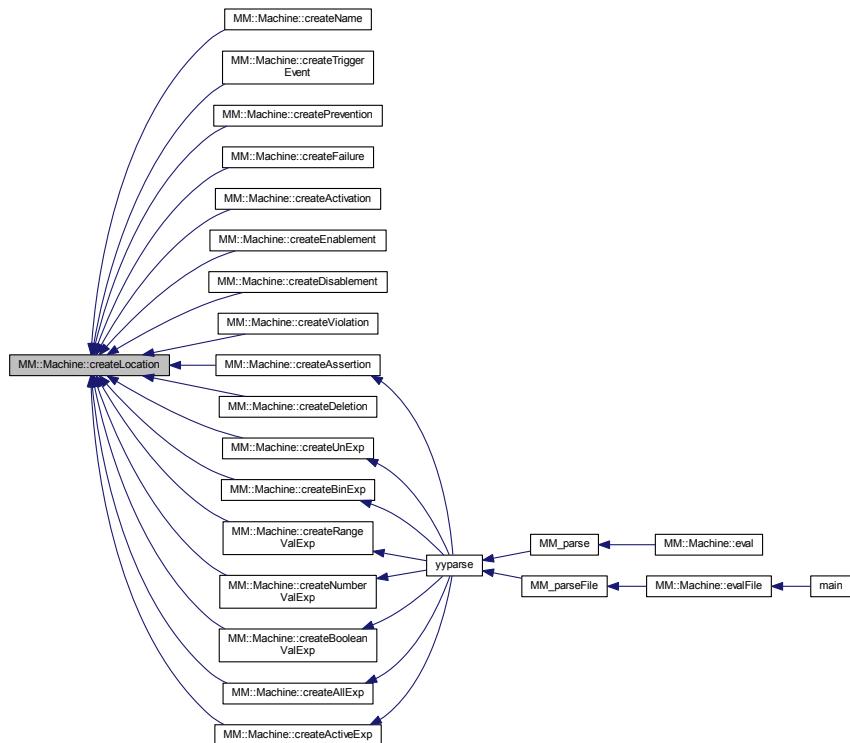


6.62.2.38 MM::Location * Machine::createLocation (YYLTYPE * loc)

Here is the call graph for this function:



Here is the caller graph for this function:

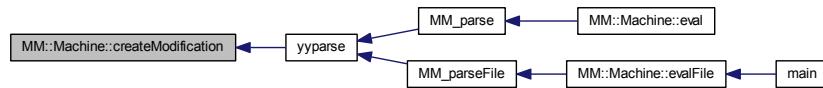


6.62.2.39 MM::Modification * Machine::createModification ()

Here is the call graph for this function:



Here is the caller graph for this function:



6.62.2.40 MM::Modification * Machine::createModification (MM::Vector< MM::Element * > * transformations)

Here is the call graph for this function:



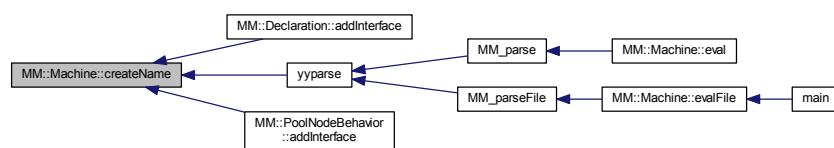
6.62.2.41 MM::Modification * Machine::createModification (MM::Vector< MM::Element * > * transformations, YYLTYPE * modifyLoc)

Here is the call graph for this function:



6.62.2.42 MM::Name* MM::Machine::createName (MM::Name * n1, MM::CHAR * str, YYLTYPE * strLoc)

Here is the caller graph for this function:



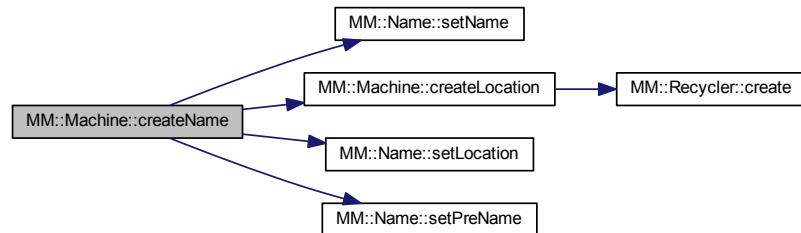
6.62.2.43 `MM::Name * Machine::createName (MM::CHAR * str, MM::UINT32 * len, MM::UINT32 * start, MM::UINT32 * end)`

Here is the call graph for this function:



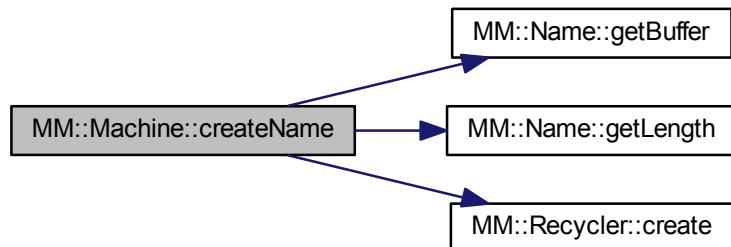
6.62.2.44 `MM::Name * Machine::createName (MM::CHAR * str, YYLTYPE * strLoc)`

Here is the call graph for this function:



6.62.2.45 `MM::Name * Machine::createName (MM::Name * name)`

Here is the call graph for this function:

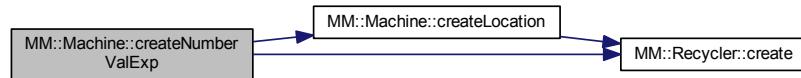


6.62.2.46 **MM::Name * Machine::createName (MM::CHAR * buf)**

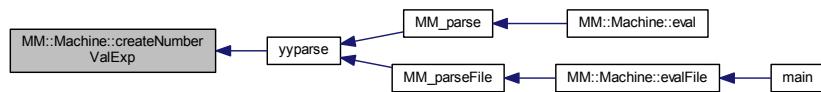
Here is the call graph for this function:

6.62.2.47 **MM::Map< MM::Name *, MM::Element *, MM::Name::Compare > * Machine::createName2ElementMap ()**6.62.2.48 **MM::Map< MM::Name *, MM::Node *, MM::Name::Compare > * Machine::createName2NodeMap ()**6.62.2.49 **MM::Vector< MM::Node * > * Machine::createNodeVector ()**6.62.2.50 **MM::NumberValExp * Machine::createNumberValExp (MM::INT32 val, YYLTYPE * valLoc)**

Here is the call graph for this function:

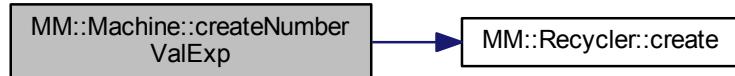


Here is the caller graph for this function:

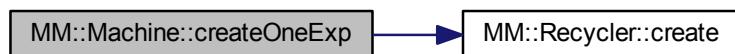


6.62.2.51 MM::NumberValExp * Machine::createNumberValExp (MM::INT32 val)

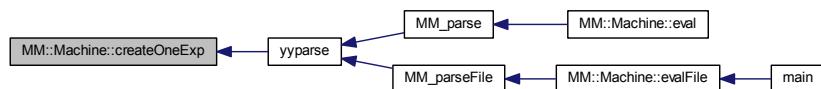
Here is the call graph for this function:

**6.62.2.52 MM::OneExp * Machine::createOneExp (YYLTYPE * epsilonLoc)**

Here is the call graph for this function:



Here is the caller graph for this function:

**6.62.2.53 MM::OneExp * Machine::createOneExp ()**

Here is the call graph for this function:

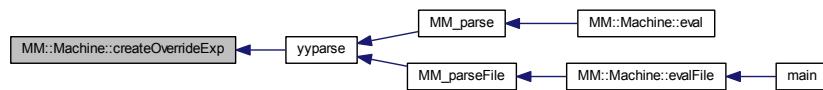


6.62.2.54 **MM::OverrideExp * Machine::createOverrideExp (YYLTYPE * lparenLoc, MM::Exp * e, YYLTYPE * rparenLoc)**

Here is the call graph for this function:



Here is the caller graph for this function:



6.62.2.55 **MM::OverrideExp * Machine::createOverrideExp (MM::Exp * e)**

Here is the call graph for this function:



6.62.2.56 **MM::Node * Machine::createPoolNode (MM::NodeBehavior::IO io, MM::NodeBehavior::When when, MM::NodeBehavior::Act act, MM::NodeBehavior::How how, MM::Name * name, MM::Name * of, MM::UINT32 at, MM::UINT32 max, MM::Exp * exp)**

Here is the call graph for this function:



Here is the caller graph for this function:

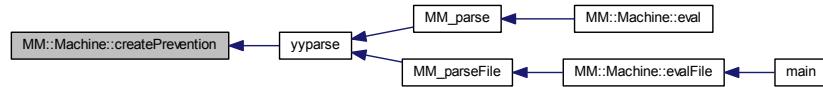


6.62.2.57 MM::Prevention * Machine::createPrevention (MM::Instance * *instance*, MM::Edge * *edge*)

Here is the call graph for this function:

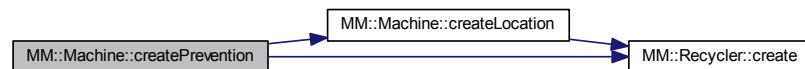


Here is the caller graph for this function:



6.62.2.58 MM::Prevention * Machine::createPrevention (YYLTYPE * *preventLoc*, MM::Name * *name*)

Here is the call graph for this function:

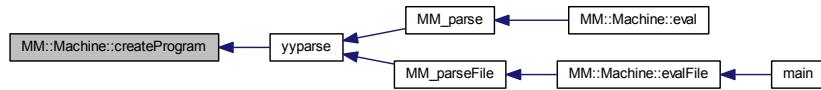


6.62.2.59 **MM::Program * Machine::createProgram ()**

Here is the call graph for this function:



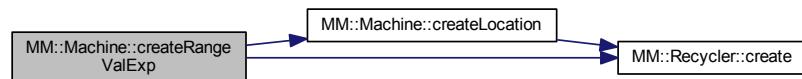
Here is the caller graph for this function:

6.62.2.60 **MM::Program * Machine::createProgram (MM::Vector< MM::Transformation * > * transformations)**

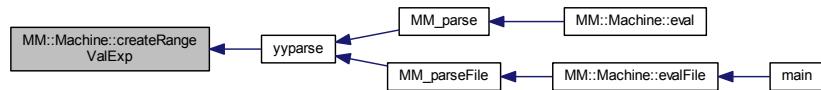
Here is the call graph for this function:

6.62.2.61 **MM::RangeValExp * Machine::createRangeValExp (MM::INT32 v1, YYLTYPE * v1Loc, YYLTYPE * rangeLoc, MM::INT32 v2, YYLTYPE * v2Loc)**

Here is the call graph for this function:



Here is the caller graph for this function:



6.62.2.62 MM::RangeValExp * Machine::createRangeValExp (MM::INT32 v1, MM::INT32 v2)

Here is the call graph for this function:

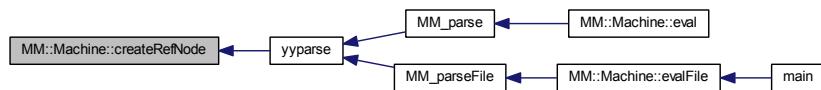


6.62.2.63 MM::Node * Machine::createRefNode (MM::NodeBehavior::IO io, MM::Name * name)

Here is the call graph for this function:



Here is the caller graph for this function:

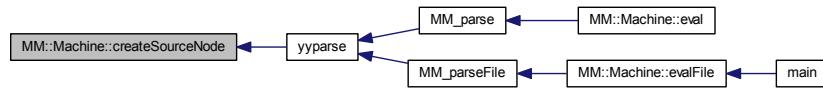


6.62.2.64 `MM::Node * Machine::createSourceNode (MM::NodeBehavior::IO io, MM::NodeBehavior::When when, MM::Name * name)`

Here is the call graph for this function:



Here is the caller graph for this function:

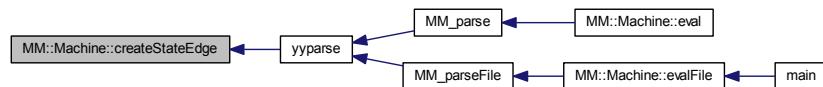


6.62.2.65 `MM::StateEdge * Machine::createStateEdge (MM::Name * name, MM::Name * src, MM::Exp * exp, MM::Name * tgt)`

Here is the call graph for this function:



Here is the caller graph for this function:



6.62.2.66 MM::String * Machine::createString (MM::UINT32 size)

Here is the call graph for this function:

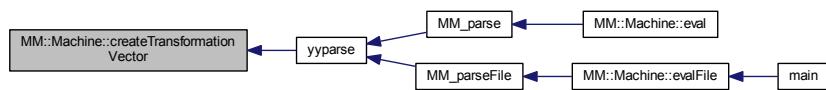


Here is the caller graph for this function:



6.62.2.67 MM::Vector< MM::Transformation * > * Machine::createTransformationVector ()

Here is the caller graph for this function:

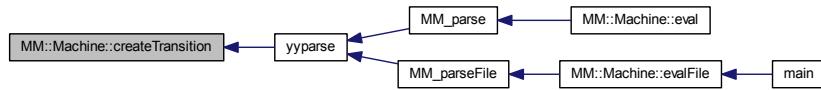


6.62.2.68 MM::Transition * Machine::createTransition ()

Here is the call graph for this function:



Here is the caller graph for this function:



6.62.2.69 MM::Transition * Machine::createTransition (MM::Vector< MM::Element * > * elements)

Here is the call graph for this function:



6.62.2.70 MM::Transition * Machine::createTransition (MM::Vector< MM::Element * > * elements, YYLTYPE * stepLoc)

Here is the call graph for this function:

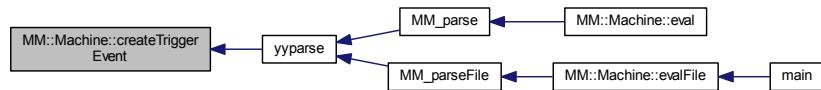


6.62.2.71 MM::TriggerEvent * Machine::createTriggerEvent (MM::Instance * instance, MM::Edge * edge)

Here is the call graph for this function:

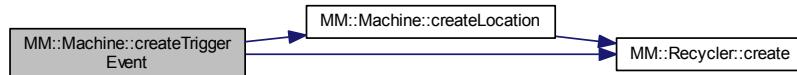


Here is the caller graph for this function:



6.62.2.72 MM::TriggerEvent * Machine::createTriggerEvent (YYLTYPE * failLoc, MM::Name * name)

Here is the call graph for this function:

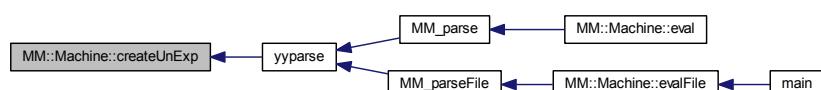


6.62.2.73 MM::UnExp * Machine::createUnExp (MM::Operator::OP op, YYLTYPE * opLoc, MM::Exp * exp)

Here is the call graph for this function:



Here is the caller graph for this function:



6.62.2.74 MM::UnExp * Machine::createUnExp (MM::Operator::OP *op*, MM::Exp * *exp*)

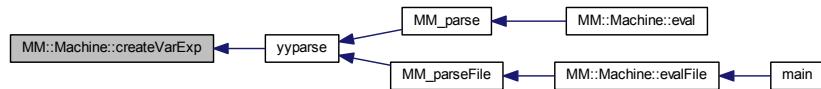
Here is the call graph for this function:

**6.62.2.75 MM::VarExp * Machine::createVarExp (MM::Name * *name*)**

Here is the call graph for this function:



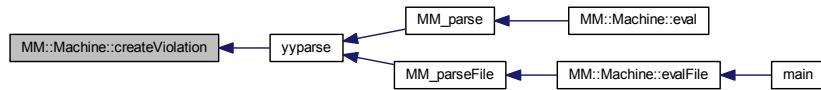
Here is the caller graph for this function:

**6.62.2.76 MM::Violation * Machine::createViolation (MM::Instance * *instance*, MM::Assertion * *assertion*)**

Here is the call graph for this function:

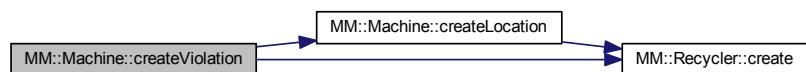


Here is the caller graph for this function:



6.62.2.77 `MM::Violation * Machine::createViolation (YYLTYPE * deleteLoc, MM::Name * name)`

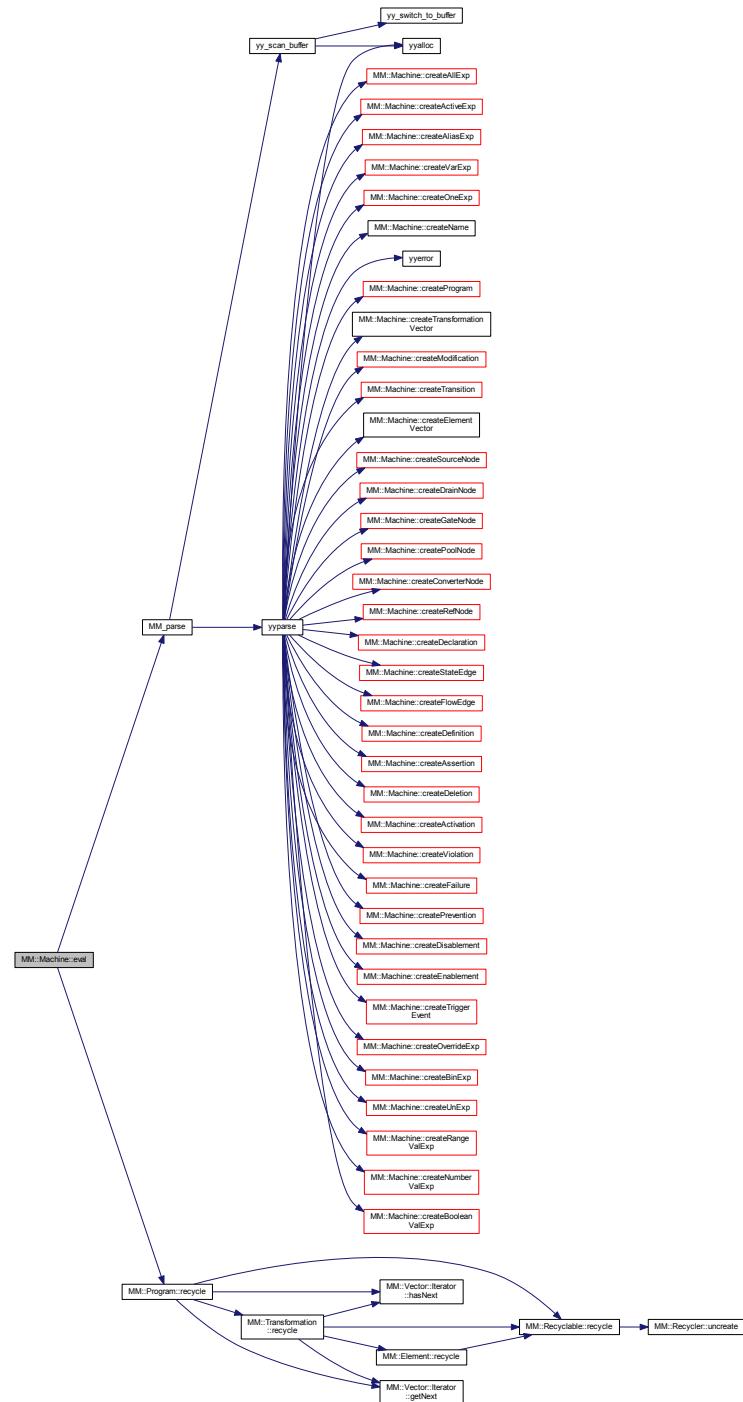
Here is the call graph for this function:



6.62.2.78 `MM::VOID Machine::eatWhiteSpace (MM::CHAR * str, MM::UINT32 * start, MM::UINT32 * end)`

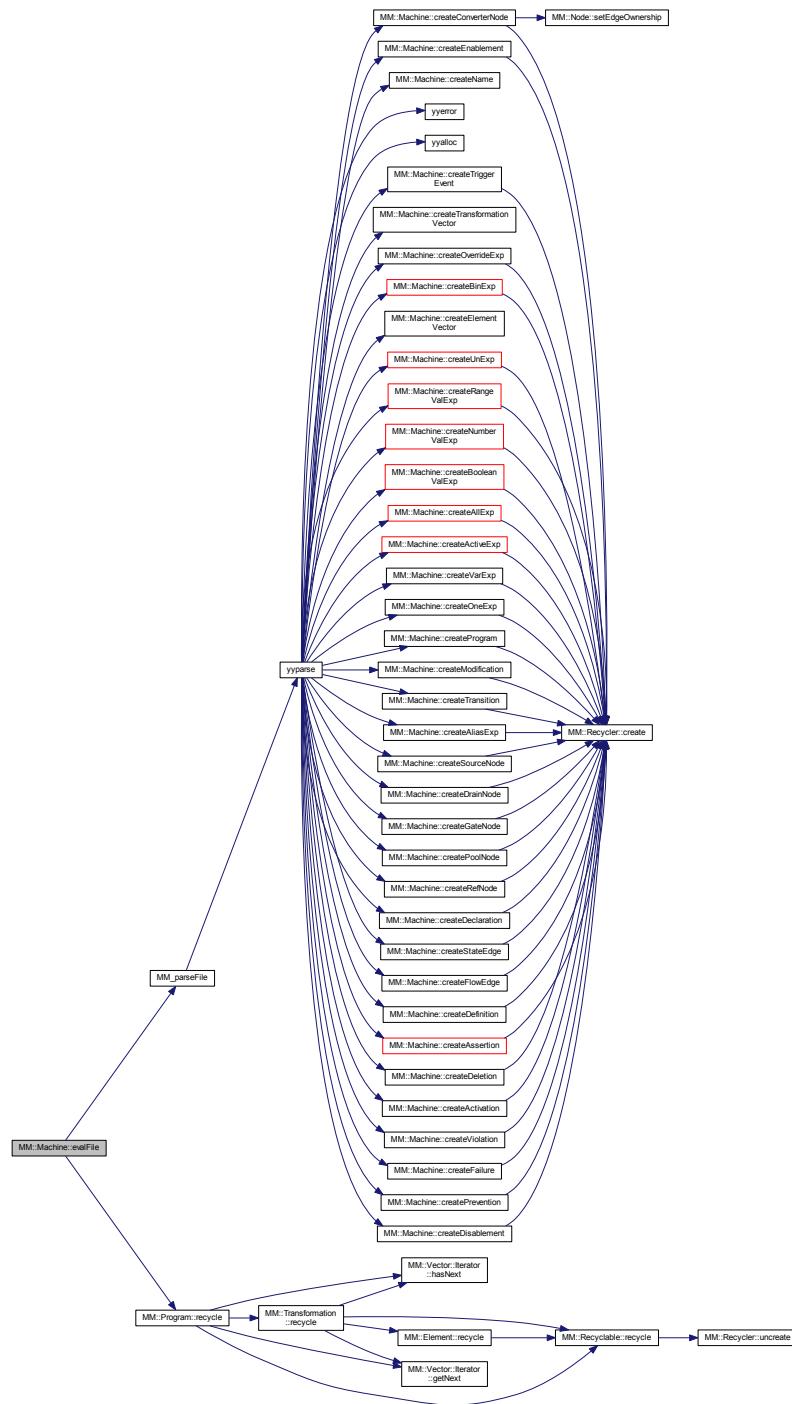
6.62.2.79 MM::VOID Machine::eval (const MM::CHAR * *input*)

Here is the call graph for this function:



6.62.2.80 MM::VOID Machine::evalFile (const MM::CHAR * file)

Here is the call graph for this function:



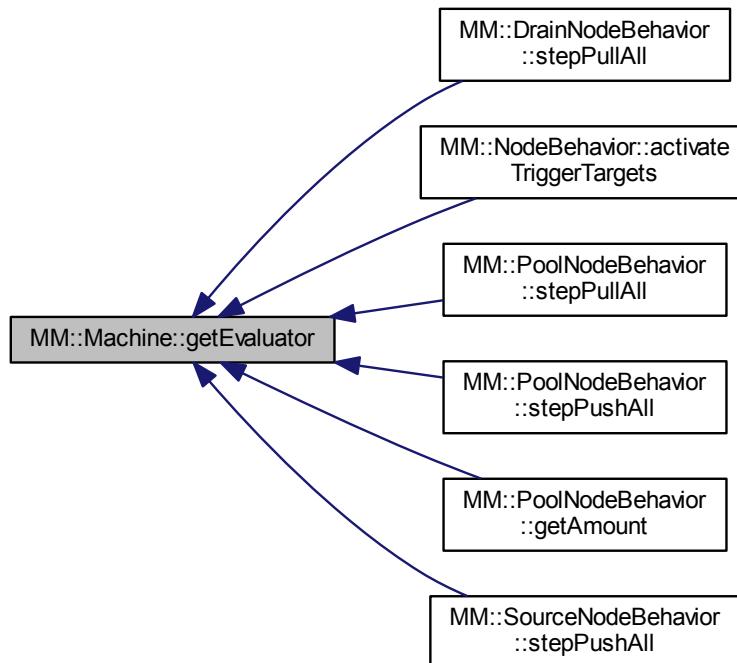
Here is the caller graph for this function:



6.62.2.81 MM::Definition * Machine::getDefinition()

6.62.2.82 MM::Evaluator * Machine::getEvaluator()

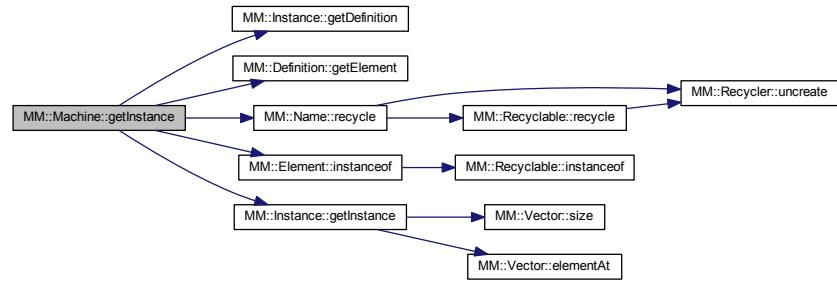
Here is the caller graph for this function:



6.62.2.83 MM::Instance * Machine::getInstance()

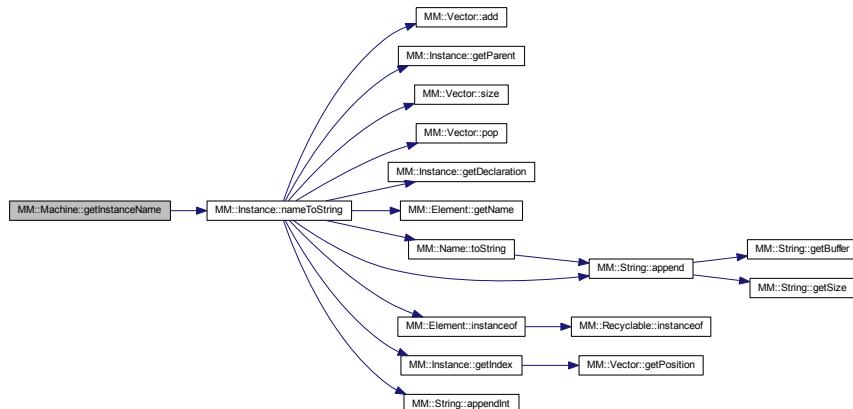
6.62.2.84 MM::UINT32 Machine::getInstance (MM::UINT32 *instance*, MM::CHAR * *name*)

Here is the call graph for this function:



6.62.2.85 MM::VOID Machine::getInstanceName (MM::UINT32 *instance*, MM::CHAR * *buffer*, MM::UINT32 *bufferSize*)

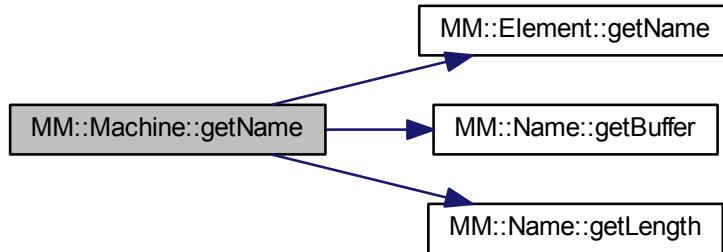
Here is the call graph for this function:



6.62.2.86 MM::String * Machine::getLog ()

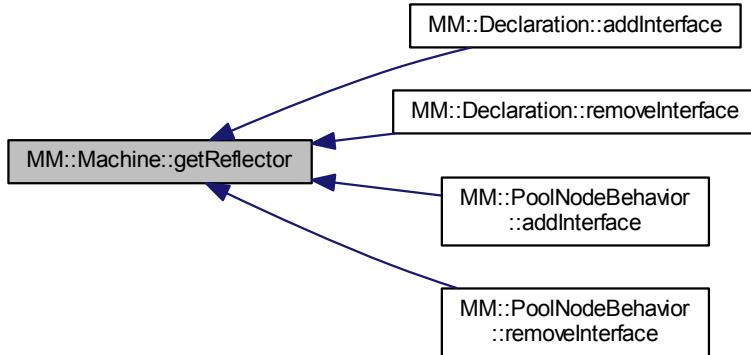
6.62.2.87 MM::VOID Machine::getName (MM::UINT32 element, MM::CHAR * buffer, MM::UINT32 bufferSize)

Here is the call graph for this function:



6.62.2.88 MM::Reflector * Machine::getReflector ()

Here is the caller graph for this function:



6.62.2.89 MM::TID Machine::getTypeld () [virtual]

Reimplemented from [MM::Recycler](#).

6.62.2.90 MM::BOOLEAN Machine::instanceof (MM::TID tid) [virtual]

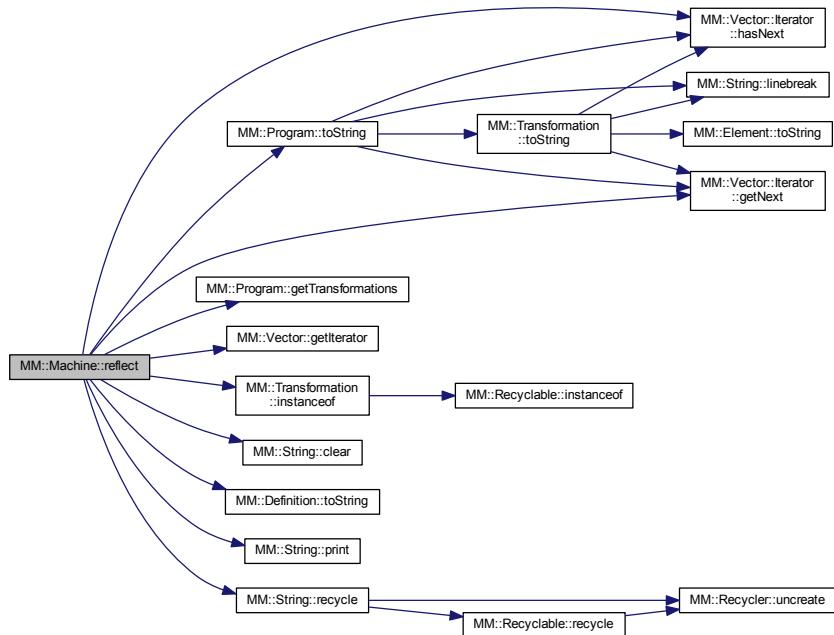
Reimplemented from [MM::Recycler](#).

Here is the call graph for this function:



6.62.2.91 MM::VOID Machine::reflect (MM::Program * program)

Here is the call graph for this function:



6.62.2.92 MM::VOID Machine::removeObserver (MM::UINT32 observer)

6.62.2.93 MM::VOID MM::Machine::reset ()

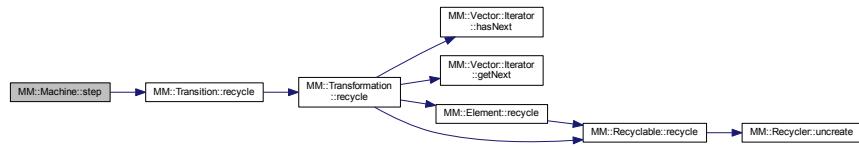
6.62.2.94 MM::VOID MM::Machine::reset (MM::UINT32 instance)

6.62.2.95 MM::VOID MM::Machine::setDefinition (MM::Definition * def)

6.62.2.96 MM::VOID MM::Machine::setInstance (MM::Instance * inst)

6.62.2.97 MM::VOID Machine::step()

Here is the call graph for this function:



Here is the caller graph for this function:



6.62.2.98 MM::VOID MM::Machine::step(MM::CHAR * buf, MM::UINT32 size)

6.62.2.99 MM::VOID MM::Machine::step(MM::UINT32 instance, MM::CHAR * buf, MM::UINT32 size)

6.62.3 Member Data Documentation

6.62.3.1 MM::Vector<MM::Machine::Delegate *>* MM::Machine::delegates

6.62.3.2 const MM::UINT32 Machine::LOG_SIZE = 1024 * 32 * 8 [static]

The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Machine.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Machine.cpp](#)

6.63 MM::Map< MAP_KEY, MAP_VALUE, COMPARE > Class Template Reference

```
#include <Map.h>
```

Classes

- class [Iterator](#)

Public Member Functions

- [Map\(\)](#)
- [Map\(MM::Map< MAP_KEY, MAP_VALUE, COMPARE > *map\)](#)

- `~Map ()`
- virtual `MM::TID getTypeId ()`
- virtual `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::VOID put (MAP_KEY k, MAP_VALUE v)`
- `MM::VOID putAll (MM::Map< MAP_KEY, MAP_VALUE, COMPARE > *other)`
- `MAP_VALUE get (MAP_KEY k)`
- `MM::VOID remove (MAP_KEY k)`
- `MM::BOOLEAN contains (MAP_KEY k)`
- `MM::VOID clear ()`
- `MM::BOOLEAN isEmpty ()`
- `MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::Iterator getIterator ()`

6.63.1 Constructor & Destructor Documentation

6.63.1.1 `template<class MAP_KEY, class MAP_VALUE, class COMPARE = std::less<MAP_KEY>> MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::Map () [inline]`

6.63.1.2 `template<class MAP_KEY, class MAP_VALUE, class COMPARE = std::less<MAP_KEY>> MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::Map (MM::Map< MAP_KEY, MAP_VALUE, COMPARE > * map) [inline]`

6.63.1.3 `template<class MAP_KEY, class MAP_VALUE, class COMPARE = std::less<MAP_KEY>> MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::~Map () [inline]`

6.63.2 Member Function Documentation

6.63.2.1 `template<class MAP_KEY, class MAP_VALUE, class COMPARE = std::less<MAP_KEY>> MM::VOID MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::clear () [inline]`

Here is the caller graph for this function:

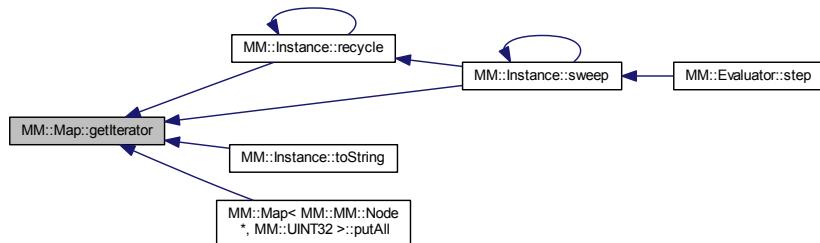


6.63.2.2 `template<class MAP_KEY, class MAP_VALUE, class COMPARE = std::less<MAP_KEY>> MM::BOOLEAN MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::contains (MAP_KEY k) [inline]`

6.63.2.3 `template<class MAP_KEY, class MAP_VALUE, class COMPARE = std::less<MAP_KEY>> MAP_VALUE MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::get (MAP_KEY k) [inline]`

6.63.2.4 template<class MAP_KEY, class MAP_VALUE, class COMPARE = std::less<MAP_KEY>> MM::Map<MAP_KEY, MAP_VALUE, COMPARE>::Iterator MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::getIterator () [inline]

Here is the caller graph for this function:



6.63.2.5 template<class MAP_KEY, class MAP_VALUE, class COMPARE = std::less<MAP_KEY>> virtual MM::TID MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::getTypeId () [inline], [virtual]

6.63.2.6 template<class MAP_KEY, class MAP_VALUE, class COMPARE = std::less<MAP_KEY>> virtual MM::BOOLEAN MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::instanceof(MM::TID tid) [inline], [virtual]

6.63.2.7 template<class MAP_KEY, class MAP_VALUE, class COMPARE = std::less<MAP_KEY>> MM::BOOLEAN MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::isEmpty () [inline]

6.63.2.8 template<class MAP_KEY, class MAP_VALUE, class COMPARE = std::less<MAP_KEY>> MM::VOID MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::put(MAP_KEY k, MAP_VALUE v) [inline]

Here is the caller graph for this function:



6.63.2.9 template<class MAP_KEY, class MAP_VALUE, class COMPARE = std::less<MAP_KEY>> MM::VOID MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::putAll(MM::Map< MAP_KEY, MAP_VALUE, COMPARE > * other) [inline]

6.63.2.10 template<class MAP_KEY, class MAP_VALUE, class COMPARE = std::less<MAP_KEY>> MM::VOID MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::remove(MAP_KEY k) [inline]

The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Map.h

6.64 Map Class Reference

The [Map](#) abstraction enables quickly looking up stored values using a storage key.

```
#include <Map.h>
```

6.64.1 Detailed Description

The [Map](#) abstraction enables quickly looking up stored values using a storage key.

Note

Currently wraps STL, we intend to remove this dependency.

The [Map](#) class hides internal details of the STL implementation of std::map

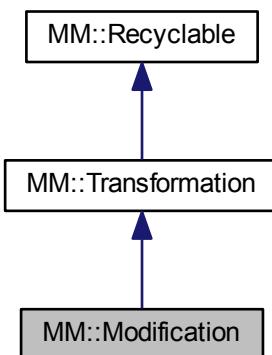
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Map.h](#)

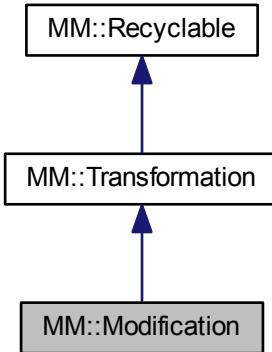
6.65 MM::Modification Class Reference

```
#include <Modification.h>
```

Inheritance diagram for MM::Modification:



Collaboration diagram for MM::Modification:



Public Member Functions

- `Modification (MM::Vector< MM::Element * > *elements)`
- `Modification (MM::Vector< MM::Element * > *elements, MM::Location *loc)`
- `~Modification ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::VOID toString (MM::String *buf)`

Additional Inherited Members

6.65.1 Constructor & Destructor Documentation

6.65.1.1 `Modification::Modification (MM::Vector< MM::Element * > * elements)`

modify source location

6.65.1.2 `Modification::Modification (MM::Vector< MM::Element * > * elements, MM::Location * loc)`

6.65.1.3 `Modification::~Modification ()`

6.65.2 Member Function Documentation

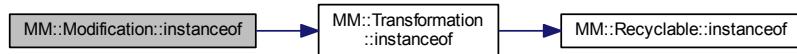
6.65.2.1 `MM::TID Modification::getTypeld () [virtual]`

Implements [MM::Transformation](#).

6.65.2.2 `MM::BOOLEAN Modification::instanceof (MM::TID tid) [virtual]`

Reimplemented from [MM::Recyclable](#).

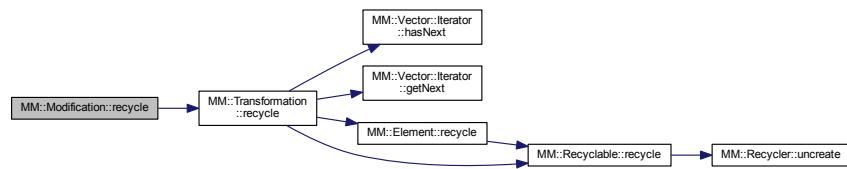
Here is the call graph for this function:



6.65.2.3 MM::VOID Modification::recycle (MM::Recycler * r) [virtual]

Reimplemented from [MM::Recyclable](#).

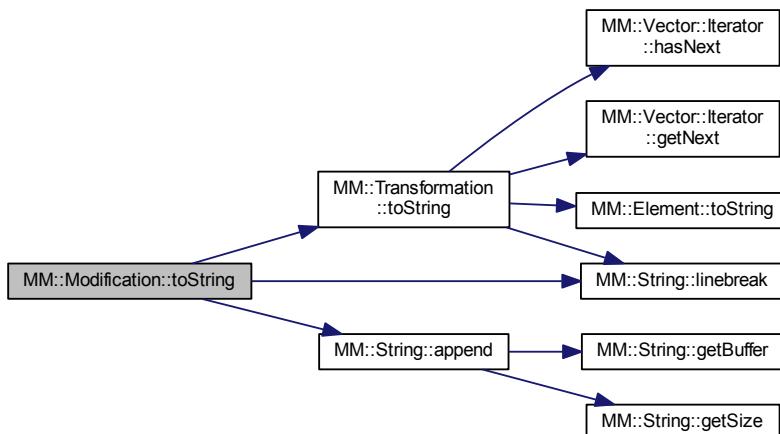
Here is the call graph for this function:



6.65.2.4 MM::VOID Modification::toString (MM::String * buf) [virtual]

Implements [MM::Recyclable](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Modification.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Modification.cpp](#)

6.66 Modification Class Reference

The [Modification](#) abstraction consists of elements that are modifications made by the reflector to definitions between steps.

```
#include <Modification.h>
```

6.66.1 Detailed Description

The [Modification](#) abstraction consists of elements that are modifications made by the reflector to definitions between steps.

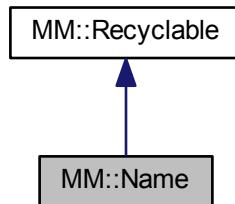
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Modification.h](#)

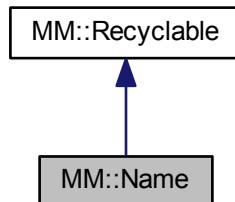
6.67 MM::Name Class Reference

```
#include <Name.h>
```

Inheritance diagram for MM::Name:



Collaboration diagram for MM::Name:



Classes

- class [Compare](#)

Public Member Functions

- [Name \(MM::CHAR *str, MM::UINT32 len\)](#)
- [MM::VOID recycle \(MM::Recycler *r\)](#)
- [MM::TID getTypeld \(\)](#)
Retrieves the type id of a [Name](#) object.
- [MM::BOOLEAN instanceof \(MM::TID tid\)](#)
Assesses if an object is an instance of a type tid.
- [MM::BOOLEAN equals \(MM::Name *name\)](#)
- [MM::BOOLEAN equals \(MM::CHAR *str, MM::UINT32 len\)](#)
- [MM::UINT32 getLength \(\)](#)
Retrieves the length of a [Name](#) object.
- [MM::VOID setName \(MM::Name *name\)](#)
Stores the sub name in this name.
- [MM::Name * getName \(\)](#)
Retrieves the sub name of this name.
- [MM::VOID setPreName \(MM::Name *name\)](#)
Stores a prefix name in this name.
- [MM::Name * getPreName \(\)](#)
- [MM::VOID append \(MM::Name *name\)](#)
- [MM::VOID print \(\)](#)
Prints a name on the screen.
- [MM::CHAR * getBuffer \(\)](#)
- [MM::Location * getLocation \(\)](#)
Retrieves the location of this name.
- [MM::VOID setLocation \(MM::Location *loc\)](#)
Stores a location in this name.
- [MM::VOID toString \(MM::String *buf\)](#)
Serializes a [Name](#) object into a [String](#) buffer.

Protected Member Functions

- [~Name \(\)](#)

Friends

- [MM::BOOLEAN operator< \(const Name &n1, const Name &n2\)](#)

6.67.1 Constructor & Destructor Documentation

6.67.1.1 Name::~Name () [protected]

prefix name (used only for edges)

Destructs a [Name](#) object.

6.67.1.2 Name::Name (MM::CHAR * str, MM::UINT32 len)

Constructs a [Name](#) object.

Parameters

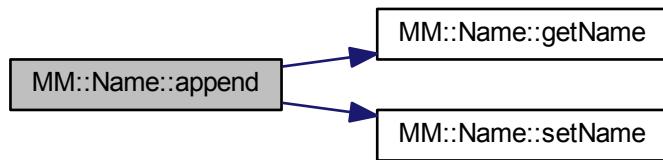
<i>str</i>	character buffer
<i>len</i>	character buffer length

Returnsnew [Name](#) object

6.67.2 Member Function Documentation

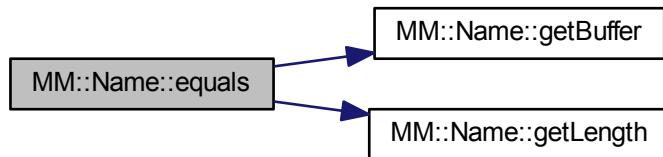
6.67.2.1 MM::VOID Name::append (MM::Name * *name*)

Here is the call graph for this function:



6.67.2.2 MM::BOOLEAN Name::equals (MM::Name * *name*)

Here is the call graph for this function:



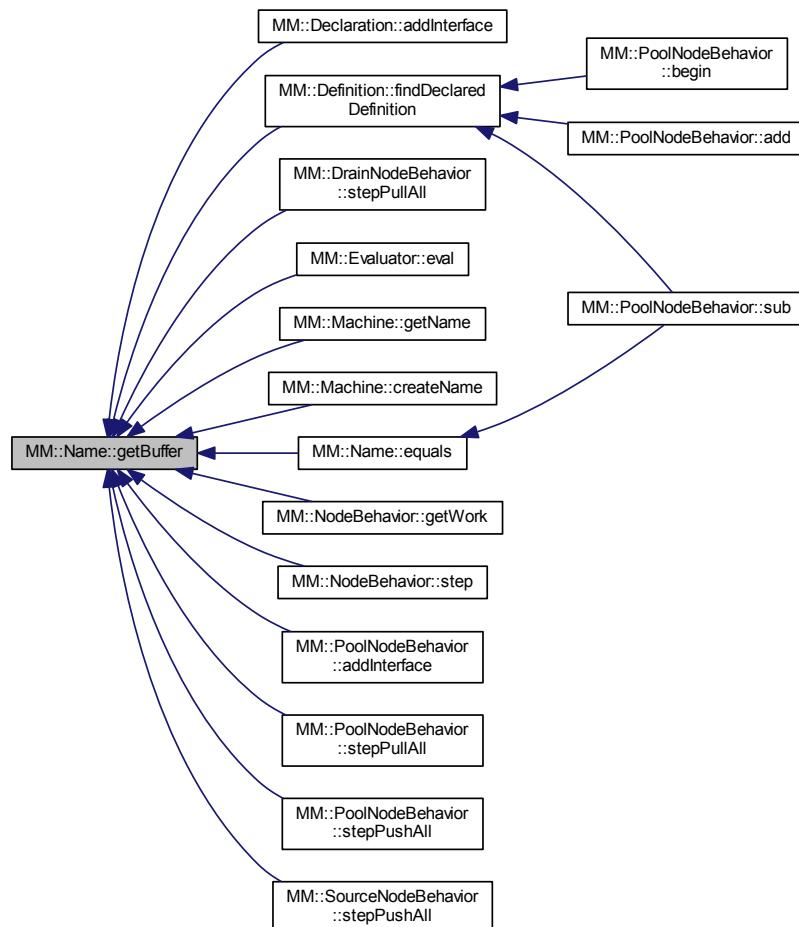
Here is the caller graph for this function:



6.67.2.3 MM::BOOLEAN Name::equals (MM::CHAR * str, MM::UINT32 len)

6.67.2.4 MM::CHAR * Name::getBuffer ()

Here is the caller graph for this function:



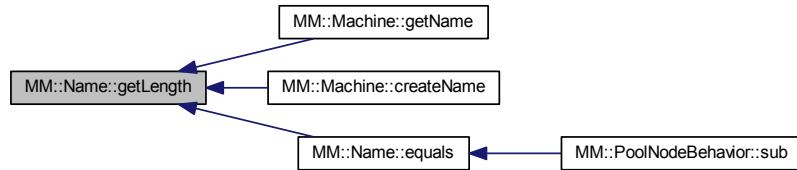
6.67.2.5 MM::UINT32 Name::getLength()

Retrieves the length of a [Name](#) object.

Returns

column number

Here is the caller graph for this function:



6.67.2.6 MM::Location * Name::getLocation()

Retrieves the location of this name.

Retrieves the prefix name of this name.

Returns

location
prefix name

Note

used for naming edges

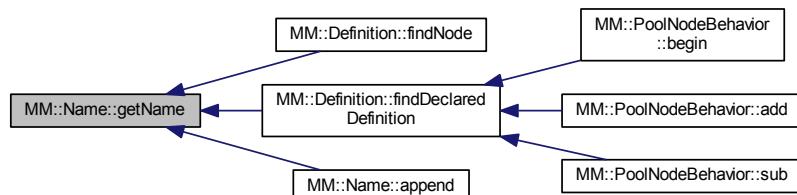
6.67.2.7 MM::Name * Name::getName()

Retrieves the sub name of this name.

Returns

sub name

Here is the caller graph for this function:



6.67.2.8 **MM::Name *** Name::getPreName()

6.67.2.9 **MM::TID** Name::getTypeId() [virtual]

Retrieves the type id of a [Name](#) object.

Returns

type id

Reimplemented from [MM::Recyclable](#).

6.67.2.10 **MM::BOOLEAN** Name::instanceof(**MM::TID** tid) [virtual]

Assesses if an object is an instance of a type tid.

Parameters

<i>tid</i>	type id
------------	---------

Returns

MM_TRUE if this object is instance of tid, MM_FALSE otherwise

Reimplemented from [MM::Recyclable](#).

Here is the call graph for this function:



6.67.2.11 **MM::VOID** Name::print()

Prints a name on the screen.

Note

debug only

Here is the caller graph for this function:



6.67.2.12 MM::VOID Name::recycle(MM::Recycler * r) [virtual]

Recycles a [Name](#) object in a [Recycler](#).

Parameters

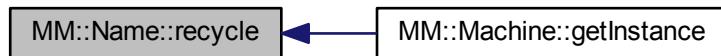
<i>r</i>	Recycler object
----------	-----------------

Reimplemented from [MM::Recyclable](#).

Here is the call graph for this function:



Here is the caller graph for this function:

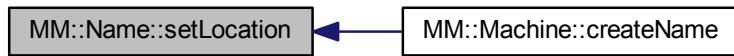
**6.67.2.13 MM::VOID Name::setLocation (MM::Location * *loc*)**

Stores a location in this name.

Parameters

<i>loc</i>	location
------------	----------

Here is the caller graph for this function:

**6.67.2.14 MM::VOID Name::setName (MM::Name * *name*)**

Stores the sub name in this name.

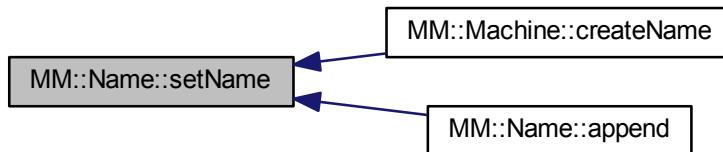
Parameters

<i>name</i>	sub name
-------------	----------

Returns

column number

Here is the caller graph for this function:

**6.67.2.15 MM::VOID Name::setPreName (MM::Name * *name*)**

Stores a prefix name in this name.

Parameters

<i>name</i>	to be stored as prefix
-------------	------------------------

Note

used for naming edges

Here is the caller graph for this function:

**6.67.2.16 MM::VOID Name::toString (MM::String * *buf*) [virtual]**

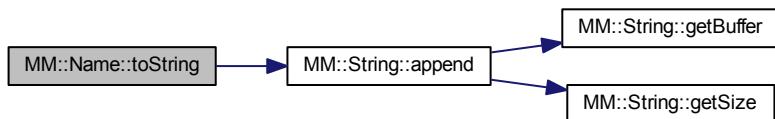
Serializes a [Name](#) object into a [String](#) buffer.

Parameters

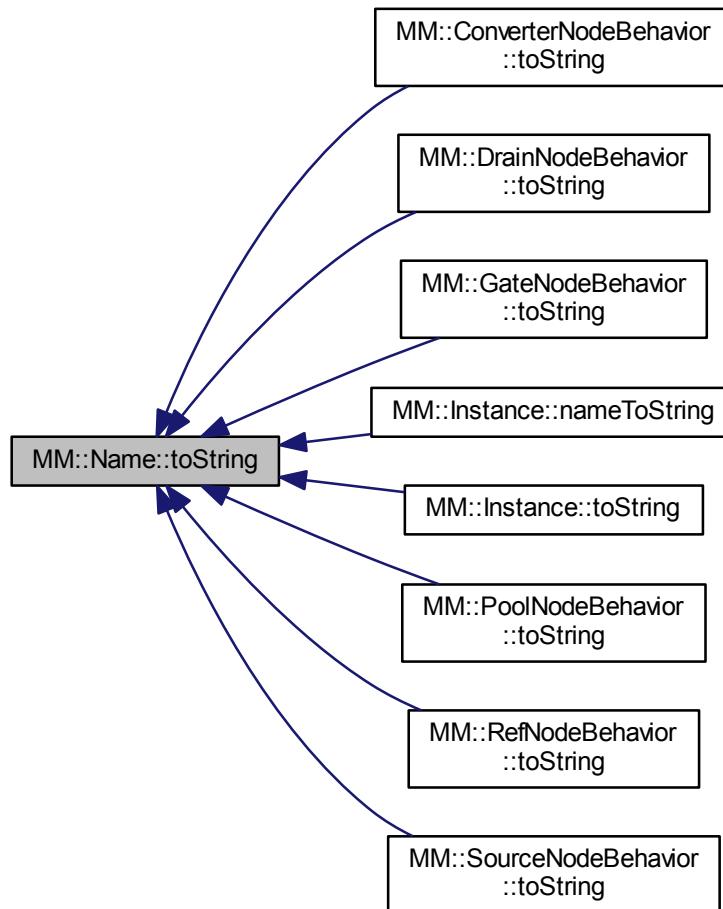
<code>buf</code>	String buffer to serialize this object into
------------------	---

Implements [MM::Recyclable](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.67.3 Friends And Related Function Documentation

6.67.3.1 MM::BOOLEAN operator< (const Name & n1, const Name & n2) [friend]

The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Name.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Name.cpp](#)

6.68 Name Class Reference

The [Name](#) abstraction is used to compose identifiers into qualified names used to lookup named language elements.

```
#include <Name.h>
```

6.68.1 Detailed Description

The [Name](#) abstraction is used to compose identifiers into qualified names used to lookup named language elements.

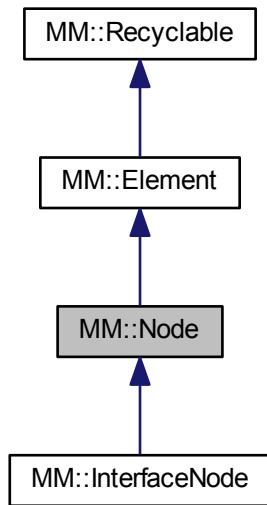
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Name.h](#)

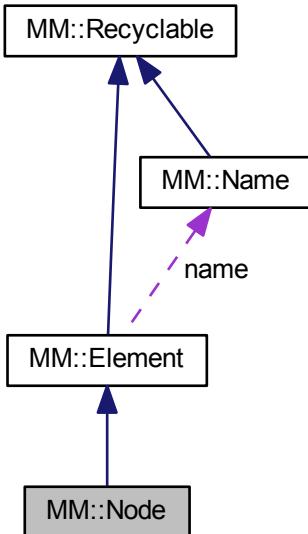
6.69 MM::Node Class Reference

```
#include <Node.h>
```

Inheritance diagram for MM::Node:



Collaboration diagram for MM::Node:



Classes

- class [Compare](#)

Public Member Functions

- [Node \(MM::Name *name, MM::NodeBehavior *behavior\)](#)
- [~Node \(\)](#)
- [MM::VOID recycle \(MM::Recycler *r\)](#)
- [virtual MM::TID getTypeId \(\)](#)
- [virtual MM::BOOLEAN instanceof \(MM::TID tid\)](#)
- [MM::VOID setEdgeOwnership \(MM::BOOLEAN isOwner\)](#)
- [MM::BOOLEAN hasEdgeOwnership \(\)](#)
- [virtual MM::NodeBehavior * getBehavior \(\)](#)
- [MM::VOID setBehavior \(MM::NodeBehavior *behavior\)](#)
- [MM::VOID addInput \(MM::Edge *edge\)](#)
- [MM::VOID addOutput \(MM::Edge *edge\)](#)
- [MM::VOID addCondition \(MM::Edge *edge\)](#)
- [MM::VOID addTrigger \(MM::Edge *edge\)](#)
- [MM::VOID addAlias \(MM::Edge *edge\)](#)
- [MM::VOID removeInput \(MM::Edge *edge\)](#)
- [MM::VOID removeOutput \(MM::Edge *edge\)](#)
- [MM::VOID removeCondition \(MM::Edge *edge\)](#)
- [MM::VOID removeTrigger \(MM::Edge *edge\)](#)
- [MM::VOID removeAlias \(MM::Edge *alias\)](#)
- [MM::Vector< Edge * > * getInput \(\)](#)
- [MM::Vector< Edge * > * getOutput \(\)](#)
- [MM::Vector< Edge * > * getConditions \(\)](#)

- `MM::Vector< Edge * > * getTriggers ()`
- `MM::Vector< Edge * > * getAliases ()`
- `MM::VOID setInput (MM::Vector< MM::Edge * > *input)`
- `MM::VOID setOutput (MM::Vector< MM::Edge * > *output)`
- `MM::VOID setConditions (MM::Vector< MM::Edge * > *conditions)`
- `MM::VOID setTriggers (MM::Vector< MM::Edge * > *triggers)`
- `MM::VOID setAliases (MM::Vector< MM::Edge * > *aliases)`
- `virtual MM::INT32 getAmount (MM::Instance *i, MM::Machine *m)`
- `virtual MM::VOID begin (MM::Instance *i, MM::Machine *m)`
- `virtual MM::VOID end (MM::Instance *i, MM::Machine *m)`
- `virtual MM::VOID change (MM::Instance *i, MM::Machine *m)`
- `virtual MM::VOID add (MM::Instance *i, MM::Machine *m, MM::UINT32 amount)`
- `virtual MM::VOID sub (MM::Instance *i, MM::Machine *m, MM::UINT32 amount)`
- `virtual MM::UINT32 getCapacity (MM::Instance *i)`
- `virtual MM::UINT32 getResources (MM::Instance *i)`
- `virtual MM::BOOLEAN hasCapacity (MM::Instance *i, MM::UINT32 amount)`
- `virtual MM::BOOLEAN hasResources (MM::Instance *i, MM::UINT32 amount)`
- `MM::VOID step (MM::Instance *i, MM::Machine *m, MM::Transition *tr)`
- `MM::BOOLEAN isDisabled (MM::Instance *i, MM::Evaluator *e, MM::Recycler *r)`
- `MM::BOOLEAN isSatisfied (MM::Instance *i, MM::Transition *tr)`
- `virtual MM::VOID activateTriggerTargets (MM::Instance *i, MM::Machine *m)`
- `MM::VOID toString (MM::String *buf)`
- `MM::VOID toString (MM::String *buf, MM::UINT32 indent)`

Additional Inherited Members

6.69.1 Constructor & Destructor Documentation

6.69.1.1 Node::Node (`MM::Name * name, MM::NodeBehavior * behavior`)

node behavior (strategy)

6.69.1.2 Node::~Node ()

6.69.2 Member Function Documentation

6.69.2.1 MM::VOID Node::activateTriggerTargets (`MM::Instance * i, MM::Machine * m`) [virtual]

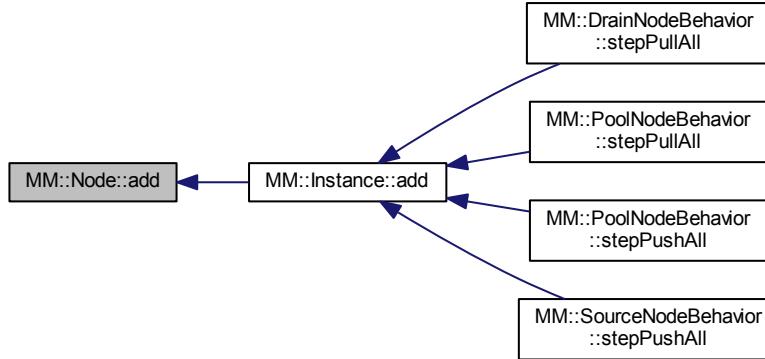
Reimplemented in [MM::InterfaceNode](#).

Here is the caller graph for this function:



6.69.2.2 **MM::VOID Node::add (MM::Instance * *i*, MM::Machine * *m*, MM::UINT32 *amount*) [virtual]**

Here is the caller graph for this function:



6.69.2.3 **MM::VOID Node::addAlias (MM::Edge * *edge*)**

6.69.2.4 **MM::VOID Node::addCondition (MM::Edge * *edge*)**

6.69.2.5 **MM::VOID Node::addInput (MM::Edge * *edge*)**

6.69.2.6 **MM::VOID Node::addOutput (MM::Edge * *edge*)**

6.69.2.7 **MM::VOID Node::addTrigger (MM::Edge * *edge*)**

6.69.2.8 **MM::VOID Node::begin (MM::Instance * *i*, MM::Machine * *m*) [virtual]**

Here is the call graph for this function:

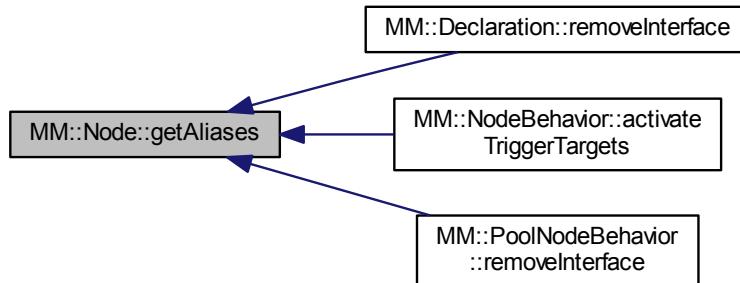


6.69.2.9 **MM::VOID Node::change (MM::Instance * *i*, MM::Machine * *m*) [virtual]**

6.69.2.10 **MM::VOID Node::end (MM::Instance * *i*, MM::Machine * *m*) [virtual]**

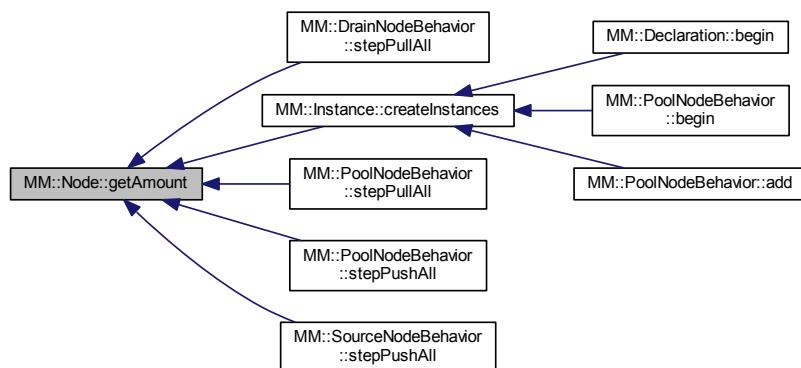
6.69.2.11 MM::Vector< MM::Edge * > * Node::getAliases()

Here is the caller graph for this function:



6.69.2.12 MM::INT32 Node::getAmount(MM::Instance * i, MM::Machine * m) [virtual]

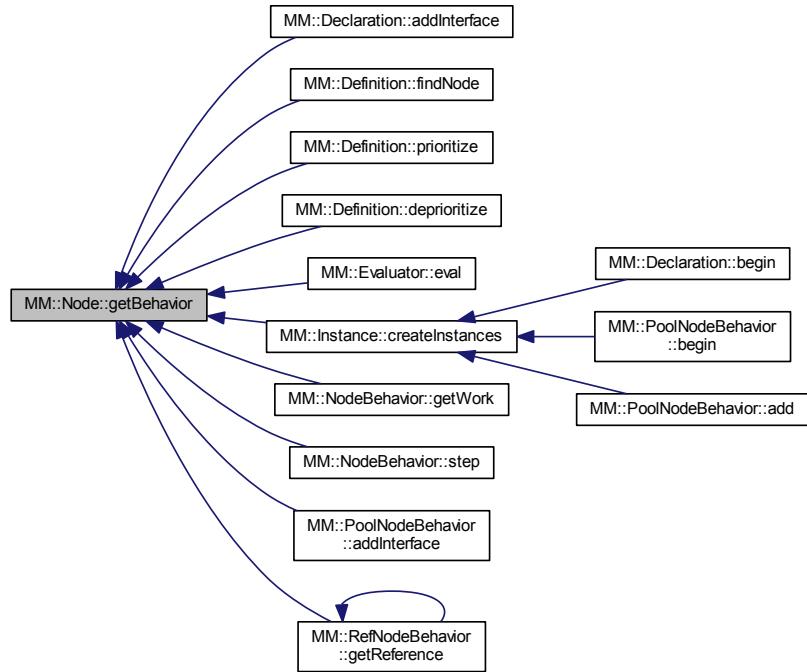
Here is the caller graph for this function:



6.69.2.13 MM::NodeBehavior * Node::getBehavior() [virtual]

Reimplemented in [MM::InterfaceNode](#).

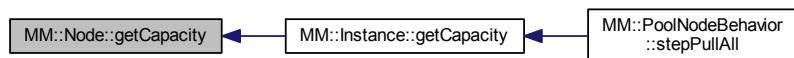
Here is the caller graph for this function:



6.69.2.14 `MM::UINT32 Node::getCapacity (MM::Instance * i) [virtual]`

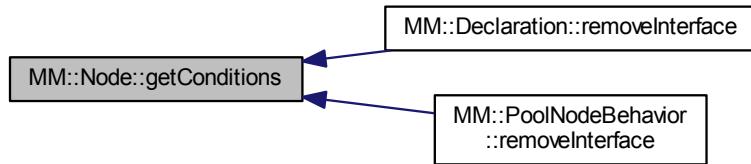
Reimplemented in [MM::InterfaceNode](#).

Here is the caller graph for this function:

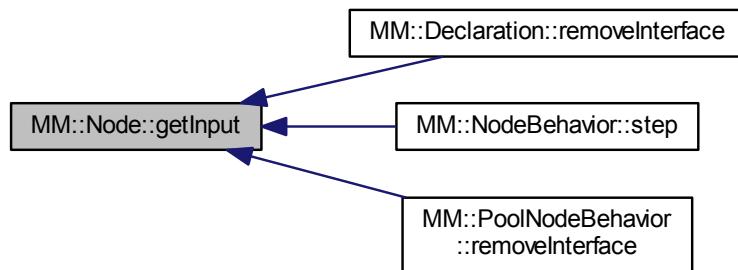


6.69.2.15 MM::Vector< MM::Edge * > * Node::getConditions()

Here is the caller graph for this function:

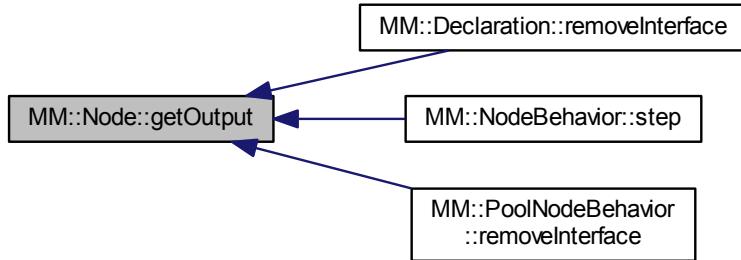
**6.69.2.16 MM::Vector< MM::Edge * > * Node::getInput()**

Here is the caller graph for this function:



6.69.2.17 `MM::Vector< MM::Edge * > * Node::getOutput()`

Here is the caller graph for this function:



6.69.2.18 `MM::UINT32 Node::getResources(MM::Instance * i) [virtual]`

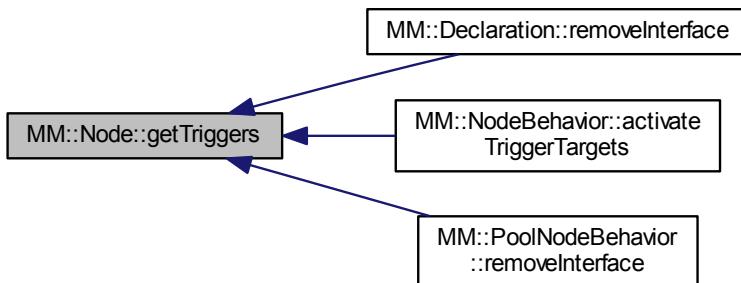
Reimplemented in [MM::InterfaceNode](#).

Here is the caller graph for this function:



6.69.2.19 `MM::Vector< MM::Edge * > * Node::getTriggers()`

Here is the caller graph for this function:



6.69.2.20 MM::TID Node::getTypeId() [virtual]

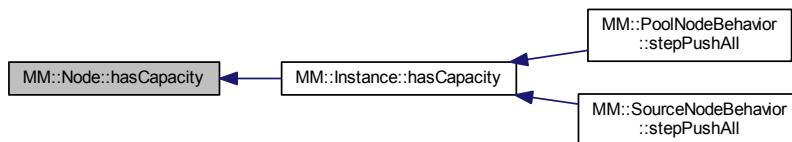
Reimplemented from [MM::Element](#).

Reimplemented in [MM::InterfaceNode](#).

6.69.2.21 MM::BOOLEAN Node::hasCapacity(MM::Instance * i, MM::UINT32 amount) [virtual]

Reimplemented in [MM::InterfaceNode](#).

Here is the caller graph for this function:

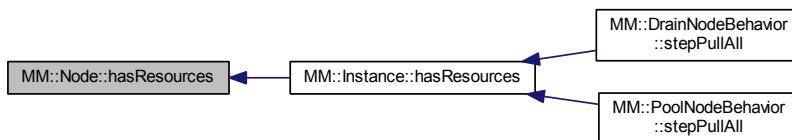


6.69.2.22 MM::BOOLEAN Node::hasEdgeOwnership()

6.69.2.23 MM::BOOLEAN Node::hasResources(MM::Instance * i, MM::UINT32 amount) [virtual]

Reimplemented in [MM::InterfaceNode](#).

Here is the caller graph for this function:



6.69.2.24 MM::BOOLEAN Node::instanceof(MM::TID tid) [virtual]

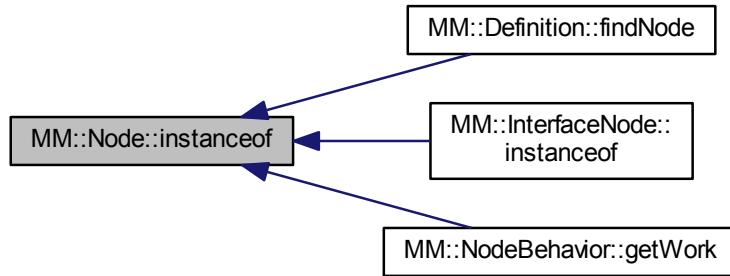
Reimplemented from [MM::Element](#).

Reimplemented in [MM::InterfaceNode](#).

Here is the call graph for this function:

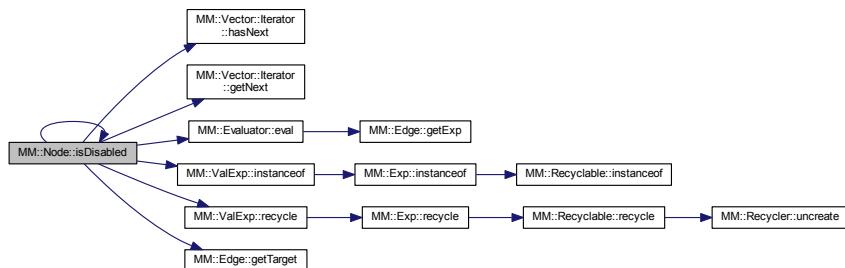


Here is the caller graph for this function:

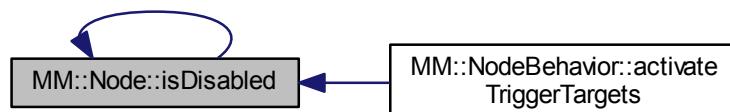


6.69.2.25 MM::BOOLEAN Node::isDisabled (MM::Instance * i, MM::Evaluator * e, MM::Recycler * r)

Here is the call graph for this function:

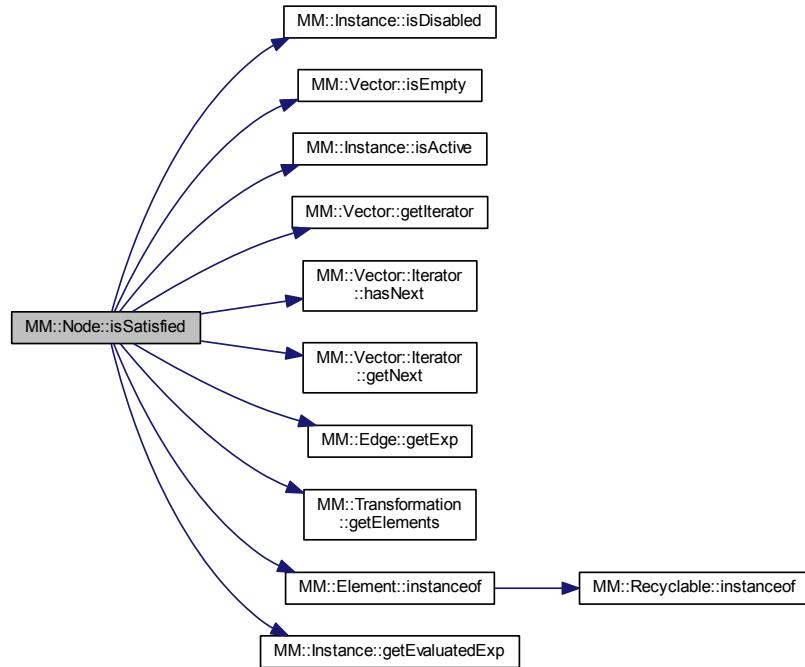


Here is the caller graph for this function:



6.69.2.26 MM::BOOLEAN Node::isSatisfied (MM::Instance * i, MM::Transition * tr)

Here is the call graph for this function:



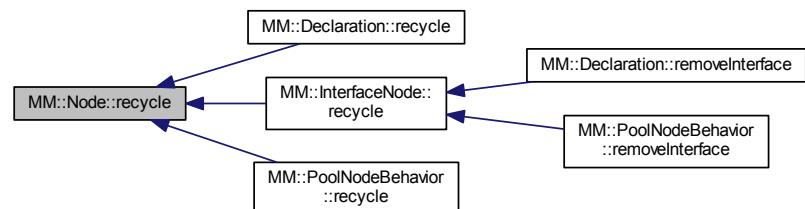
6.69.2.27 MM::VOID Node::recycle (MM::Recycler * r) [virtual]

Reimplemented from [MM::Element](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.69.2.28 **MM::VOID Node::removeAlias (MM::Edge * alias)**

6.69.2.29 **MM::VOID Node::removeCondition (MM::Edge * edge)**

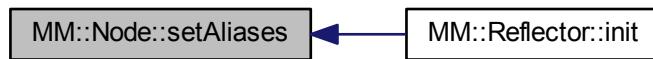
6.69.2.30 **MM::VOID Node::removeInput (MM::Edge * edge)**

6.69.2.31 **MM::VOID Node::removeOutput (MM::Edge * edge)**

6.69.2.32 **MM::VOID Node::removeTrigger (MM::Edge * edge)**

6.69.2.33 **MM::VOID Node::setAliases (MM::Vector< MM::Edge * > * aliases)**

Here is the caller graph for this function:



6.69.2.34 **MM::VOID Node::setBehavior (MM::NodeBehavior * behavior)**

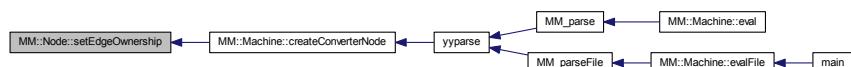
6.69.2.35 **MM::VOID Node::setConditions (MM::Vector< MM::Edge * > * conditions)**

Here is the caller graph for this function:



6.69.2.36 **MM::VOID Node::setEdgeOwnership (MM::BOOLEAN isOwner)**

Here is the caller graph for this function:



6.69.2.37 MM::VOID Node::setInput (MM::Vector< MM::Edge * > * *input*)

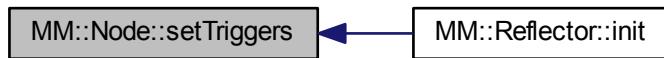
Here is the caller graph for this function:

**6.69.2.38 MM::VOID Node::setOutput (MM::Vector< MM::Edge * > * *output*)**

Here is the caller graph for this function:

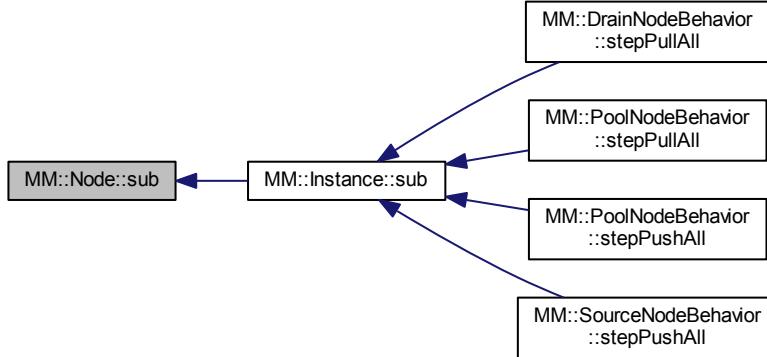
**6.69.2.39 MM::VOID Node::setTriggers (MM::Vector< MM::Edge * > * *triggers*)**

Here is the caller graph for this function:

**6.69.2.40 MM::VOID Node::step (MM::Instance * *i*, MM::Machine * *m*, MM::Transition * *tr*)**

6.69.2.41 MM::VOID Node::sub (MM::Instance * i, MM::Machine * m, MM::UINT32 amount) [virtual]

Here is the caller graph for this function:



6.69.2.42 MM::VOID Node::toString (MM::String * buf) [virtual]

Implements [MM::Element](#).

6.69.2.43 MM::VOID Node::toString (MM::String * buf, MM::UINT32 indent) [virtual]

Implements [MM::Element](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Node.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Node.cpp](#)

6.70 Node Class Reference

The [Node](#) abstraction defines program nodes that are connected by edges, and its behavior can be modified by changing its node behavior.

```
#include <Node.h>
```

6.70.1 Detailed Description

The [Node](#) abstraction defines program nodes that are connected by edges, and its behavior can be modified by changing its node behavior.

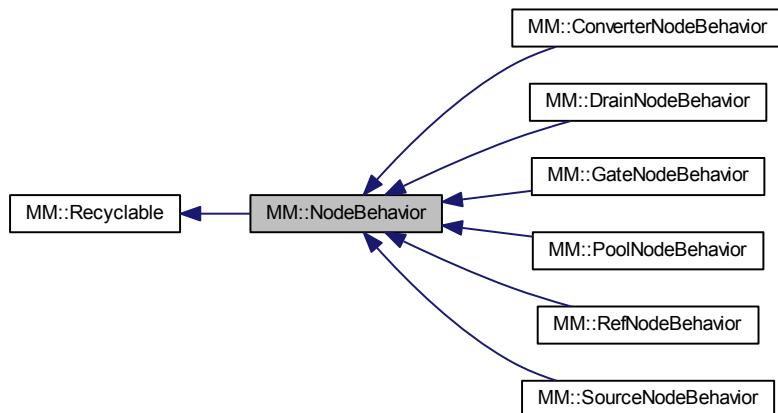
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Node.h](#)

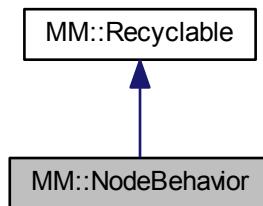
6.71 MM::NodeBehavior Class Reference

```
#include <NodeBehavior.h>
```

Inheritance diagram for MM::NodeBehavior:



Collaboration diagram for MM::NodeBehavior:



Public Types

- enum `_IO` {
 `IO_ERROR`, `IO_PRIVATE`, `IO_IN`, `IO_OUT`,
`IO_INOUT` }

- enum `_When` {
 `WHEN_ERROR`, `WHEN_PASSIVE`, `WHEN_AUTO`, `WHEN_USER`,
`WHEN_START` }
- enum `_Act` { `ACT_ERROR`, `ACT_PULL`, `ACT_PUSH` }
- enum `_How` { `HOW_ERROR`, `HOW_ANY`, `HOW_ALL` }
- typedef enum `MM::NodeBehavior::_IO` `IO`
- typedef enum
`MM::NodeBehavior::_When` `When`
- typedef enum
`MM::NodeBehavior::_Act` `Act`
- typedef enum
`MM::NodeBehavior::_How` `How`

Public Member Functions

- virtual ~`NodeBehavior` ()
- `MM::VOID recycle (MM::Recycler *r)`
- virtual `MM::TID getTypeId ()`
- virtual `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::NodeBehavior::IO getIO ()`
- `MM::NodeBehavior::When getWhen ()`
- `MM::NodeBehavior::Act getAct ()`
- `MM::NodeBehavior::How getHow ()`
- virtual `MM::UINT32 getCreateMessage ()=0`
- virtual `MM::UINT32 getUpdateMessage ()=0`
- virtual `MM::UINT32 getDeleteMessage ()=0`
- `MM::VOID setIO (MM::NodeBehavior::IO io)`
- `MM::VOID setWhen (MM::NodeBehavior::When when)`
- `MM::VOID setAct (MM::NodeBehavior::Act act)`
- `MM::VOID setHow (MM::NodeBehavior::How how)`
- `MM::BOOLEAN conformsTo (MM::NodeBehavior::IO direction)`
- `MM::VOID getWork (MM::Node *node, MM::Instance *instance, MM::Edge *edge, MM::Vector< MM::NodeWorkItem * > *work)`
- virtual `MM::VOID step (MM::Node *n, MM::Instance *i, MM::Machine *m, MM::Transition *t)`
- virtual `MM::VOID stepPullAny (MM::Node *node, MM::Instance *i, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)`
- virtual `MM::VOID stepPushAny (MM::Node *node, MM::Instance *i, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)`
- virtual `MM::VOID stepPullAll (MM::Node *node, MM::Instance *i, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)=0`
- virtual `MM::VOID stepPushAll (MM::Node *node, MM::Instance *i, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)=0`
- virtual `MM::VOID begin (MM::Instance *i, MM::Machine *m, MM::Node *n)=0`
- virtual `MM::VOID end (MM::Instance *i, MM::Machine *m, MM::Node *n)=0`
- virtual `MM::VOID change (MM::Instance *i, MM::Machine *m, MM::Node *n)=0`
- virtual `MM::INT32 getAmount (MM::Instance *i, MM::Machine *m, MM::Node *n)`
- virtual `MM::VOID add (MM::Instance *i, MM::Machine *m, MM::Node *n, MM::UINT32 amount)=0`
- virtual `MM::VOID sub (MM::Instance *i, MM::Machine *m, MM::Node *n, MM::UINT32 amount)=0`
- virtual `MM::UINT32 getCapacity (MM::Instance *i, MM::Node *n)=0`
- virtual `MM::UINT32 getResources (MM::Instance *i, MM::Node *n)=0`
- virtual `MM::BOOLEAN hasCapacity (MM::Instance *i, MM::Node *n, MM::UINT32 amount)=0`
- virtual `MM::BOOLEAN hasResources (MM::Instance *i, MM::Node *n, MM::UINT32 amount)=0`
- virtual `MM::VOID activateTriggerTargets (MM::Node *, MM::Instance *i, MM::Machine *m)`
- virtual `MM::VOID toString (MM::String *buf, MM::Name *name)`

Static Public Attributes

- static const MM::CHAR * IO_STR []
- static const MM::CHAR * WHEN_STR []
- static const MM::CHAR * ACT_STR []
- static const MM::CHAR * HOW_STR []
- static const MM::UINT32 IO_LEN []
- static const MM::UINT32 WHEN_LEN []
- static const MM::UINT32 ACT_LEN []
- static const MM::UINT32 HOW_LEN []

Protected Member Functions

- `NodeBehavior (MM::NodeBehavior::IO io, MM::NodeBehavior::When when, MM::NodeBehavior::Act act, MM::NodeBehavior::How how)`

6.71.1 Member Typedef Documentation

6.71.1.1 `typedef enum MM::NodeBehavior::__Act MM::NodeBehavior::Act`

6.71.1.2 `typedef enum MM::NodeBehavior::__How MM::NodeBehavior::How`

6.71.1.3 `typedef enum MM::NodeBehavior::__IO MM::NodeBehavior::IO`

6.71.1.4 `typedef enum MM::NodeBehavior::__When MM::NodeBehavior::When`

6.71.2 Member Enumeration Documentation

6.71.2.1 `enum MM::NodeBehavior::__Act`

Enumerator

`ACT_ERROR`

`ACT_PULL`

`ACT_PUSH`

6.71.2.2 `enum MM::NodeBehavior::__How`

Enumerator

`HOW_ERROR`

`HOW_ANY`

`HOW_ALL`

6.71.2.3 `enum MM::NodeBehavior::__IO`

Enumerator

`IO_ERROR`

`IO_PRIVATE`

`IO_IN`

`IO_OUT`

`IO_INOUT`

6.71.2.4 enum MM::NodeBehavior::__When

Enumerator

WHEN_ERROR
WHEN_PASSIVE
WHEN_AUTO
WHEN_USER
WHEN_START

6.71.3 Constructor & Destructor Documentation

6.71.3.1 NodeBehavior::NodeBehavior (MM::NodeBehavior::IO *io*, MM::NodeBehavior::When *when*, MM::NodeBehavior::Act *act*, MM::NodeBehavior::How *how*) [protected]

the how modifier determines how a node performs its act

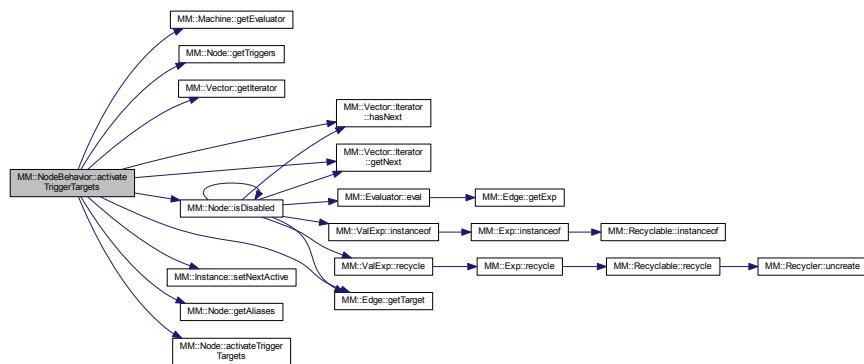
6.71.3.2 NodeBehavior::~NodeBehavior () [virtual]

6.71.4 Member Function Documentation

6.71.4.1 MM::VOID NodeBehavior::activateTriggerTargets (MM::Node * *node*, MM::Instance * *i*, MM::Machine * *m*) [virtual]

Reimplemented in [MM::ConverterNodeBehavior](#).

Here is the call graph for this function:



6.71.4.2 virtual MM::VOID MM::NodeBehavior::add (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*, MM::UINT32 *amount*) [pure virtual]

Implemented in [MM::PoolNodeBehavior](#), [MM::ConverterNodeBehavior](#), [MM::RefNodeBehavior](#), [MM::DrainNodeBehavior](#), [MM::SourceNodeBehavior](#), and [MM::GateNodeBehavior](#).

6.71.4.3 virtual MM::VOID MM::NodeBehavior::begin (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*) [pure virtual]

Implemented in [MM::PoolNodeBehavior](#), [MM::ConverterNodeBehavior](#), [MM::RefNodeBehavior](#), [MM::DrainNodeBehavior](#), [MM::SourceNodeBehavior](#), and [MM::GateNodeBehavior](#).

6.71.4.4 virtual MM::VOID MM::NodeBehavior::change (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*)
 [pure virtual]

Implemented in [MM::PoolNodeBehavior](#), [MM::ConverterNodeBehavior](#), [MM::RefNodeBehavior](#), [MM::DrainNodeBehavior](#), [MM::SourceNodeBehavior](#), and [MM::GateNodeBehavior](#).

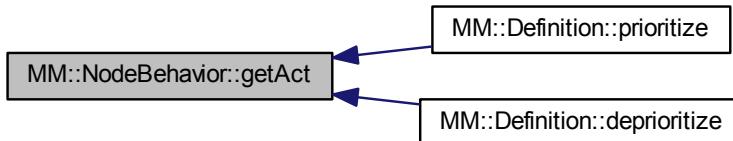
6.71.4.5 MM::BOOLEAN NodeBehavior::conformsTo (MM::NodeBehavior::IO *direction*)

6.71.4.6 virtual MM::VOID MM::NodeBehavior::end (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*) [pure virtual]

Implemented in [MM::PoolNodeBehavior](#), [MM::ConverterNodeBehavior](#), [MM::RefNodeBehavior](#), [MM::DrainNodeBehavior](#), [MM::SourceNodeBehavior](#), and [MM::GateNodeBehavior](#).

6.71.4.7 MM::NodeBehavior::Act NodeBehavior::getAct ()

Here is the caller graph for this function:



6.71.4.8 MM::INT32 NodeBehavior::getAmount (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*)
 [virtual]

Reimplemented in [MM::PoolNodeBehavior](#).

6.71.4.9 virtual MM::UINT32 MM::NodeBehavior::getCapacity (MM::Instance * *i*, MM::Node * *n*) [pure virtual]

Implemented in [MM::PoolNodeBehavior](#), [MM::ConverterNodeBehavior](#), [MM::RefNodeBehavior](#), [MM::DrainNodeBehavior](#), [MM::SourceNodeBehavior](#), and [MM::GateNodeBehavior](#).

6.71.4.10 virtual MM::UINT32 MM::NodeBehavior::getCreateMessage () [pure virtual]

Implemented in [MM::PoolNodeBehavior](#), [MM::ConverterNodeBehavior](#), [MM::RefNodeBehavior](#), [MM::GateNodeBehavior](#), [MM::DrainNodeBehavior](#), and [MM::SourceNodeBehavior](#).

Here is the caller graph for this function:

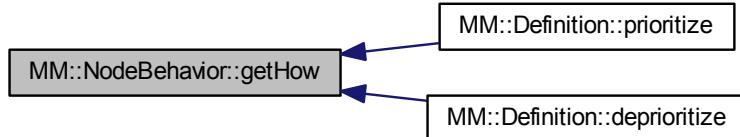


6.71.4.11 virtual MM::UINT32 MM::NodeBehavior::getDeleteMessage () [pure virtual]

Implemented in [MM::PoolNodeBehavior](#), [MM::ConverterNodeBehavior](#), [MM::RefNodeBehavior](#), [MM::GateNodeBehavior](#), [MM::DrainNodeBehavior](#), and [MM::SourceNodeBehavior](#).

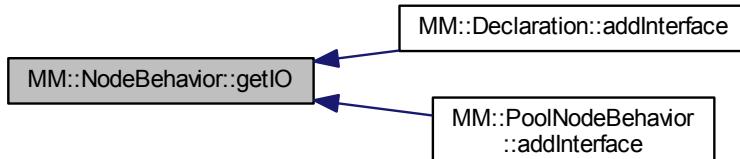
6.71.4.12 MM::NodeBehavior::How NodeBehavior::getHow ()

Here is the caller graph for this function:



6.71.4.13 MM::NodeBehavior::IO NodeBehavior::getIO ()

Here is the caller graph for this function:



6.71.4.14 virtual MM::UINT32 MM::NodeBehavior::getResources (MM::Instance * *i*, MM::Node * *n*) [pure virtual]

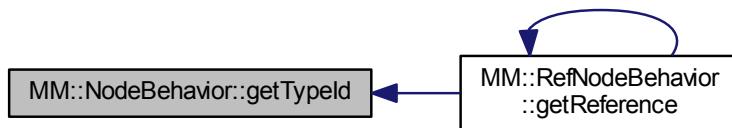
Implemented in [MM::PoolNodeBehavior](#), [MM::ConverterNodeBehavior](#), [MM::RefNodeBehavior](#), [MM::DrainNodeBehavior](#), [MM::SourceNodeBehavior](#), and [MM::GateNodeBehavior](#).

6.71.4.15 MM::TID NodeBehavior::getTypeId () [virtual]

Reimplemented from [MM::Recyclable](#).

Reimplemented in [MM::PoolNodeBehavior](#), [MM::ConverterNodeBehavior](#), [MM::GateNodeBehavior](#), [MM::DrainNodeBehavior](#), [MM::RefNodeBehavior](#), and [MM::SourceNodeBehavior](#).

Here is the caller graph for this function:



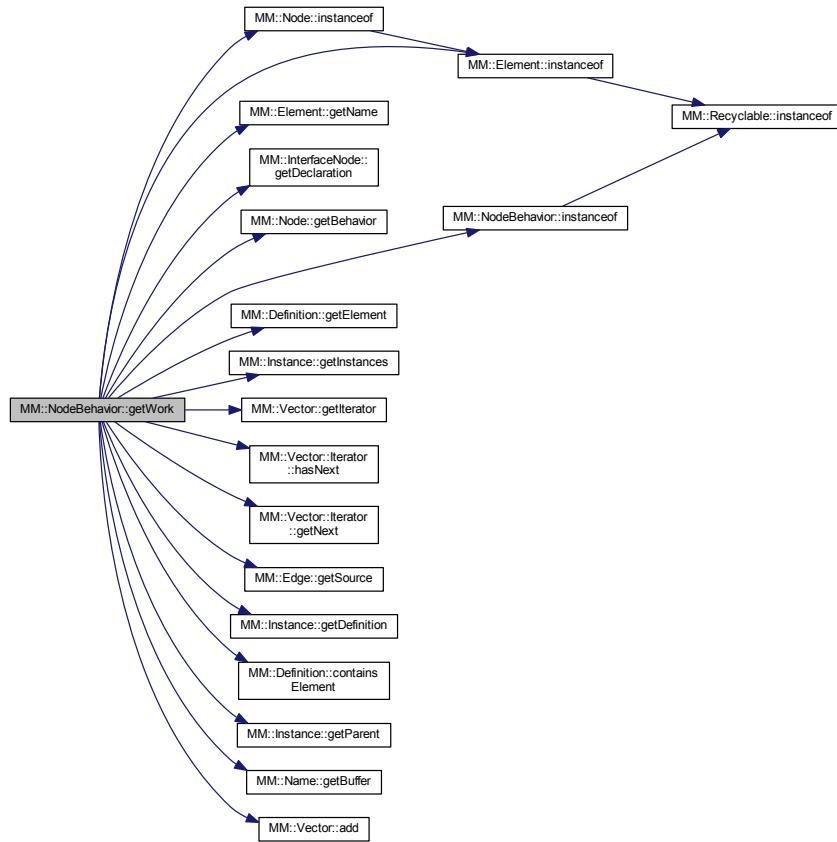
6.71.4.16 virtual MM::UINT32 MM::NodeBehavior::getUpdateMessage () [pure virtual]

Implemented in [MM::PoolNodeBehavior](#), [MM::ConverterNodeBehavior](#), [MM::RefNodeBehavior](#), [MM::GateNodeBehavior](#), [MM::DrainNodeBehavior](#), and [MM::SourceNodeBehavior](#).

6.71.4.17 MM::NodeBehavior::When NodeBehavior::getWhen ()

6.71.4.18 **MM::VOID NodeBehavior::getWork (MM::Node * node, MM::Instance * instance, MM::Edge * edge, MM::Vector< MM::NodeWorkItem * > * work)**

Here is the call graph for this function:



6.71.4.19 **virtual MM::BOOLEAN MM::NodeBehavior::hasCapacity (MM::Instance * i, MM::Node * n, MM::UINT32 amount) [pure virtual]**

Implemented in [MM::PoolNodeBehavior](#), [MM::ConverterNodeBehavior](#), [MM::RefNodeBehavior](#), [MM::DrainNodeBehavior](#), [MM::SourceNodeBehavior](#), and [MM::GateNodeBehavior](#).

6.71.4.20 **virtual MM::BOOLEAN MM::NodeBehavior::hasResources (MM::Instance * i, MM::Node * n, MM::UINT32 amount) [pure virtual]**

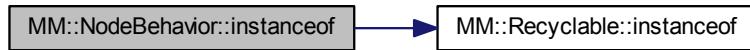
Implemented in [MM::PoolNodeBehavior](#), [MM::ConverterNodeBehavior](#), [MM::RefNodeBehavior](#), [MM::DrainNodeBehavior](#), [MM::SourceNodeBehavior](#), and [MM::GateNodeBehavior](#).

6.71.4.21 **MM::BOOLEAN NodeBehavior::instanceof (MM::TID tid) [virtual]**

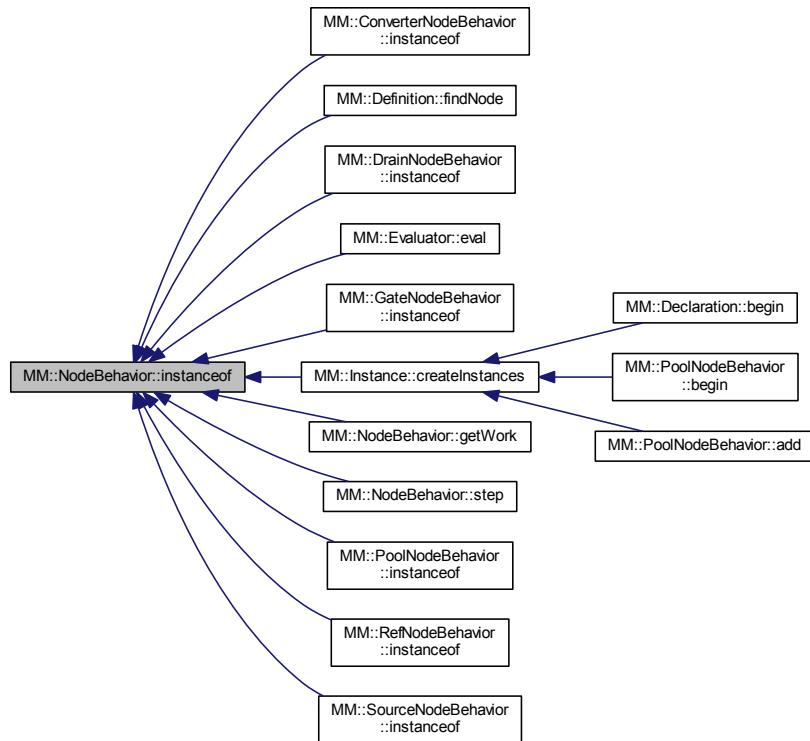
Reimplemented from [MM::Recyclable](#).

Reimplemented in [MM::PoolNodeBehavior](#), [MM::ConverterNodeBehavior](#), [MM::GateNodeBehavior](#), [MM::DrainNodeBehavior](#), [MM::RefNodeBehavior](#), and [MM::SourceNodeBehavior](#).

Here is the call graph for this function:



Here is the caller graph for this function:

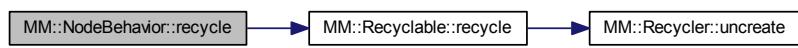


6.71.4.22 MM::VOID NodeBehavior::recycle (MM::Recycler * r) [virtual]

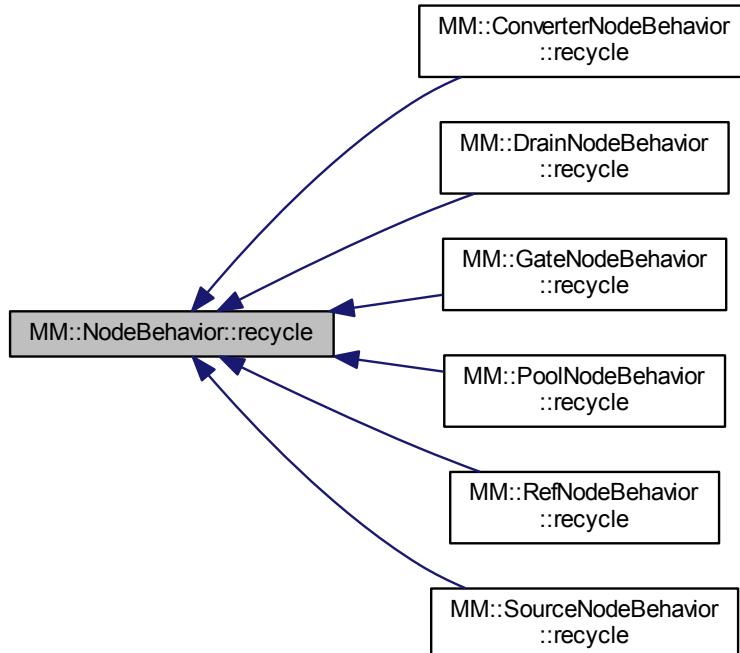
Reimplemented from [MM::Recyclable](#).

Reimplemented in [MM::PoolNodeBehavior](#), [MM::RefNodeBehavior](#), and [MM::SourceNodeBehavior](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.71.4.23 **MM::VOID NodeBehavior::setAct (MM::NodeBehavior::Act act)**

6.71.4.24 **MM::VOID NodeBehavior::setHow (MM::NodeBehavior::How how)**

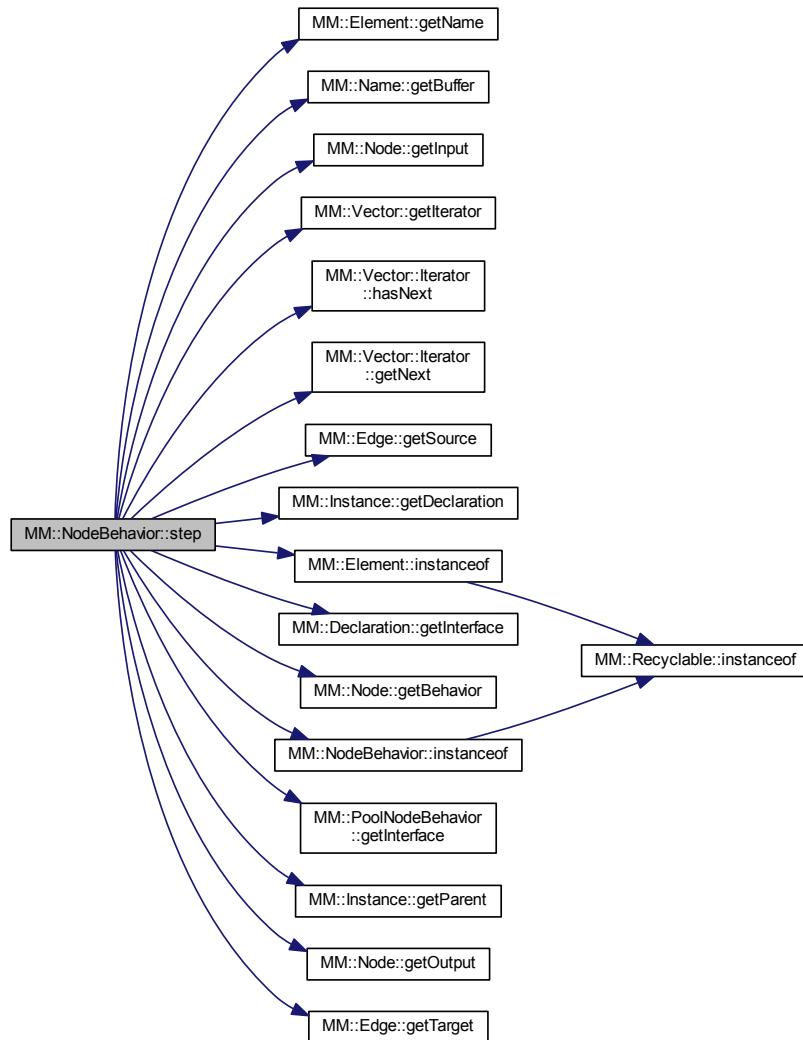
6.71.4.25 **MM::VOID NodeBehavior::setIO (MM::NodeBehavior::IO io)**

6.71.4.26 **MM::VOID NodeBehavior::setWhen (MM::NodeBehavior::When when)**

6.71.4.27 **MM::VOID NodeBehavior::step (MM::Node * n, MM::Instance * i, MM::Machine * m, MM::Transition * t) [virtual]**

Reimplemented in [MM::ConverterNodeBehavior](#), and [MM::RefNodeBehavior](#).

Here is the call graph for this function:



6.71.4.28 virtual MM::VOID MM::NodeBehavior::stepPullAll (MM::Node * *node*, MM::Instance * *i*, MM::Vector< MM::NodeWorkItem * > * *work*, MM::Machine * *m*, MM::Transition * *tr*) [pure virtual]

Implemented in [MM::PoolNodeBehavior](#), [MM::ConverterNodeBehavior](#), [MM::RefNodeBehavior](#), [MM::DrainNodeBehavior](#), [MM::SourceNodeBehavior](#), and [MM::GateNodeBehavior](#).

6.71.4.29 MM::VOID NodeBehavior::stepPullAny (MM::Node * *node*, MM::Instance * *i*, MM::Vector< MM::NodeWorkItem * > * *work*, MM::Machine * *m*, MM::Transition * *tr*) [virtual]

Reimplemented in [MM::ConverterNodeBehavior](#), [MM::RefNodeBehavior](#), and [MM::SourceNodeBehavior](#).

6.71.4.30 virtual MM::VOID MM::NodeBehavior::stepPushAll (MM::Node * *node*, MM::Instance * *i*, MM::Vector< MM::NodeWorkItem * > * *work*, MM::Machine * *m*, MM::Transition * *tr*) [pure virtual]

Implemented in [MM::PoolNodeBehavior](#), [MM::ConverterNodeBehavior](#), [MM::RefNodeBehavior](#), [MM::DrainNodeBehavior](#), [MM::SourceNodeBehavior](#), and [MM::GateNodeBehavior](#).

6.71.4.31 MM::VOID NodeBehavior::stepPushAny (MM::Node * *node*, MM::Instance * *i*, MM::Vector< MM::NodeWorkItem * > * *work*, MM::Machine * *m*, MM::Transition * *tr*) [virtual]

Reimplemented in [MM::ConverterNodeBehavior](#), [MM::RefNodeBehavior](#), and [MM::DrainNodeBehavior](#).

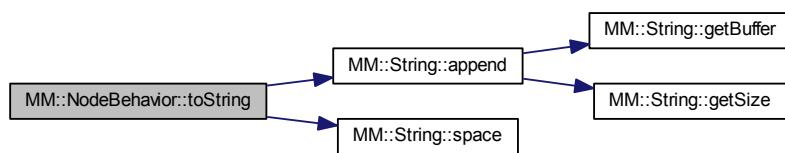
6.71.4.32 virtual MM::VOID MM::NodeBehavior::sub (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*, MM::UINT32 *amount*) [pure virtual]

Implemented in [MM::PoolNodeBehavior](#), [MM::ConverterNodeBehavior](#), [MM::RefNodeBehavior](#), [MM::DrainNodeBehavior](#), [MM::SourceNodeBehavior](#), and [MM::GateNodeBehavior](#).

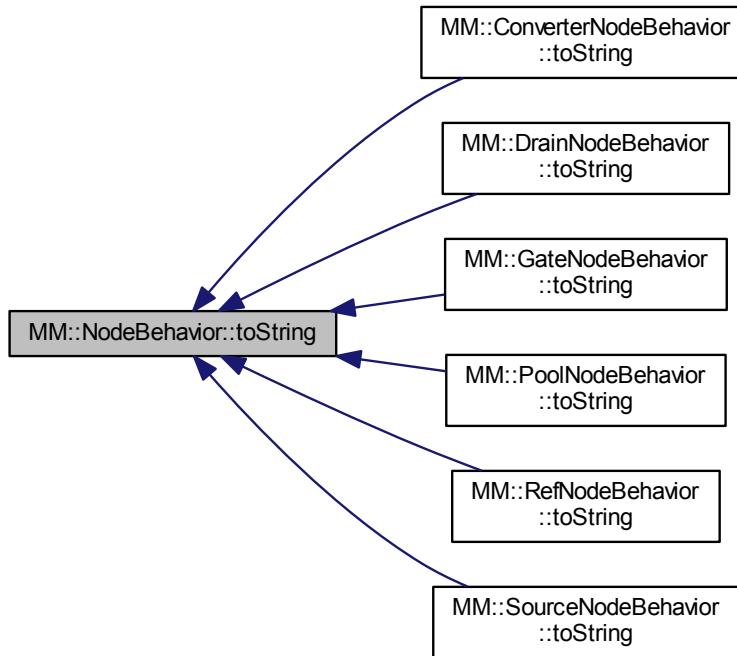
6.71.4.33 MM::VOID NodeBehavior::toString (MM::String * *buf*, MM::Name * *name*) [virtual]

Reimplemented in [MM::PoolNodeBehavior](#), [MM::ConverterNodeBehavior](#), [MM::RefNodeBehavior](#), [MM::DrainNodeBehavior](#), [MM::SourceNodeBehavior](#), and [MM::GateNodeBehavior](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.71.5 Member Data Documentation

6.71.5.1 const MM::UINT32 NodeBehavior::ACT_LEN [static]

Initial value:

```
=
{
  strlen(MM::NodeBehavior::ACT_STR[0]),
  strlen(MM::NodeBehavior::ACT_STR[1]),
  strlen(MM::NodeBehavior::ACT_STR[2])
}
```

6.71.5.2 const MM::CHAR * NodeBehavior::ACT_STR [static]

Initial value:

```
=
{
  "error",
  "",
  "push"
}
```

6.71.5.3 const MM::UINT32 NodeBehavior::HOW_LEN [static]

Initial value:

```
=
{
    strlen(MM::NodeBehavior::HOW_STR[0]),
    strlen(MM::NodeBehavior::HOW_STR[1]),
    strlen(MM::NodeBehavior::HOW_STR[2])
}
```

6.71.5.4 const MM::CHAR * NodeBehavior::HOW_STR [static]

Initial value:

```
=
{
    "error",
    "",
    "all"
}
```

6.71.5.5 const MM::UINT32 NodeBehavior::IO_LEN [static]

Initial value:

```
=
{
    strlen(MM::NodeBehavior::IO_STR[0]),
    strlen(MM::NodeBehavior::IO_STR[1]),
    strlen(MM::NodeBehavior::IO_STR[2]),
    strlen(MM::NodeBehavior::IO_STR[3]),
    strlen(MM::NodeBehavior::IO_STR[4])
}
```

6.71.5.6 const MM::CHAR * NodeBehavior::IO_STR [static]

Initial value:

```
=
{
    "error",
    "",
    "in",
    "out",
    "inout"
}
```

6.71.5.7 const MM::UINT32 NodeBehavior::WHEN_LEN [static]

Initial value:

```
=
{
    strlen(MM::NodeBehavior::WHEN_STR[0]),
    strlen(MM::NodeBehavior::WHEN_STR[1]),
    strlen(MM::NodeBehavior::WHEN_STR[2]),
    strlen(MM::NodeBehavior::WHEN_STR[3]),
    strlen(MM::NodeBehavior::WHEN_STR[4])
}
```

6.71.5.8 const MM::CHAR * NodeBehavior::WHEN_STR [static]

Initial value:

```
=
{
    "error",
    "",
    "auto",
    "user",
    "start"
}
```

The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[NodeBehavior.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[NodeBehavior.cpp](#)

6.72 NodeBehavior Class Reference

The [NodeBehavior](#) abstraction is the abstract super class of all node behaviors and makes node behavior a modifiable strategy.

```
#include <NodeBehavior.h>
```

6.72.1 Detailed Description

The [NodeBehavior](#) abstraction is the abstract super class of all node behaviors and makes node behavior a modifiable strategy.

The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[NodeBehavior.h](#)

6.73 MM::NodeWorkItem Class Reference

```
#include <NodeWorkItem.h>
```

Public Member Functions

- [NodeWorkItem \(MM::Instance *instance, MM::Node *node, MM::Edge *edge\)](#)
- [~NodeWorkItem \(\)](#)
- [MM::Instance * getInstance \(\)](#)
- [MM::Node * getNode \(\)](#)
- [MM::Edge * getEdge \(\)](#)

6.73.1 Constructor & Destructor Documentation

6.73.1.1 NodeWorkItem::NodeWorkItem ([MM::Instance * instance](#), [MM::Node * node](#), [MM::Edge * edge](#))

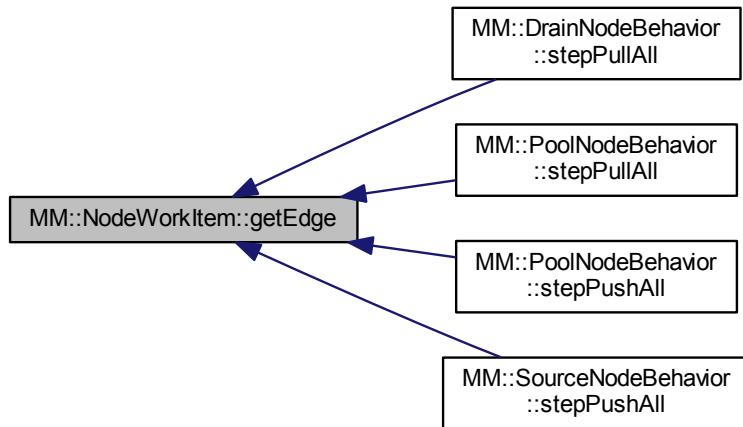
edge the node acts on

6.73.1.2 NodeWorkItem::~NodeWorkItem ()

6.73.2 Member Function Documentation

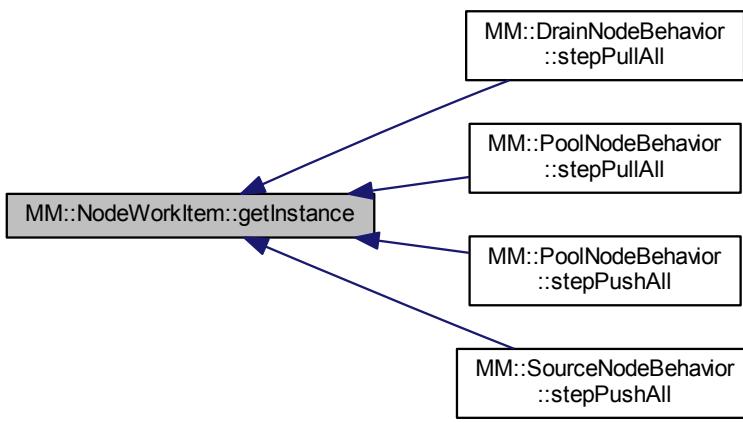
6.73.2.1 MM::Edge * NodeWorkItem::getEdge ()

Here is the caller graph for this function:



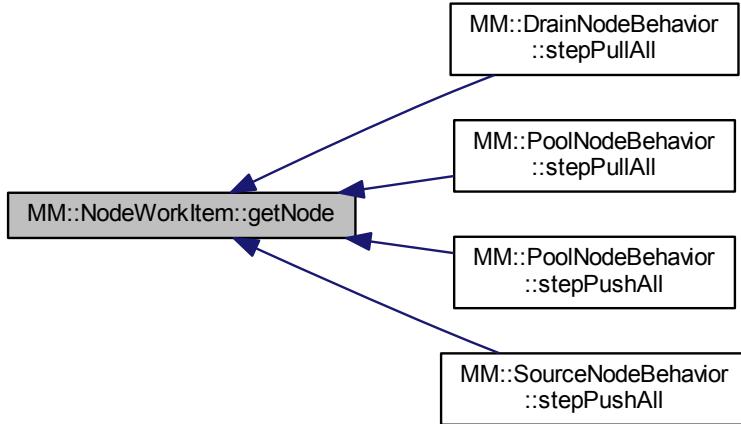
6.73.2.2 MM::Instance * NodeWorkItem::getInstance ()

Here is the caller graph for this function:



6.73.2.3 MM::Node * NodeWorkItem::getNode ()

Here is the caller graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[NodeWorkItem.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[NodeWorkItem.cpp](#)

6.74 NodeWorkItem Class Reference

The [NodeWorkItem](#) abstraction defines a piece of work to be acted out by a node in an instance on an edge.

```
#include <NodeWorkItem.h>
```

6.74.1 Detailed Description

The [NodeWorkItem](#) abstraction defines a piece of work to be acted out by a node in an instance on an edge.

The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[NodeWorkItem.h](#)

6.75 NumberValExp Class Reference

The [NumberValExp](#) abstraction defines number value expressions.

```
#include <NumberValExp.h>
```

6.75.1 Detailed Description

The [NumberValExp](#) abstraction defines number value expressions.

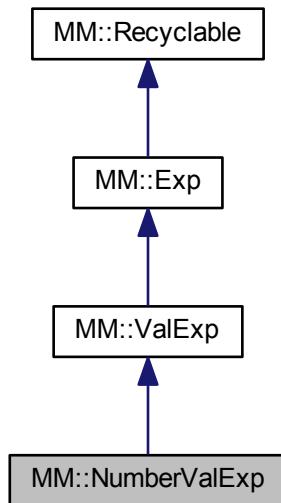
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[NumberValExp.h](#)

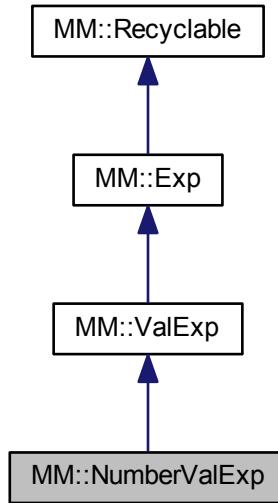
6.76 MM::NumberValExp Class Reference

```
#include <NumberValExp.h>
```

Inheritance diagram for MM::NumberValExp:



Collaboration diagram for MM::NumberValExp:



Public Member Functions

- [NumberValExp \(MM::INT32 val\)](#)
- [NumberValExp \(MM::INT32 val, MM::Location *loc\)](#)
- [NumberValExp \(MM::INT32 val, MM::UINT8 fraction\)](#)
- [NumberValExp \(MM::INT32 val, MM::UINT8 fraction, MM::Location *loc\)](#)
- [~NumberValExp \(\)](#)
- [MM::VOID recycle \(MM::Recycler *r\)](#)
- [MM::TID getTypeld \(\)](#)

Retrieves the type id of a [Exp](#) object.

- [MM::BOOLEAN instanceof \(MM::TID tid\)](#)

Assesses if an object is an instance of a type tid.

- [MM::INT32 getValue \(\)](#)
- [MM::INT32 getIntValue \(\)](#)
- [MM::Location * getLocation \(\)](#)
- [MM::BOOLEAN greaterEquals \(MM::UINT32 val\)](#)
- [MM::VOID toString \(MM::String *buf\)](#)

Additional Inherited Members

6.76.1 Constructor & Destructor Documentation

6.76.1.1 NumberValExp::NumberValExp (MM::INT32 val)

value source location

6.76.1.2 `NumberValExp::NumberValExp (MM::INT32 val, MM::Location * loc)`

6.76.1.3 `NumberValExp::NumberValExp (MM::INT32 val, MM::UINT8 fraction)`

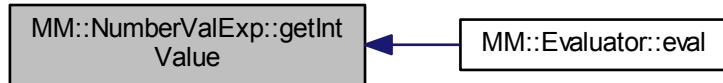
6.76.1.4 `NumberValExp::NumberValExp (MM::INT32 val, MM::UINT8 fraction, MM::Location * loc)`

6.76.1.5 `NumberValExp::~NumberValExp ()`

6.76.2 Member Function Documentation

6.76.2.1 `MM::INT32 NumberValExp::getIntValue ()`

Here is the caller graph for this function:



6.76.2.2 `MM::Location * NumberValExp::getLocation ()`

6.76.2.3 `MM::TID NumberValExp::getTypeId () [virtual]`

Retrieves the type id of a [Exp](#) object.

Retrieves the type id of a [TriggerExp](#) object.

Returns

`type id`

Reimplemented from [MM::ValExp](#).

6.76.2.4 `MM::INT32 NumberValExp::getValue ()`

Here is the caller graph for this function:



6.76.2.5 MM::BOOLEAN NumberValExp::greaterEquals (MM::UINT32 *val*) [virtual]

Implements [MM::ValExp](#).

6.76.2.6 MM::BOOLEAN NumberValExp::instanceof (MM::TID *tid*) [virtual]

Assesses if an object is an instance of a type tid.

Parameters

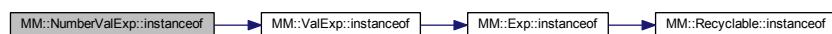
<i>tid</i>	type id
------------	---------

Returns

MM_TRUE if this object is instance of tid, MM_FALSE otherwise

Reimplemented from [MM::ValExp](#).

Here is the call graph for this function:

6.76.2.7 MM::VOID NumberValExp::recycle (MM::Recycler * *r*) [virtual]

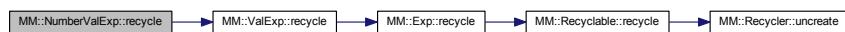
Recycles an [Exp](#) object in a [Recycler](#).

Parameters

<i>r</i>	Recycler object
----------	-----------------

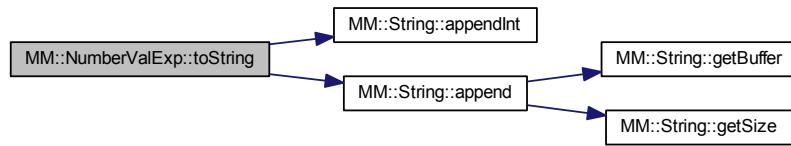
Implements [MM::Exp](#).

Here is the call graph for this function:

6.76.2.8 MM::VOID NumberValExp::toString (MM::String * *buf*) [virtual]

Implements [MM::ValExp](#).

Here is the call graph for this function:



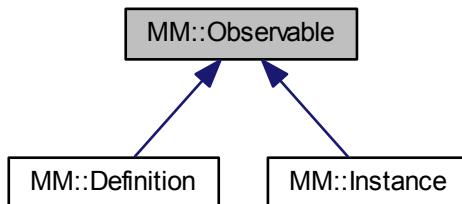
The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[NumberValExp.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[NumberValExp.cpp](#)

6.77 MM::Observable Class Reference

```
#include <Observable.h>
```

Inheritance diagram for MM::Observable:



Public Member Functions

- [Observable \(\)](#)
- [~Observable \(\)](#)
- [MM::VOID recyclce \(MM::Recycler *r\)](#)
- virtual [MM::TID getTypeld \(\)](#)
- virtual [MM::BOOLEAN instanceof \(MM::TID tid\)](#)
- [MM::VOID addObserver \(MM::Observer *observer\)](#)
- [MM::VOID removeObserver \(MM::Observer *observer\)](#)
- [MM::VOID notifyObservers \(MM::Observable *observable, MM::VOID *aux, MM::UINT32 message, MM::V-OID *object\)](#)

6.77.1 Constructor & Destructor Documentation

6.77.1.1 Observable::Observable()

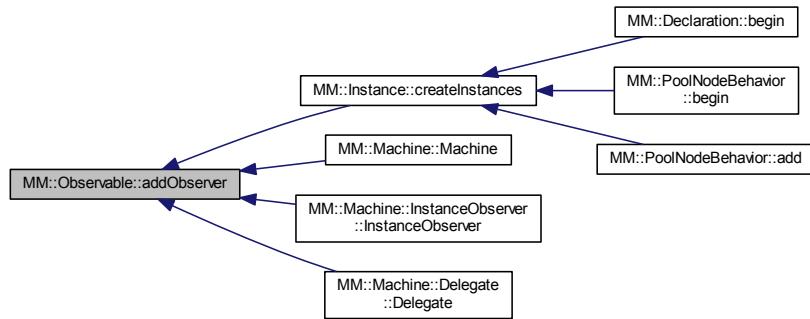
objects observing the observable

6.77.1.2 Observable::~Observable()

6.77.2 Member Function Documentation

6.77.2.1 MM::VOID Observable::addObserver(MM::Observer * observer)

Here is the caller graph for this function:



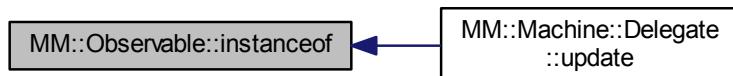
6.77.2.2 MM::TID Observable::getTypeId() [virtual]

Reimplemented in [MM::Instance](#), and [MM::Definition](#).

6.77.2.3 MM::BOOLEAN Observable::instanceof(MM::TID tid) [virtual]

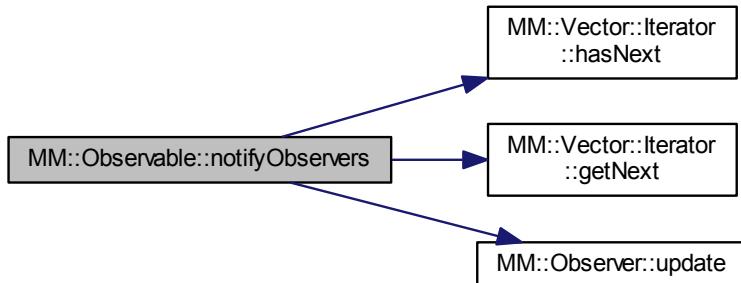
Reimplemented in [MM::Instance](#), and [MM::Definition](#).

Here is the caller graph for this function:

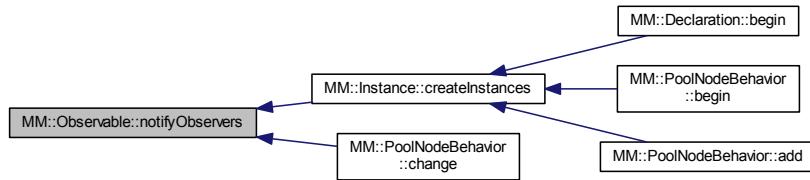


6.77.2.4 **MM::VOID Observable:::notifyObservers (MM::Observable * *observable*, MM::VOID * *aux*, MM::UINT32 *message*, MM::VOID * *object*)**

Here is the call graph for this function:



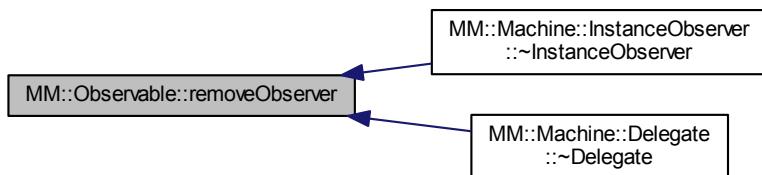
Here is the caller graph for this function:



6.77.2.5 **MM::VOID Observable:::recycle (MM::Recycler * *r*)**

6.77.2.6 **MM::VOID Observable:::removeObserver (MM::Observer * *observer*)**

Here is the caller graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Observable.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Observable.cpp](#)

6.78 Observable Class Reference

The [Observable](#) abstraction enables notifying Observers of changes to [Observable](#) objects when `notifyObservers` is called.

```
#include <Observable.h>
```

6.78.1 Detailed Description

The [Observable](#) abstraction enables notifying Observers of changes to [Observable](#) objects when `notifyObservers` is called.

The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Observable.h](#)

6.79 Observer Class Reference

The [Observer](#) abstraction enables observing changes to [Observable](#) objects via the `update` callback.

```
#include <Observer.h>
```

6.79.1 Detailed Description

The [Observer](#) abstraction enables observing changes to [Observable](#) objects via the `update` callback.

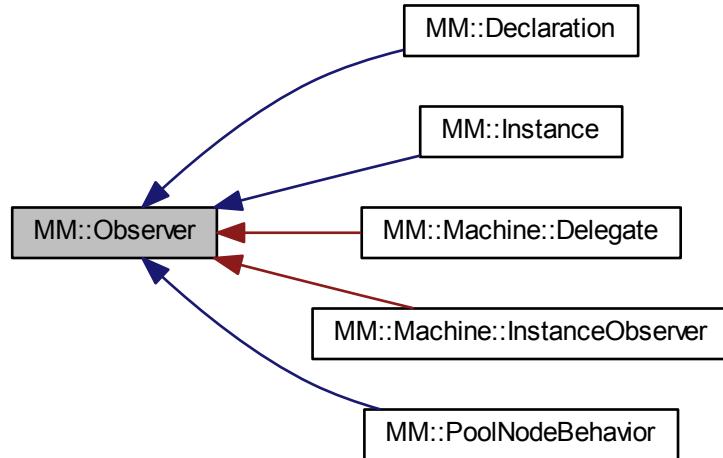
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Observer.h](#)

6.80 MM::Observer Class Reference

```
#include <Observer.h>
```

Inheritance diagram for MM::Observer:



Public Member Functions

- `Observer ()`
- `~Observer ()`
- virtual `MM::TID getTypeId ()`
- virtual `MM::BOOLEAN instanceof (MM::TID tid)`
- virtual `MM::VOID update (MM::Observable *observable, MM::VOID *aux, MM::UINT32 message, MM::VOID *object)=0`

6.80.1 Constructor & Destructor Documentation

6.80.1.1 `Observer::Observer ()`

6.80.1.2 `Observer::~Observer ()`

6.80.2 Member Function Documentation

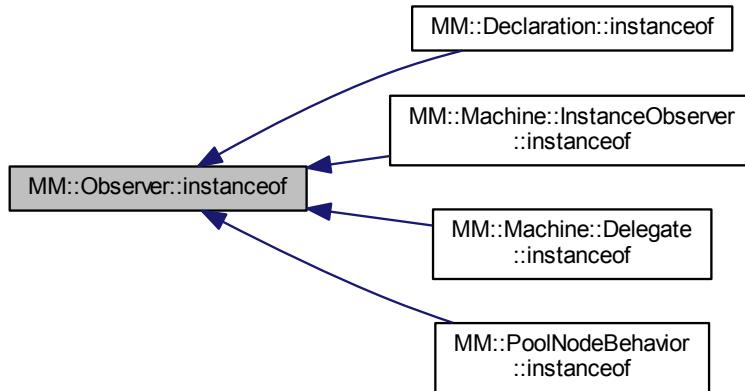
6.80.2.1 `MM::TID Observer::getTypeId () [virtual]`

Reimplemented in `MM::Machine::Delegate`, `MM::Instance`, `MM::PoolNodeBehavior`, `MM::Machine::InstanceObserver`, and `MM::Declaration`.

6.80.2.2 `MM::BOOLEAN Observer::instanceof (MM::TID tid) [virtual]`

Reimplemented in `MM::Machine::Delegate`, `MM::Instance`, `MM::PoolNodeBehavior`, `MM::Machine::InstanceObserver`, and `MM::Declaration`.

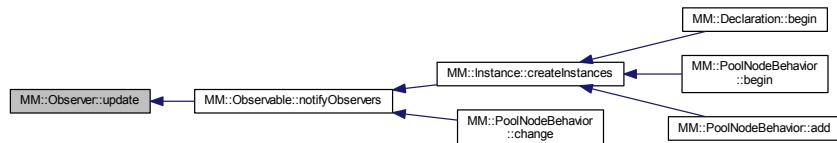
Here is the caller graph for this function:



6.80.2.3 virtual MM::VOID MM::Observer::update (MM::Observable * *observable*, MM::VOID * *aux*, MM::UINT32 *message*, MM::VOID * *object*) [pure virtual]

Implemented in [MM::Instance](#), [MM::Machine::Delegate](#), [MM::PoolNodeBehavior](#), [MM::Machine::InstanceObserver](#), and [MM::Declaration](#).

Here is the caller graph for this function:



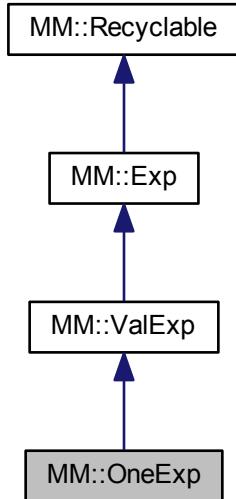
The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Observer.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Observer.cpp](#)

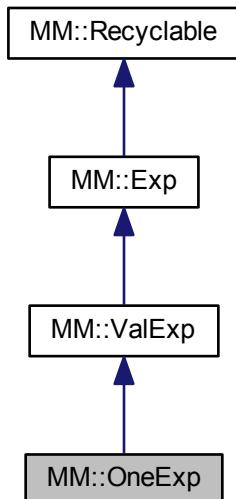
6.81 MM::OneExp Class Reference

```
#include <OneExp.h>
```

Inheritance diagram for MM::OneExp:



Collaboration diagram for MM::OneExp:



Public Member Functions

- [**OneExp \(\)**](#)
- [**OneExp \(MM::Location *loc\)**](#)

- [~OneExp \(\)](#)
- [MM::VOID recycle \(MM::Recycler *r\)](#)
- [MM::TID getTypeId \(\)](#)

Retrieves the type id of a [OneExp](#) object.
- [MM::BOOLEAN instanceof \(MM::TID tid\)](#)

Assesses if an object is an instance of a type tid.
- [MM::BOOLEAN greaterEquals \(MM::UINT32 val\)](#)
- [MM::VOID toString \(MM::String *buf\)](#)

Serializes an [OneExp](#) object into a [String](#) buffer.

Additional Inherited Members

6.81.1 Constructor & Destructor Documentation

6.81.1.1 OneExp::OneExp()

source location

Constructs a [OneExp](#) object.

Returns

new [OneExp](#) object

6.81.1.2 OneExp::OneExp(MM::Location * loc)

Constructs a [OneExp](#) object.

Parameters

<i>loc</i>	source location
------------	-----------------

Returns

new [OneExp](#) object

6.81.1.3 OneExp::~OneExp()

Destructs an [OneExp](#) object.

6.81.2 Member Function Documentation

6.81.2.1 MM::TID OneExp::getTypeId() [virtual]

Retrieves the type id of a [OneExp](#) object.

[TID MM::OneExp::getTypeId\(\)](#)

Returns

type id

Reimplemented from [MM::ValExp](#).

6.81.2.2 MM::BOOLEAN OneExp::greaterEquals (MM::UINT32 val) [virtual]

Implements [MM::ValExp](#).

6.81.2.3 MM::BOOLEAN OneExp::instanceof (MM::TID tid) [virtual]

Assesses if an object is an instance of a type tid.

Parameters

<i>tid</i>	type id
------------	---------

Returns

MM_TRUE if this object is instance of tid, MM_FALSE otherwise

Reimplemented from [MM::ValExp](#).

Here is the call graph for this function:



6.81.2.4 MM::VOID OneExp::recycle (MM::Recycler * r) [virtual]

Recycles an [OneExp](#) object in a [Recycler](#).

Parameters

<i>r</i>	Recycler
----------	--------------------------

Implements [MM::Exp](#).

Here is the call graph for this function:



6.81.2.5 MM::VOID OneExp::toString (MM::String * buf) [virtual]

Serializes an [OneExp](#) object into a [String](#) buffer.

Parameters

<i>buf</i>	String buffer to serialize this object into
------------	---

Implements [MM::ValExp](#).

The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[OneExp.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[OneExp.cpp](#)

6.82 OneExp Class Reference

The [OneExp](#) abstraction expresses shorthand expressions on edges. \rightarrow : [OneExp](#) has a flow of one: NumberValueExp(1) . * . \rightarrow : [BinExp](#) has lhs [OneExp](#) and rhs [OneExp](#), then `isTrigger() = MM_TRUE` . == . \rightarrow : [BinExp](#) has lhs [OneExp](#) and rhs [OneExp](#): lhs [OneExp](#) refers to source and rhs [OneExp](#) refers to target.

```
#include <OneExp.h>
```

6.82.1 Detailed Description

The [OneExp](#) abstraction expresses shorthand expressions on edges. \rightarrow : [OneExp](#) has a flow of one: NumberValueExp(1) . * . \rightarrow : [BinExp](#) has lhs [OneExp](#) and rhs [OneExp](#), then `isTrigger() = MM_TRUE` . == . \rightarrow : [BinExp](#) has lhs [OneExp](#) and rhs [OneExp](#): lhs [OneExp](#) refers to source and rhs [OneExp](#) refers to target.

The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[OneExp.h](#)

6.83 MM::Operator Class Reference

```
#include <Operator.h>
```

Public Types

- enum [__OP](#) {
 [OP_ERROR](#), [OP_ADD](#), [OP_SUB](#), [OP_MUL](#),
 [OP_DIV](#), [OP_AND](#), [OP_OR](#), [OP_GT](#),
 [OP_LT](#), [OP_LE](#), [OP_GE](#), [OP_EQ](#),
 [OP_NEQ](#), [OP_UNM](#), [OP_NOT](#), [OP_PERCENT](#),
 [OP_PER](#) }
- typedef enum [MM::Operator::__OP](#) [OP](#)

Static Public Attributes

- static const [MM::CHAR *](#) [OP_STR](#) []
- static const [MM::UINT32](#) [OP_LEN](#) []

6.83.1 Member Typedef Documentation

6.83.1.1 [typedef enum MM::Operator::__OP](#) [MM::Operator::OP](#)

6.83.2 Member Enumeration Documentation

6.83.2.1 [enum MM::Operator::__OP](#)

Enumerator

[***OP_ERROR***](#)

[***OP_ADD***](#) error

[***OP_SUB***](#) •

[***OP_MUL***](#) •

[***OP_DIV***](#) •

```

OP_AND      /
OP_OR       &&
OP_GT       ||
OP_LT       >>
OP_LE       <
OP_GE       <=
OP_EQ       >>=

OP_NEQ

OP_UNM     !=
OP_NOT      ~
OP_PERCENT !
OP_PER      %
|
```

6.83.3 Member Data Documentation

6.83.3.1 const MM::UINT32 Operator::OP_LEN [static]

Initial value:

```
=
{
    strlen(OP_STR[0]),
    strlen(OP_STR[1]),
    strlen(OP_STR[2]),
    strlen(OP_STR[3]),
    strlen(OP_STR[4]),
    strlen(OP_STR[5]),
    strlen(OP_STR[6]),
    strlen(OP_STR[7]),
    strlen(OP_STR[8]),
    strlen(OP_STR[9]),
    strlen(OP_STR[10]),
    strlen(OP_STR[11]),
    strlen(OP_STR[12]),
    strlen(OP_STR[13]),
    strlen(OP_STR[14]),
    strlen(OP_STR[15]),
    strlen(OP_STR[16])
}
```

[Operator strings](#)

6.83.3.2 const MM::CHAR * Operator::OP_STR [static]

Initial value:

```
=
{
    "error",
    "+",
    "-",
    "*",
    "/",
    "&&",
    "||",
    ">",
    "<",
    "<=",
    ">=",
    "=="
```

```

"!=",
"~",
"!~",
"%",
"|"}
```

The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Operator.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Operator.cpp](#)

6.84 Operator Class Reference

The [Operator](#) abstraction defines operator lexicals and codes.

```
#include <Operator.h>
```

6.84.1 Detailed Description

The [Operator](#) abstraction defines operator lexicals and codes.

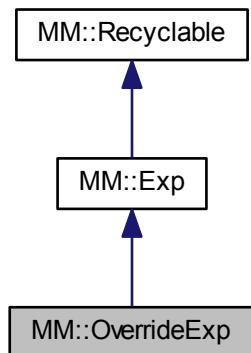
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Operator.h](#)

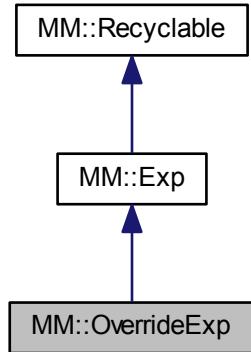
6.85 MM::OverrideExp Class Reference

```
#include <OverrideExp.h>
```

Inheritance diagram for MM::OverrideExp:



Collaboration diagram for MM::OverrideExp:



Public Member Functions

- [OverrideExp \(MM::Exp *exp\)](#)
- [OverrideExp \(MM::Location *lparenLoc, MM::Exp *exp, MM::Location *rparenLoc\)](#)
- [~OverrideExp \(\)](#)
- [MM::VOID recycle \(MM::Recycler *r\)](#)
- [MM::TID getTypeId \(\)](#)
Retrieves the type id of a `OverrideExp` object.
- [MM::BOOLEAN instanceof \(MM::TID tid\)](#)
Assesses if an object is an instance of a type tid.
- [MM::Exp * getExp \(\)](#)
Retrieves the expression of an `OverrideExp` object.
- [MM::VOID toString \(MM::String *buf\)](#)
Serializes an `OverrideExp` object into a `String` buffer.

Additional Inherited Members

6.85.1 Constructor & Destructor Documentation

6.85.1.1 MM::OverrideExp::OverrideExp (MM::Exp * exp)

right parenthesis location

Constructs an `OverrideExp` object.

Parameters

<code>exp</code>	expression
------------------	------------

Returns

`new OverrideExp` object

Constructs an `OverrideExp` object.

Parameters

<i>exp</i>	expression
<i>lparenLoc</i>	left parenthesis source location
<i>rparenLoc</i>	right parenthesis source location

Returns

new [OverrideExp](#) object

6.85.1.2 `MM::OverrideExp::OverrideExp (MM::Location * lparenLoc, MM::Exp * exp, MM::Location * rparenLoc)`

6.85.1.3 `MM::OverrideExp::~OverrideExp ()`

Deconstructs a [OverrideExp](#) object.

6.85.2 Member Function Documentation

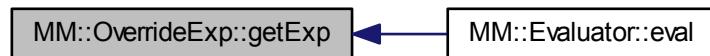
6.85.2.1 `MM::Exp * MM::OverrideExp::getExp ()`

Retrieves the expression of an [OverrideExp](#) object.

Returns

overriden expression

Here is the caller graph for this function:



6.85.2.2 `MM::TID MM::OverrideExp::getTypeId () [virtual]`

Retrieves the type id of a [OverrideExp](#) object.

Returns

type id

Reimplemented from [MM::Exp](#).

6.85.2.3 `MM::BOOLEAN MM::OverrideExp::instanceof (MM::TID tid) [virtual]`

Assesses if an object is an instance of a type tid.

Parameters

<i>tid</i>	type id
------------	---------

Returns

MM_TRUE if this object is instance of tid, MM_FALSE otherwise

Reimplemented from [MM::Exp](#).

Here is the call graph for this function:

**6.85.2.4 MM::VOID MM::OverrideExp::recycle (MM::Recycler * r) [virtual]**

Recycles an [OverrideExp](#) object in a [Recycler](#).

Parameters

<i>r</i>	Recycler
----------	----------

Implements [MM::Exp](#).

Here is the call graph for this function:

**6.85.2.5 MM::VOID MM::OverrideExp::toString (MM::String * buf) [virtual]**

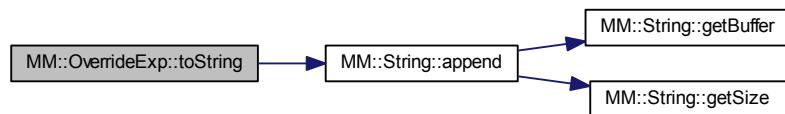
Serializes an [OverrideExp](#) object into a [String](#) buffer.

Parameters

<i>buf</i>	String buffer to serialize this object into
------------	---

Implements [MM::Exp](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[OverrideExp.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[OverrideExp.cpp](#)

6.86 OvertideExp Class Reference

The OverrideExp abstraction expresses an overriden expression.

```
#include <OverrideExp.h>
```

6.86.1 Detailed Description

The OverrideExp abstraction expresses an overriden expression.

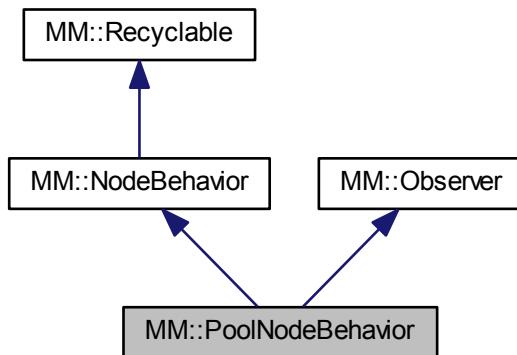
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[OverrideExp.h](#)

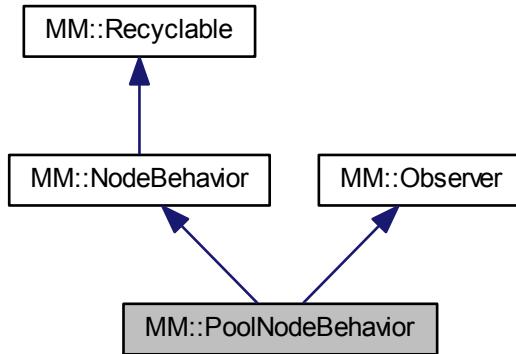
6.87 MM::PoolNodeBehavior Class Reference

```
#include <PoolNodeBehavior.h>
```

Inheritance diagram for MM::PoolNodeBehavior:



Collaboration diagram for MM::PoolNodeBehavior:



Public Member Functions

- `PoolNodeBehavior (MM::NodeBehavior::IO io, MM::NodeBehavior::When when, MM::NodeBehavior::Act act, MM::NodeBehavior::How how, MM::Name *of, MM::UINT32 at, MM::UINT32 max, MM::Exp *exp, MM::Map< MM::Name *, Node *, MM::Name::Compare > *interfaces)`
- `~PoolNodeBehavior ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::VOID update (MM::Observable *observable, MM::VOID *aux, MM::UINT32 message, MM::VOID *object)`
- `MM::Name * getTypeName ()`
- `MM::VOID setDefinition (MM::Definition *def)`
- `MM::Definition * getDefinition ()`
- `MM::Node * getInterface (MM::Name *name)`
- `MM::VOID addInterface (MM::Machine *m, MM::Node *node)`
- `MM::VOID removeInterface (MM::Machine *m, MM::Node *node)`
- `MM::Name * getOf ()`
- `MM::UINT32 getAt ()`
- `MM::UINT32 getMax ()`
- `MM::Exp * getAdd ()`
- `MM::VOID setAt (MM::UINT32 at)`
- `MM::VOID setMax (MM::UINT32 max)`
- `MM::VOID setAdd (MM::Exp *exp)`
- `MM::UINT32 getCreateMessage ()`
- `MM::UINT32 getUpdateMessage ()`
- `MM::UINT32 getDeleteMessage ()`
- `MM::VOID stepPullAll (MM::Node *tgtNode, MM::Instance *tgtInstance, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)`
- `MM::VOID stepPushAll (MM::Node *srcNode, MM::Instance *srcInstance, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)`
- `MM::VOID begin (MM::Instance *i, MM::Machine *m, MM::Node *n)`
- `MM::VOID end (MM::Instance *i, MM::Machine *m, MM::Node *n)`
- `MM::VOID change (MM::Instance *i, MM::Machine *m, MM::Node *n)`

- `MM::INT32 getAmount (MM::Instance *i, MM::Machine *m, MM::Node *n)`
- `MM::VOID add (MM::Instance *i, MM::Machine *m, MM::Node *n, MM::UINT32 amount)`
- `MM::VOID sub (MM::Instance *i, MM::Machine *m, MM::Node *n, MM::UINT32 amount)`
- `MM::UINT32 getCapacity (MM::Instance *i, MM::Node *n)`
- `MM::UINT32 getResources (MM::Instance *i, MM::Node *n)`
- `MM::BOOLEAN hasCapacity (MM::Instance *i, MM::Node *n, MM::UINT32 amount)`
- `MM::BOOLEAN hasResources (MM::Instance *i, MM::Node *n, MM::UINT32 amount)`
- `MM::VOID toString (MM::String *buf)`
- `MM::VOID toString (MM::String *buf, MM::Name *name)`

Additional Inherited Members

6.87.1 Constructor & Destructor Documentation

6.87.1.1 `PoolNodeBehavior::PoolNodeBehavior (MM::NodeBehavior::IO io, MM::NodeBehavior::When when, MM::NodeBehavior::Act act, MM::NodeBehavior::How how, MM::Name * of, MM::UINT32 at, MM::UINT32 max, MM::Exp * exp, MM::Map<MM::Name *, Node *, MM::Name::Compare > * interfaces)`

interfaces resulting from definition

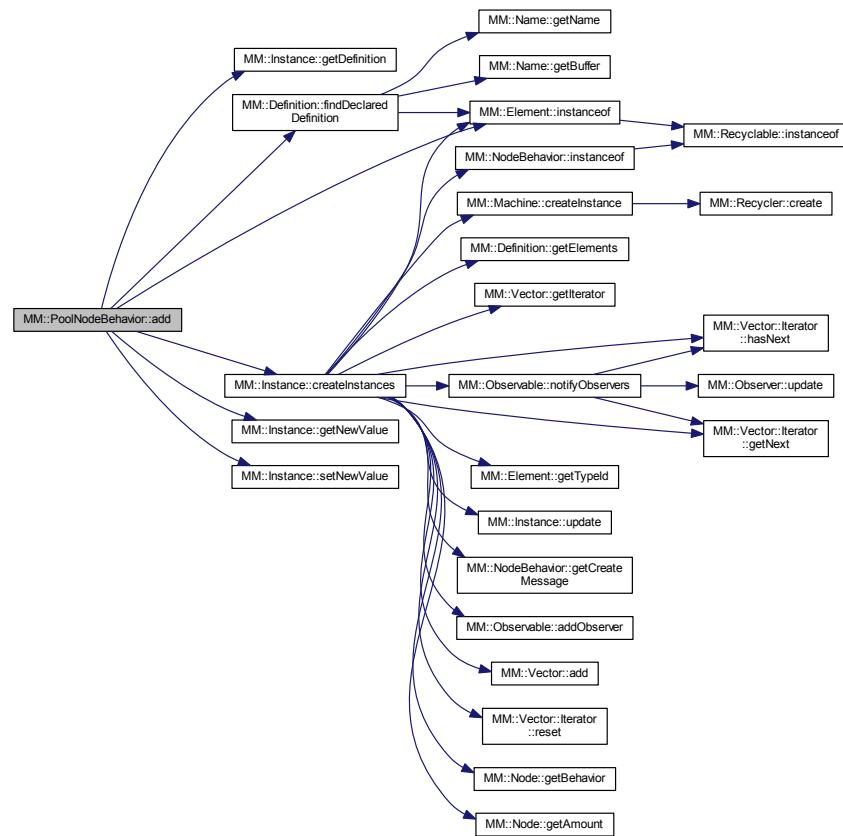
6.87.1.2 PoolNodeBehavior::~PoolNodeBehavior ()

6.87.2 Member Function Documentation

6.87.2.1 `MM::VOID PoolNodeBehavior::add (MM::Instance * i, MM::Machine * m, MM::Node * n, MM::UINT32 amount) [virtual]`

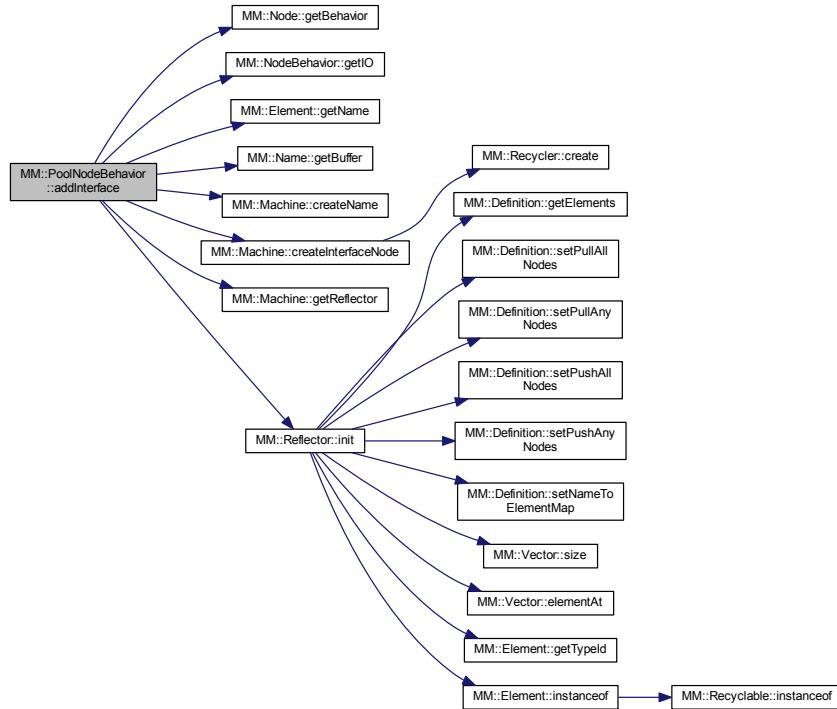
Implements [MM::NodeBehavior](#).

Here is the call graph for this function:



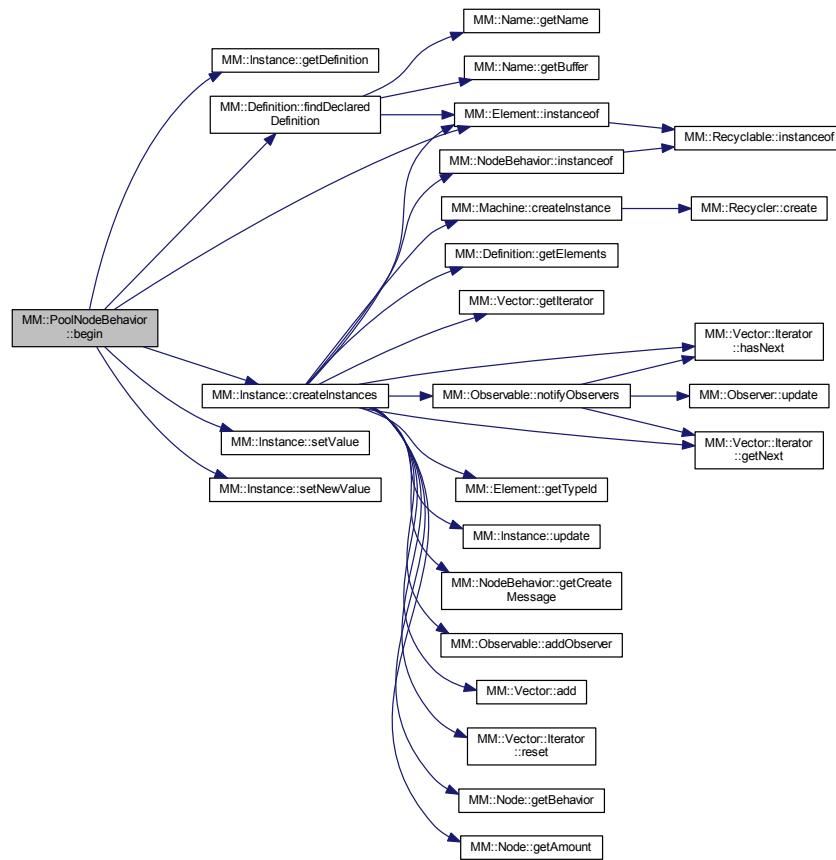
6.87.2.2 MM::VOID PoolNodeBehavior::addInterface (MM::Machine * *m*, MM::Node * *node*)

Here is the call graph for this function:

6.87.2.3 MM::VOID PoolNodeBehavior::begin (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*)
[virtual]

Implements [MM::NodeBehavior](#).

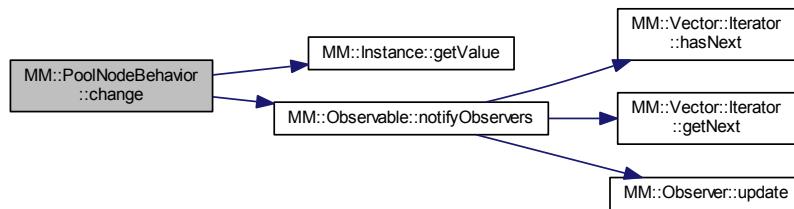
Here is the call graph for this function:



6.87.2.4 `MM::VOID PoolNodeBehavior::change (MM::Instance * i, MM::Machine * m, MM::Node * n) [virtual]`

Implements [MM::NodeBehavior](#).

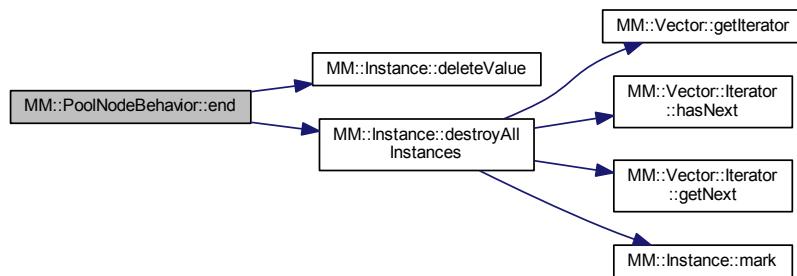
Here is the call graph for this function:



6.87.2.5 MM::VOID PoolNodeBehavior::end (MM::Instance * i, MM::Machine * m, MM::Node * n) [virtual]

Implements [MM::NodeBehavior](#).

Here is the call graph for this function:

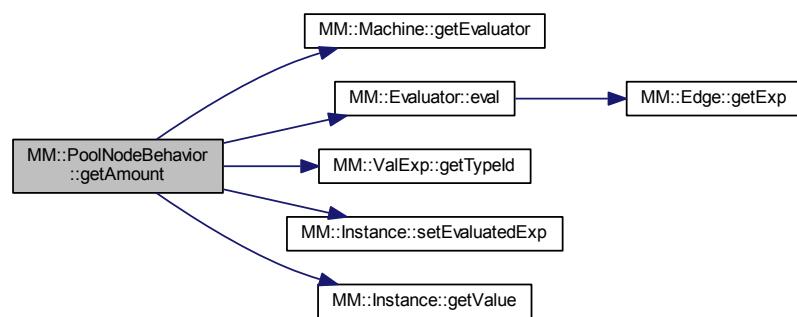


6.87.2.6 MM::Exp * PoolNodeBehavior::getAdd ()

6.87.2.7 MM::INT32 PoolNodeBehavior::getAmount (MM::Instance * i, MM::Machine * m, MM::Node * n) [virtual]

Reimplemented from [MM::NodeBehavior](#).

Here is the call graph for this function:

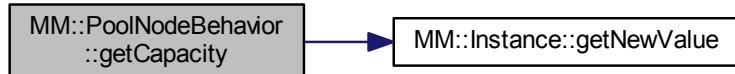


6.87.2.8 MM::UINT32 PoolNodeBehavior::getAt ()

6.87.2.9 MM::UINT32 PoolNodeBehavior::getCapacity (MM::Instance * i, MM::Node * n) [virtual]

Implements [MM::NodeBehavior](#).

Here is the call graph for this function:



6.87.2.10 MM::UINT32 PoolNodeBehavior::getCreateMessage() [virtual]

Implements [MM::NodeBehavior](#).

6.87.2.11 MM::Definition * PoolNodeBehavior::getDefinition()

6.87.2.12 MM::UINT32 PoolNodeBehavior::getDeleteMessage() [virtual]

Implements [MM::NodeBehavior](#).

6.87.2.13 MM::Node * PoolNodeBehavior::getInterface(MM::Name * name)

Here is the caller graph for this function:



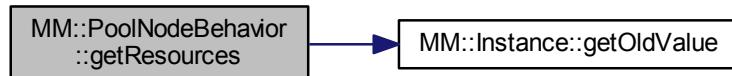
6.87.2.14 MM::UINT32 PoolNodeBehavior::getMax()

6.87.2.15 MM::Name * PoolNodeBehavior::getOf()

6.87.2.16 MM::UINT32 PoolNodeBehavior::getResources(MM::Instance * i, MM::Node * n) [virtual]

Implements [MM::NodeBehavior](#).

Here is the call graph for this function:



6.87.2.17 MM::TID PoolNodeBehavior::getTypeId() [virtual]

Reimplemented from [MM::NodeBehavior](#).

6.87.2.18 MM::Name * PoolNodeBehavior::getTypeName()

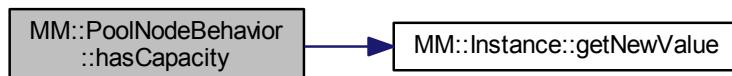
6.87.2.19 MM::UINT32 PoolNodeBehavior::getUpdateMessage() [virtual]

Implements [MM::NodeBehavior](#).

6.87.2.20 MM::BOOLEAN PoolNodeBehavior::hasCapacity(MM::Instance * i, MM::Node * n, MM::UINT32 amount) [virtual]

Implements [MM::NodeBehavior](#).

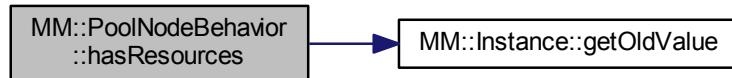
Here is the call graph for this function:



6.87.2.21 MM::BOOLEAN PoolNodeBehavior::hasResources(MM::Instance * i, MM::Node * n, MM::UINT32 amount) [virtual]

Implements [MM::NodeBehavior](#).

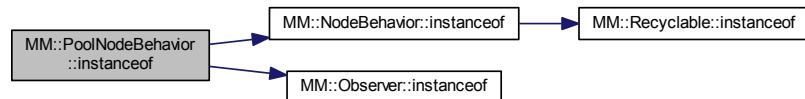
Here is the call graph for this function:



6.87.2.22 MM::BOOLEAN PoolNodeBehavior::instanceof(MM::TID tid) [virtual]

Reimplemented from [MM::NodeBehavior](#).

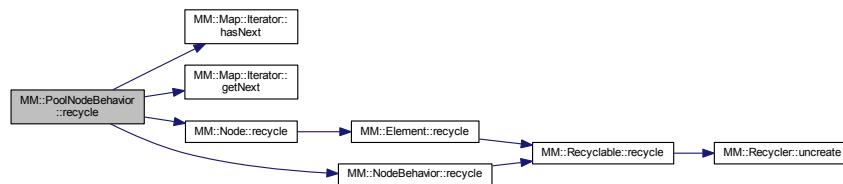
Here is the call graph for this function:



6.87.2.23 MM::VOID PoolNodeBehavior::recycle(MM::Recycler * r) [virtual]

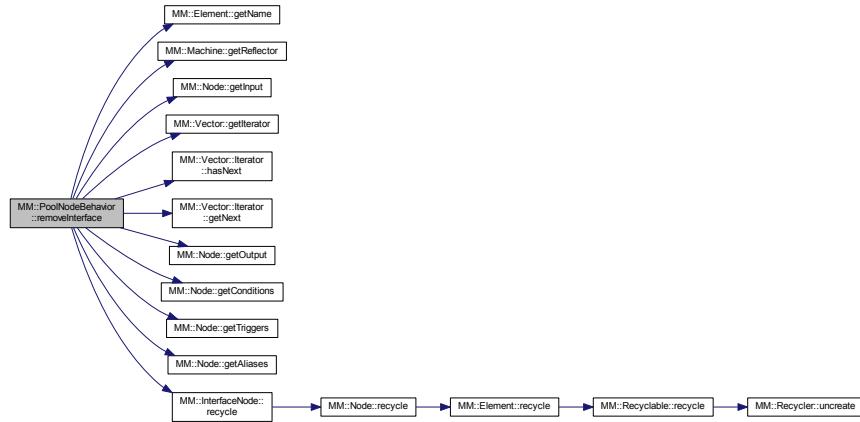
Reimplemented from [MM::NodeBehavior](#).

Here is the call graph for this function:



6.87.2.24 MM::VOID PoolNodeBehavior::removeInterface (MM::Machine * *m*, MM::Node * *node*)

Here is the call graph for this function:



6.87.2.25 MM::VOID PoolNodeBehavior::setAdd (MM::Exp * *exp*)

6.87.2.26 MM::VOID PoolNodeBehavior::setAt (MM::UINT32 *at*)

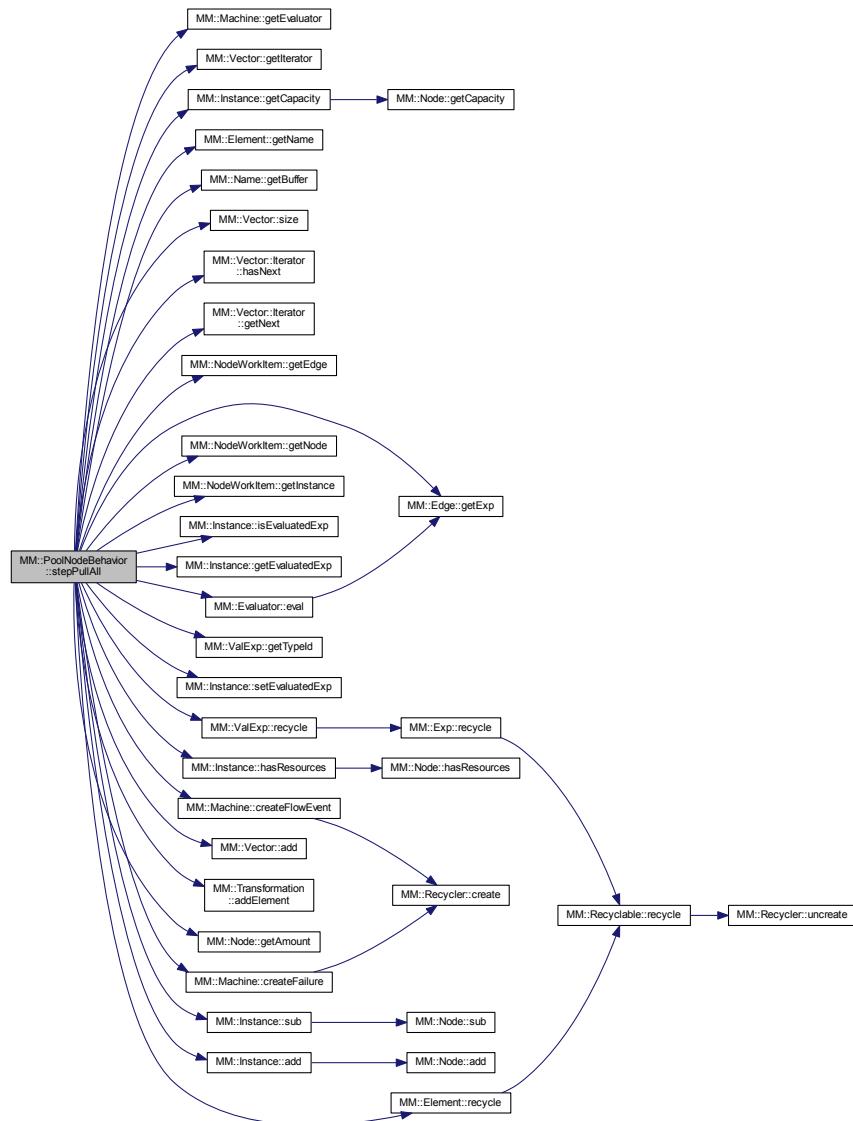
6.87.2.27 MM::VOID PoolNodeBehavior::setDefinition (MM::Definition * *def*)

6.87.2.28 MM::VOID PoolNodeBehavior::setMax (MM::UINT32 *max*)

6.87.2.29 MM::VOID PoolNodeBehavior::stepPullAll (MM::Node * *tgtNode*, MM::Instance * *tgtInstance*, MM::Vector<MM::NodeWorkItem *> * *work*, MM::Machine * *m*, MM::Transition * *tr*) [virtual]

Implements [MM::NodeBehavior](#).

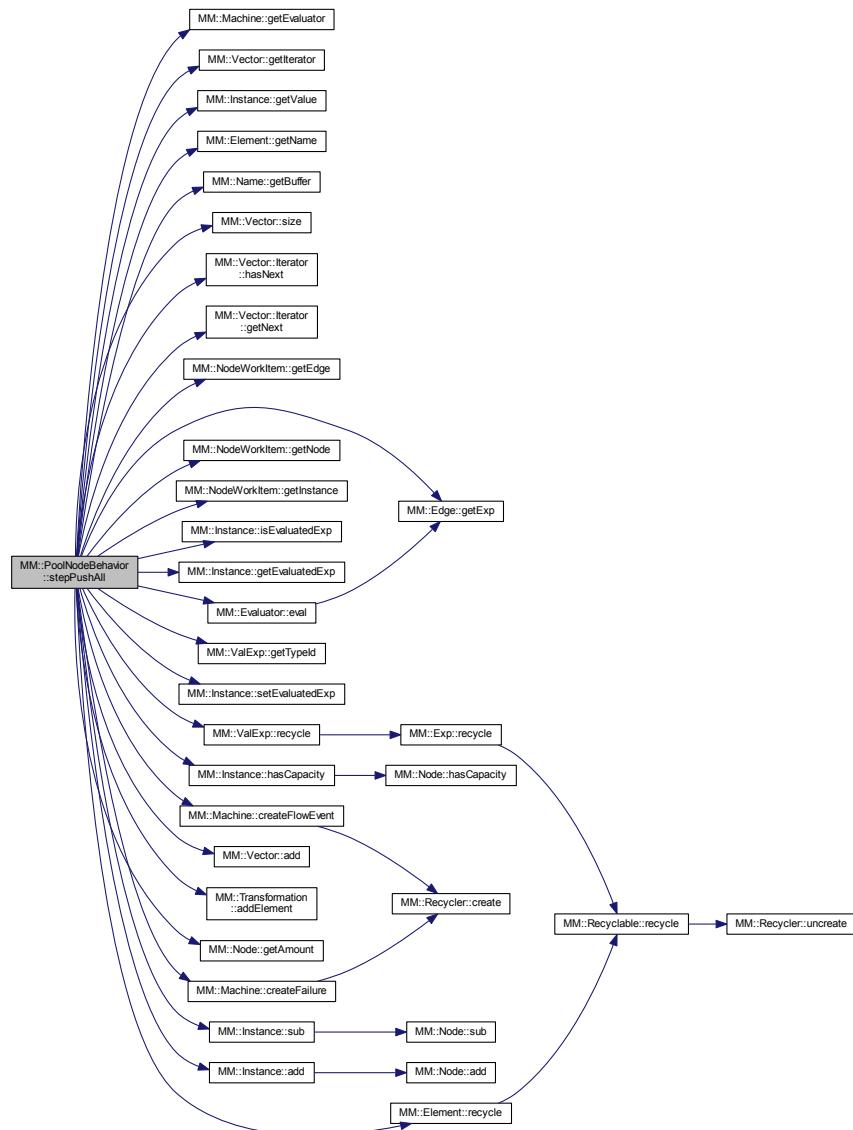
Here is the call graph for this function:



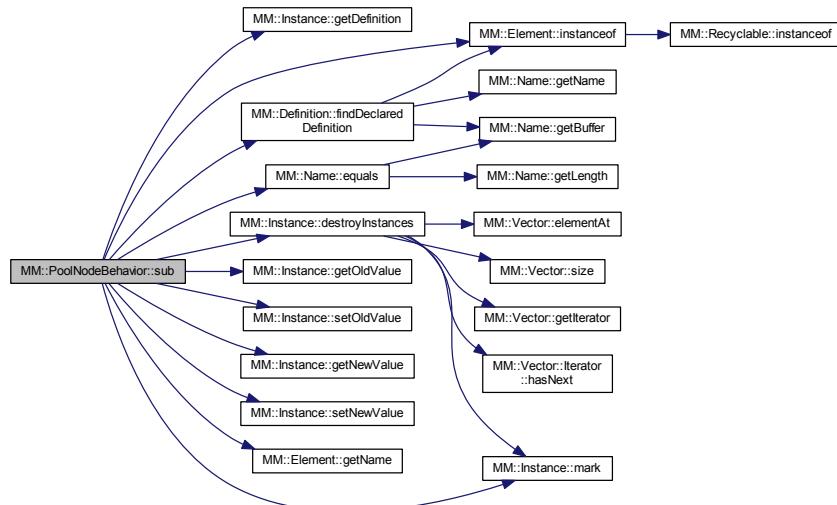
6.87.2.30 MM::VOID PoolNodeBehavior::stepPushAll (`MM::Node * srcNode, MM::Instance * srcInstance, MM::Vector< MM::NodeWorkItem * > * work, MM::Machine * m, MM::Transition * tr`) [virtual]

Implements [MM::NodeBehavior](#).

Here is the call graph for this function:



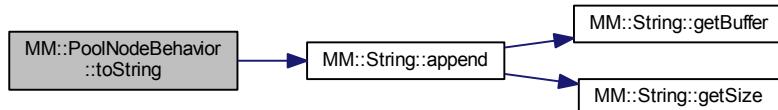
Here is the call graph for this function:



6.87.2.32 MM::VOID PoolNodeBehavior::toString (MM::String * buf) [virtual]

Implements [MM::Recyclable](#).

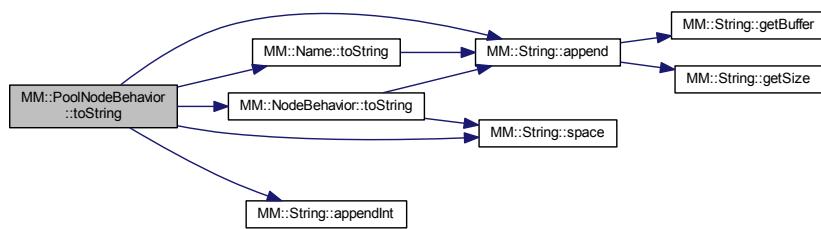
Here is the call graph for this function:



6.87.2.33 MM::VOID PoolNodeBehavior::toString (MM::String * buf, MM::Name * name) [virtual]

Reimplemented from [MM::NodeBehavior](#).

Here is the call graph for this function:



6.87.2.34 MM::VOID PoolNodeBehavior::update (MM::Observable * *observable*, MM::VOID * *aux*, MM::UINT32 *message*, MM::VOID * *object*) [virtual]

Implements [MM::Observer](#).

The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[PoolNodeBehavior.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[PoolNodeBehavior.cpp](#)

6.88 PoolNodeBehavior Class Reference

The [PoolNodeBehavior](#) abstraction defines the behavior of pool nodes.

```
#include <PoolNodeBehavior.h>
```

6.88.1 Detailed Description

The [PoolNodeBehavior](#) abstraction defines the behavior of pool nodes.

Note

Strategy

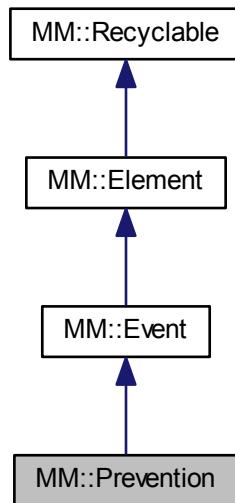
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[PoolNodeBehavior.h](#)

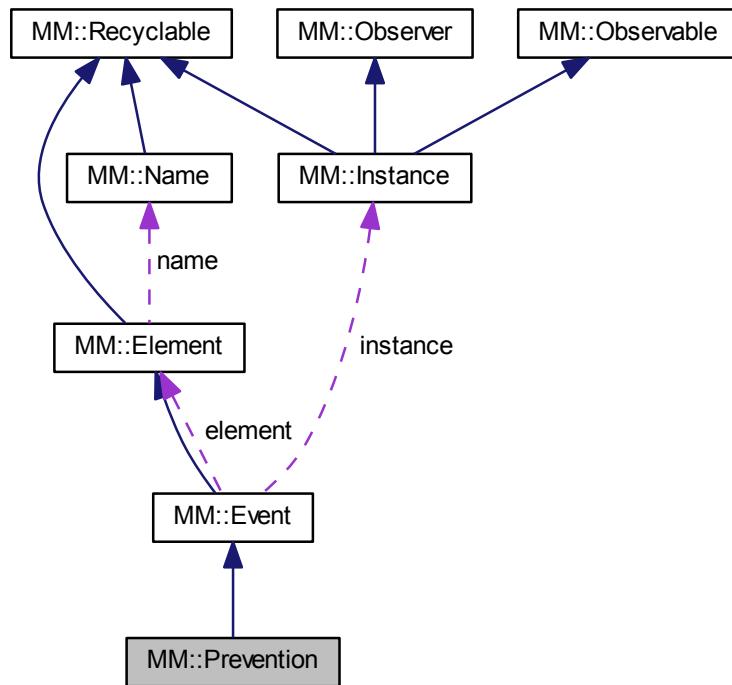
6.89 MM::Prevention Class Reference

```
#include <Prevention.h>
```

Inheritance diagram for MM::Prevention:



Collaboration diagram for MM::Prevention:



Public Member Functions

- Prevention (MM::Name *name)
- Prevention (MM::Location *loc, MM::Name *name)
- Prevention (MM::Instance *instance, MM::Edge *edge)
- ~Prevention ()
- MM::VOID recycle (MM::Recycler *r)
- MM::TID getTypeld ()
- MM::BOOLEAN instanceof (MM::TID tid)
- MM::Location * getLocation ()
- MM::MESSAGE getMessage ()
- MM::VOID toString (MM::String *buf)
- MM::VOID toString (MM::String *buf, MM::UINT32 indent)

Additional Inherited Members

6.89.1 Constructor & Destructor Documentation

6.89.1.1 MM::Prevention::Prevention (MM::Name * name)

prevent keyword location (parsed)

6.89.1.2 **MM::Prevention::Prevention (MM::Location * loc, MM::Name * name)**

6.89.1.3 **MM::Prevention::Prevention (MM::Instance * instance, MM::Edge * edge)**

6.89.1.4 **MM::Prevention::~Prevention ()**

6.89.2 Member Function Documentation

6.89.2.1 **MM::Location * MM::Prevention::getLocation ()**

6.89.2.2 **MM::MESSAGE MM::Prevention::getMessage () [virtual]**

Implements [MM::Event](#).

6.89.2.3 **MM::TID MM::Prevention::getTypeId () [virtual]**

Reimplemented from [MM::Event](#).

6.89.2.4 **MM::BOOLEAN MM::Prevention::instanceof (MM::TID tid) [virtual]**

Reimplemented from [MM::Event](#).

Here is the call graph for this function:



6.89.2.5 **MM::VOID MM::Prevention::recycle (MM::Recycler * r) [virtual]**

Reimplemented from [MM::Event](#).

Here is the call graph for this function:



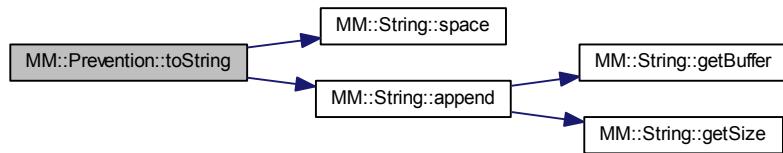
6.89.2.6 **MM::VOID MM::Prevention::toString (MM::String * buf) [virtual]**

Implements [MM::Event](#).

6.89.2.7 **MM::VOID MM::Prevention::toString (MM::String * buf, MM::UINT32 indent) [virtual]**

Implements [MM::Event](#).

Here is the call graph for this function:



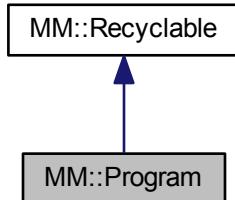
The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Prevention.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Prevention.cpp](#)

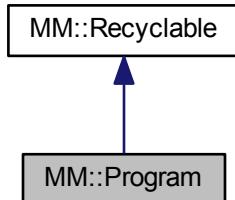
6.90 MM::Program Class Reference

```
#include <Program.h>
```

Inheritance diagram for MM::Program:



Collaboration diagram for MM::Program:



Public Member Functions

- [Program \(MM::Vector< Transformation * > *transformations\)](#)
- [~Program \(\)](#)
- [MM::VOID recycle \(MM::Recycler *r\)](#)
- [MM::TID getTypeld \(\)](#)
- [MM::BOOLEAN instanceof \(MM::TID tid\)](#)
- [MM::VOID addTransformation \(MM::Transformation *tr\)](#)
- [MM::Vector< MM::Transformation * > * getTransformations \(\)](#)
- [MM::VOID toString \(MM::String *buf\)](#)

6.90.1 Constructor & Destructor Documentation

6.90.1.1 Program::Program (MM::Vector< Transformation * > * *transformations*)

6.90.1.2 Program::~Program ()

6.90.2 Member Function Documentation

6.90.2.1 MM::VOID Program::addTransformation (MM::Transformation * *tr*)

6.90.2.2 MM::Vector< MM::Transformation * > * Program::getTransformations ()

Here is the caller graph for this function:



6.90.2.3 MM::TID Program::getTypeld () [virtual]

Reimplemented from [MM::Recyclable](#).

6.90.2.4 MM::BOOLEAN Program::instanceof (MM::TID *tid*) [virtual]

Reimplemented from [MM::Recyclable](#).

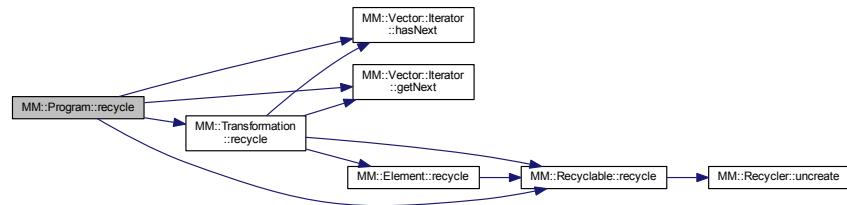
Here is the call graph for this function:



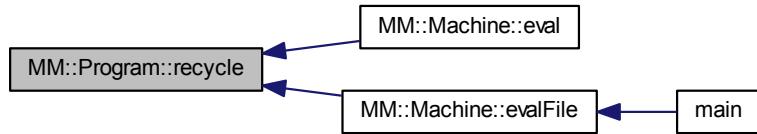
6.90.2.5 MM::VOID Program::recycle (MM::Recycler * r) [virtual]

Reimplemented from [MM::Recyclable](#).

Here is the call graph for this function:



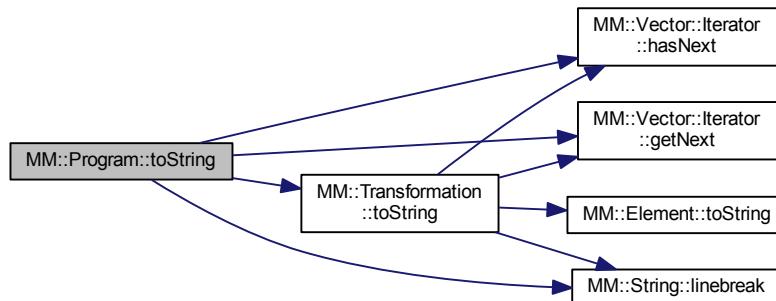
Here is the caller graph for this function:



6.90.2.6 MM::VOID Program::toString (MM::String * buf) [virtual]

Implements [MM::Recyclable](#).

Here is the call graph for this function:



Here is the caller graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Program.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Program.cpp](#)

6.91 Program Class Reference

A [Program](#) is a collection of elements that define state and behavior.

```
#include <Program.h>
```

6.91.1 Detailed Description

A [Program](#) is a collection of elements that define state and behavior.

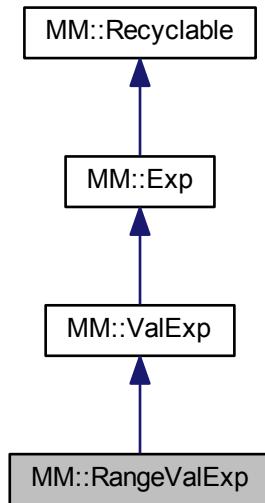
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Program.h](#)

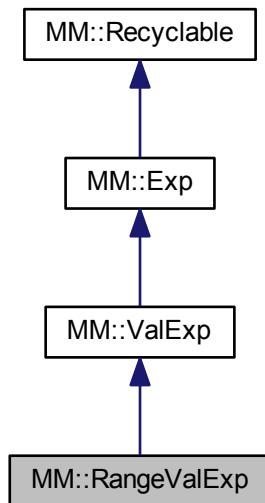
6.92 MM::RangeValExp Class Reference

```
#include <RangeValExp.h>
```

Inheritance diagram for MM::RangeValExp:



Collaboration diagram for MM::RangeValExp:



Public Member Functions

- [RangeValExp \(MM::UINT32 min, MM::UINT32 max\)](#)
- [RangeValExp \(MM::UINT32 min, MM::UINT32 max, MM::Location *minLoc, MM::Location *maxLoc, MM::Location *rangeLoc\)](#)

- `~RangeValExp ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeId ()`
Retrieves the type id of a `Exp` object.
- `MM::BOOLEAN instanceof (MM::TID tid)`
Assesses if an object is an instance of a type tid.
- `MM::UINT32 getMin ()`
- `MM::UINT32 getMax ()`
- `MM::INT32 getIntValue ()`
- `MM::BOOLEAN greaterEquals (MM::UINT32 val)`
- `MM::VOID toString (MM::String *buf)`

Additional Inherited Members

6.92.1 Constructor & Destructor Documentation

6.92.1.1 `RangeValExp::RangeValExp (MM::UINT32 min, MM::UINT32 max)`

range (dotdot) source location

6.92.1.2 `RangeValExp::RangeValExp (MM::UINT32 min, MM::UINT32 max, MM::Location * minLoc, MM::Location * maxLoc, MM::Location * rangeLoc)`

6.92.1.3 `RangeValExp::~RangeValExp ()`

6.92.2 Member Function Documentation

6.92.2.1 `MM::INT32 RangeValExp::getIntValue ()`

6.92.2.2 `MM::UINT32 RangeValExp::getMax ()`

Here is the caller graph for this function:



6.92.2.3 MM::UINT32 RangeValExp::getMin ()

Here is the caller graph for this function:



6.92.2.4 MM::TID RangeValExp::getTypeId () [virtual]

Retrieves the type id of a [Exp](#) object.

Retrieves the type id of a [TriggerExp](#) object.

Returns

type id

Reimplemented from [MM::ValExp](#).

6.92.2.5 MM::BOOLEAN RangeValExp::greaterEquals (MM::UINT32 val) [virtual]

Implements [MM::ValExp](#).

6.92.2.6 MM::BOOLEAN RangeValExp::instanceof (MM::TID tid) [virtual]

Assesses if an object is an instance of a type tid.

Parameters

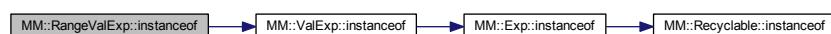
<i>tid</i>	type id
------------	---------

Returns

MM_TRUE if this object is instance of tid, MM_FALSE otherwise

Reimplemented from [MM::ValExp](#).

Here is the call graph for this function:



6.92.2.7 MM::VOID RangeValExp::recycle (MM::Recycler * r) [virtual]

Recycles an [Exp](#) object in a [Recycler](#).

Parameters

<i>r</i>	Recycler object
----------	-----------------

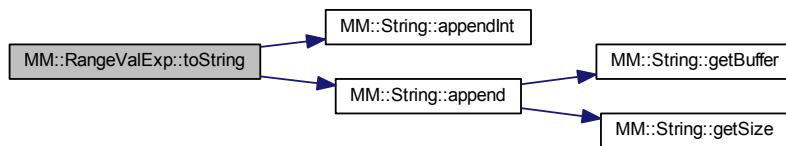
Implements [MM::Exp](#).

Here is the call graph for this function:

**6.92.2.8 MM::VOID RangeValExp::toString (MM::String * buf) [virtual]**

Implements [MM::ValExp](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[RangeValExp.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[RangeValExp.cpp](#)

6.93 RangeValExp Class Reference

The [RangeValExp](#) abstraction defines range value expressions.

```
#include <RangeValExp.h>
```

6.93.1 Detailed Description

The [RangeValExp](#) abstraction defines range value expressions.

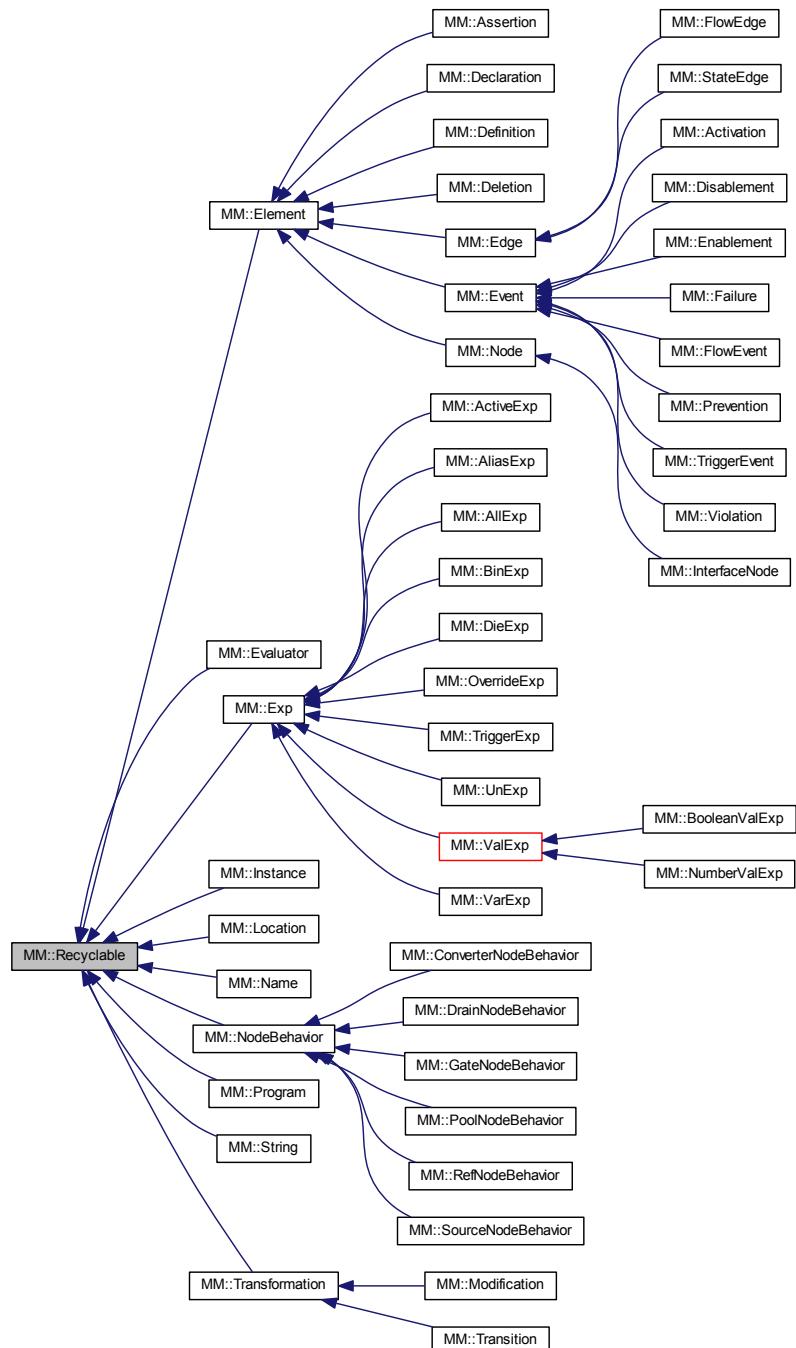
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[RangeValExp.h](#)

6.94 MM::Recyclable Class Reference

```
#include <Recyclable.h>
```

Inheritance diagram for MM::Recyclable:



Public Member Functions

- [`Recyclable \(\)`](#)
- [`virtual ~Recyclable \(\)`](#)
- [`virtual MM::TID getTypeId \(\)`](#)
- [`virtual MM::BOOLEAN instanceof \(MM::TID tid\)`](#)
- [`virtual MM::VOID recycle \(MM::Recycler *r\)`](#)
- [`virtual MM::VOID toString \(MM::String *buf\)=0`](#)

6.94.1 Constructor & Destructor Documentation

6.94.1.1 `Recyclable::Recyclable()`

6.94.1.2 `Recyclable::~Recyclable() [virtual]`

6.94.2 Member Function Documentation

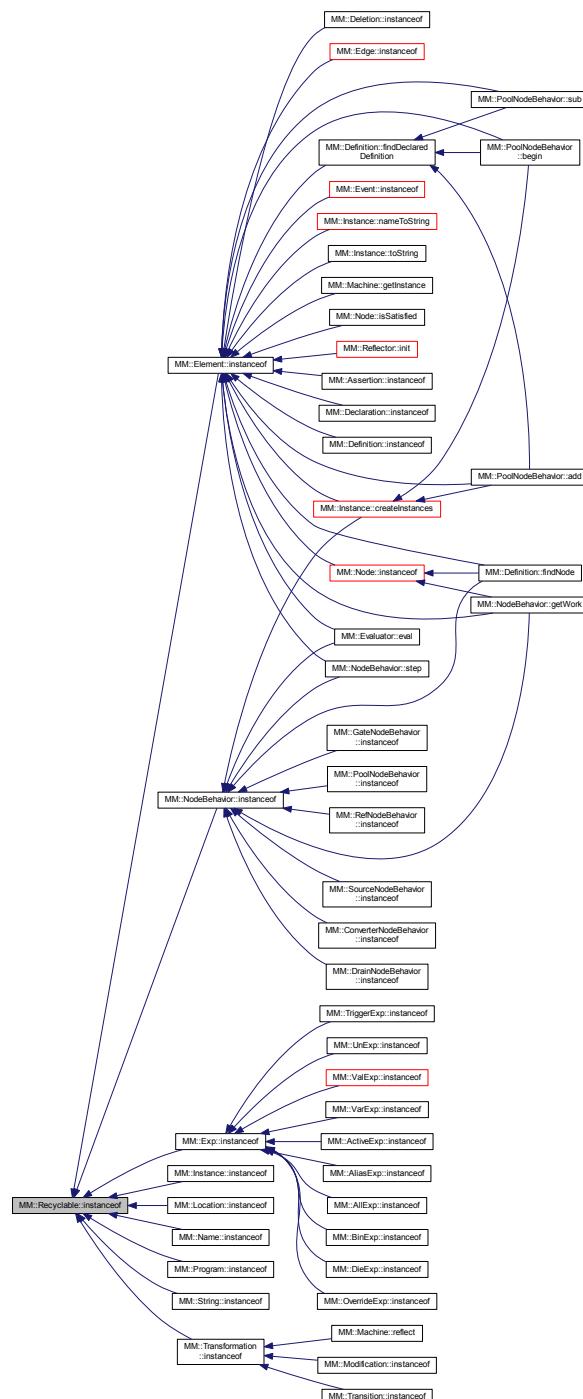
6.94.2.1 `MM::TID Recyclable::getTypeId() [virtual]`

Reimplemented in [MM::NodeBehavior](#), [MM::Instance](#), [MM::PoolNodeBehavior](#), [MM::Definition](#), [MM::Evaluator](#), [MM::FlowEvent](#), [MM::ConverterNodeBehavior](#), [MM::Declaration](#), [MM::Node](#), [MM::Location](#), [MM::DieExp](#), [MM::Assertion](#), [MM::RangeValExp](#), [MM::BinExp](#), [MM::UnExp](#), [MM::Activation](#), [MM::Disablement](#), [MM::Edge](#), [MM::Enablement](#), [MM::Failure](#), [MM::TriggerEvent](#), [MM::ActiveExp](#), [MM::OverrideExp](#), [MM::String](#), [MM::TriggerExp](#), [MM::FlowEdge](#), [MM::Name](#), [MM::OneExp](#), [MM::Prevention](#), [MM::Violation](#), [MM::AllExp](#), [MM::NumberValExp](#), [MM::BooleanValExp](#), [MM::GateNodeBehavior](#), [MM::InterfaceNode](#), [MM::StateEdge](#), [MM::VarExp](#), [MM::Deletion](#), [MM::DrainNodeBehavior](#), [MM::Element](#), [MM::Exp](#), [MM::Modification](#), [MM::RefNodeBehavior](#), [MM::Transition](#), [MM::ValExp](#), [MM::AliasExp](#), [MM::Event](#), [MM::SourceNodeBehavior](#), [MM::Transformation](#), and [MM::Program](#).

6.94.2.2 `MM::BOOLEAN Recyclable::instanceof(MM::TID tid) [virtual]`

Reimplemented in [MM::NodeBehavior](#), [MM::Instance](#), [MM::PoolNodeBehavior](#), [MM::Definition](#), [MM::Evaluator](#), [MM::FlowEvent](#), [MM::ConverterNodeBehavior](#), [MM::Declaration](#), [MM::Node](#), [MM::Location](#), [MM::DieExp](#), [MM::Assertion](#), [MM::RangeValExp](#), [MM::BinExp](#), [MM::UnExp](#), [MM::Activation](#), [MM::Disablement](#), [MM::Edge](#), [MM::Enablement](#), [MM::Failure](#), [MM::TriggerEvent](#), [MM::ActiveExp](#), [MM::OverrideExp](#), [MM::String](#), [MM::TriggerExp](#), [MM::FlowEdge](#), [MM::Name](#), [MM::OneExp](#), [MM::Prevention](#), [MM::Violation](#), [MM::AllExp](#), [MM::NumberValExp](#), [MM::BooleanValExp](#), [MM::GateNodeBehavior](#), [MM::InterfaceNode](#), [MM::StateEdge](#), [MM::VarExp](#), [MM::Deletion](#), [MM::DrainNodeBehavior](#), [MM::Element](#), [MM::Exp](#), [MM::Modification](#), [MM::RefNodeBehavior](#), [MM::Transition](#), [MM::ValExp](#), [MM::AliasExp](#), [MM::Event](#), [MM::SourceNodeBehavior](#), [MM::Transformation](#), and [MM::Program](#).

Here is the caller graph for this function:



6.94.2.3 `MM::VOID Recyclable::recycle(MM::Recycler * r) [virtual]`

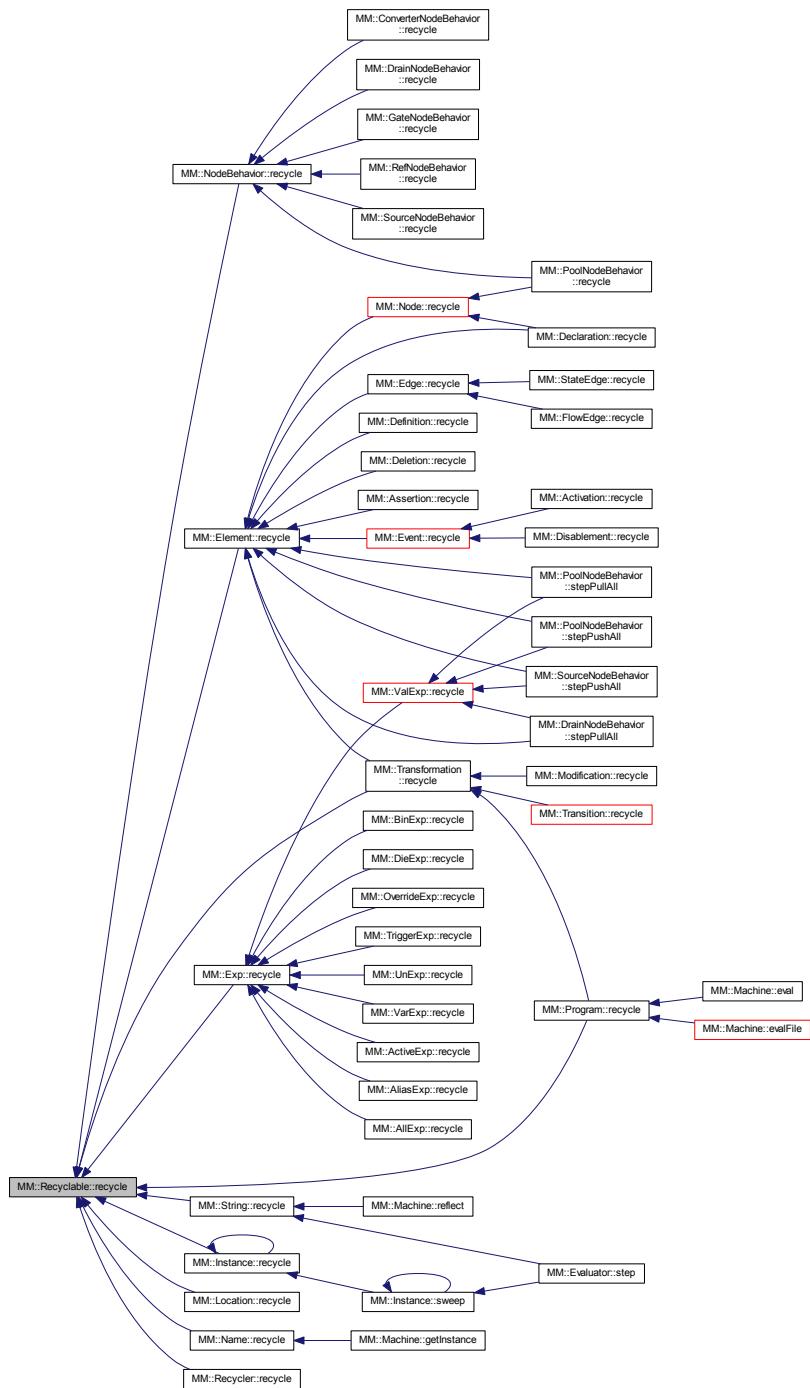
Reimplemented in `MM::NodeBehavior`, `MM::Instance`, `MM::PoolNodeBehavior`, `MM::Definition`, `MM::Evaluator`, `MM::FlowEvent`, `MM::ConverterNodeBehavior`, `MM::Declaration`, `MM::Location`, `MM::Node`, `MM::DieExp`, `MM::Assertion`, `MM::RangeValExp`, `MM::BinExp`, `MM::UnExp`, `MM::Activation`, `MM::Disablement`, `MM::Edge`, `MM::Enablement`, `MM::Failure`, `MM::TriggerEvent`, `MM::ActiveExp`, `MM::BooleanValExp`, `MM::OverrideExp`, `MM::String`, `MM::TriggerExp`, `MM::FlowEdge`, `MM::Name`, `MM::OneExp`, `MM::Prevention`, `MM::Violation`, `MM::AllExp`, `MM::`.

[NumberValExp](#), [MM::GateNodeBehavior](#), [MM::InterfaceNode](#), [MM::StateEdge](#), [MM::VarExp](#), [MM::Deletion](#), [MM::DrainNodeBehavior](#), [MM::Element](#), [MM::Exp](#), [MM::Modification](#), [MM::RefNodeBehavior](#), [MM::Transition](#), [MM::ValExp](#), [MM::AliasExp](#), [MM::Event](#), [MM::SourceNodeBehavior](#), [MM::Transformation](#), and [MM::Program](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.94.2.4 virtual MM::VOID MM::Recyclable::toString (MM::String * buf) [pure virtual]

Implemented in [MM::Instance](#), [MM::PoolNodeBehavior](#), [MM::ConverterNodeBehavior](#), [MM::Node](#), [MM::Definition](#), [MM::RefNodeBehavior](#), [MM::Name](#), [MM::DrainNodeBehavior](#), [MM::SourceNodeBehavior](#), [MM::GateNodeBehavior](#), [MM::FlowEvent](#), [MM::Declaration](#), [MM::Evaluator](#), [MM::InterfaceNode](#), [MM::String](#), [MM::Location](#), [MM::BinExp](#), [MM::RangeValExp](#), [MM::Edge](#), [MM::Assertion](#), [MM::DieExp](#), [MM::UnExp](#), [MM::Element](#), [MM::Disablement](#), [MM::Enablement](#), [MM::Failure](#), [MM::TriggerEvent](#), [MM::Activation](#), [MM::Prevention](#), [MM::Event](#),

[MM::NumberValExp](#), [MM::ActiveExp](#), [MM::BooleanValExp](#), [MM::OverrideExp](#), [MM::Violation](#), [MM::OneExp](#), [MM::StateEdge](#), [MM::TriggerExp](#), [MM::FlowEdge](#), [MM::AllExp](#), [MM::VarExp](#), [MM::Deletion](#), [MM::Transformation](#), [MM::ValExp](#), [MM::Exp](#), [MM::Modification](#), [MM::Transition](#), [MM::AliasExp](#), and [MM::Program](#).

The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Recyclable.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Recyclable.cpp](#)

6.95 Recyclable Class Reference

Objects that are recyclable can be recycled by their recycler, which is the factory in which the object was created.

```
#include <Recyclable.h>
```

6.95.1 Detailed Description

Objects that are recyclable can be recycled by their recycler, which is the factory in which the object was created.

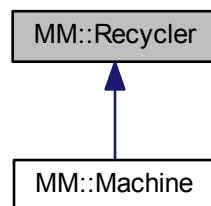
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Recyclable.h](#)

6.96 MM::Recycler Class Reference

```
#include <Recycler.h>
```

Inheritance diagram for MM::Recycler:



Public Member Functions

- virtual [MM::TID getTypeId \(\)](#)
- virtual [MM::BOOLEAN instanceof \(MM::TID tid\)](#)
- [MM::VOID create \(MM::Recyclable *r\)](#)
- [MM::VOID recycle \(MM::Recyclable *r\)](#)
- [MM::VOID uncreate \(MM::Recyclable *r\)](#)

Static Public Member Functions

- static MM::VOID uncreate (MM::CHAR *str)
 - static MM::CHAR * createBuffer (MM::UINT32 size)

Static Public Attributes

- static const MM::CHAR * TYPE_STR []

Protected Member Functions

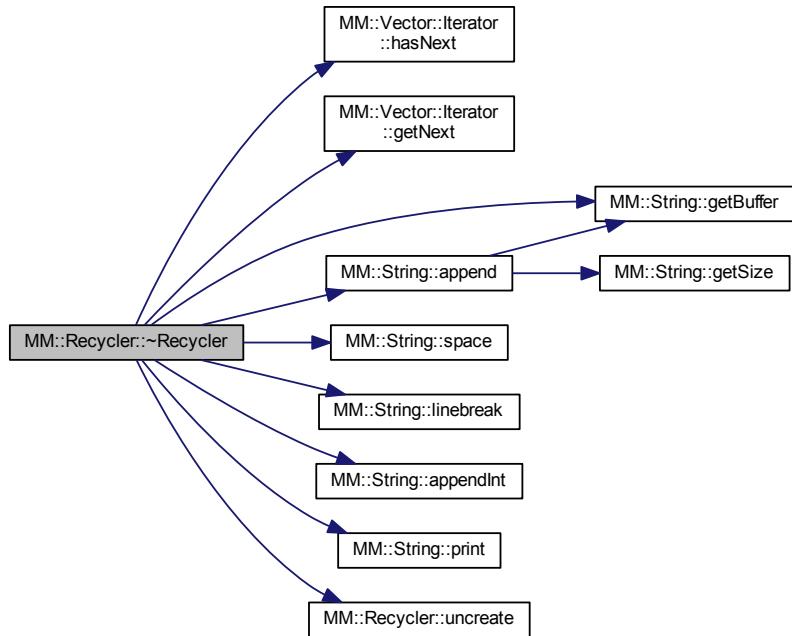
- Recycler ()
 - ~Recycler ()

6.96.1 Constructor & Destructor Documentation

6.96.1.1 Recycler::Recycler() [protected]

6.96.1.2 Recycler::~Recycler() [protected]

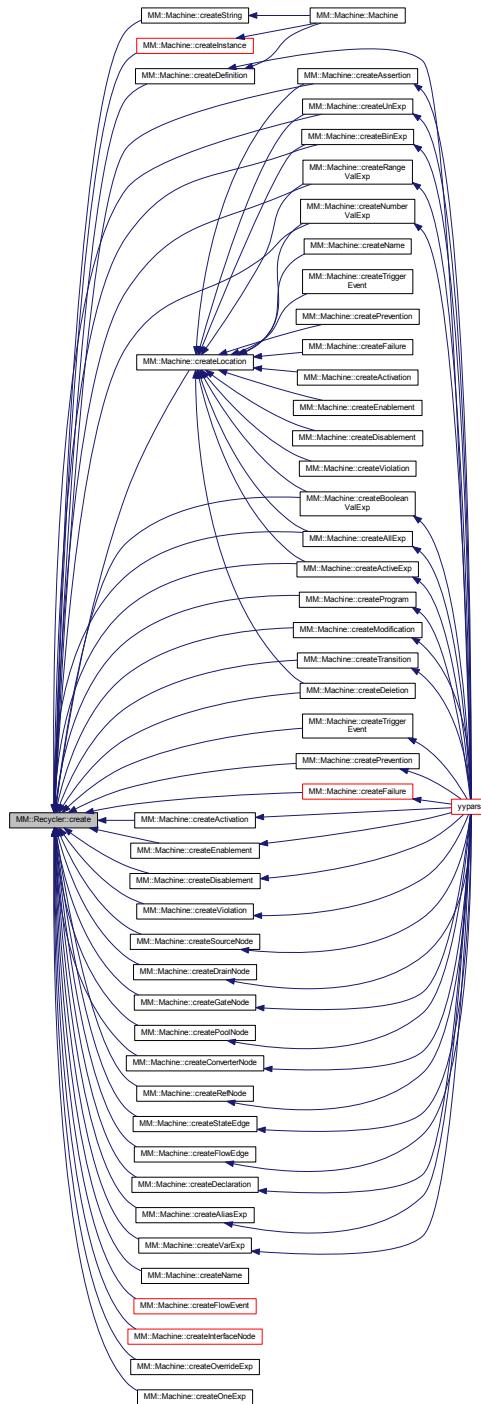
Here is the call graph for this function:



6.96.2 Member Function Documentation

6.96.2.1 MM::VOID Recycler::create(MM::Recyclable * r)

Here is the caller graph for this function:



6.96.2.2 MM::CHAR * Recycler::createBuffer (MM::UINT32 size) [static]

Here is the caller graph for this function:



6.96.2.3 MM::TID Recycler::getTypeld () [virtual]

Reimplemented in [MM::Machine](#).

6.96.2.4 MM::BOOLEAN Recycler::instanceof (MM::TID tid) [virtual]

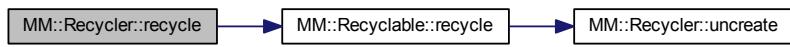
Reimplemented in [MM::Machine](#).

Here is the caller graph for this function:



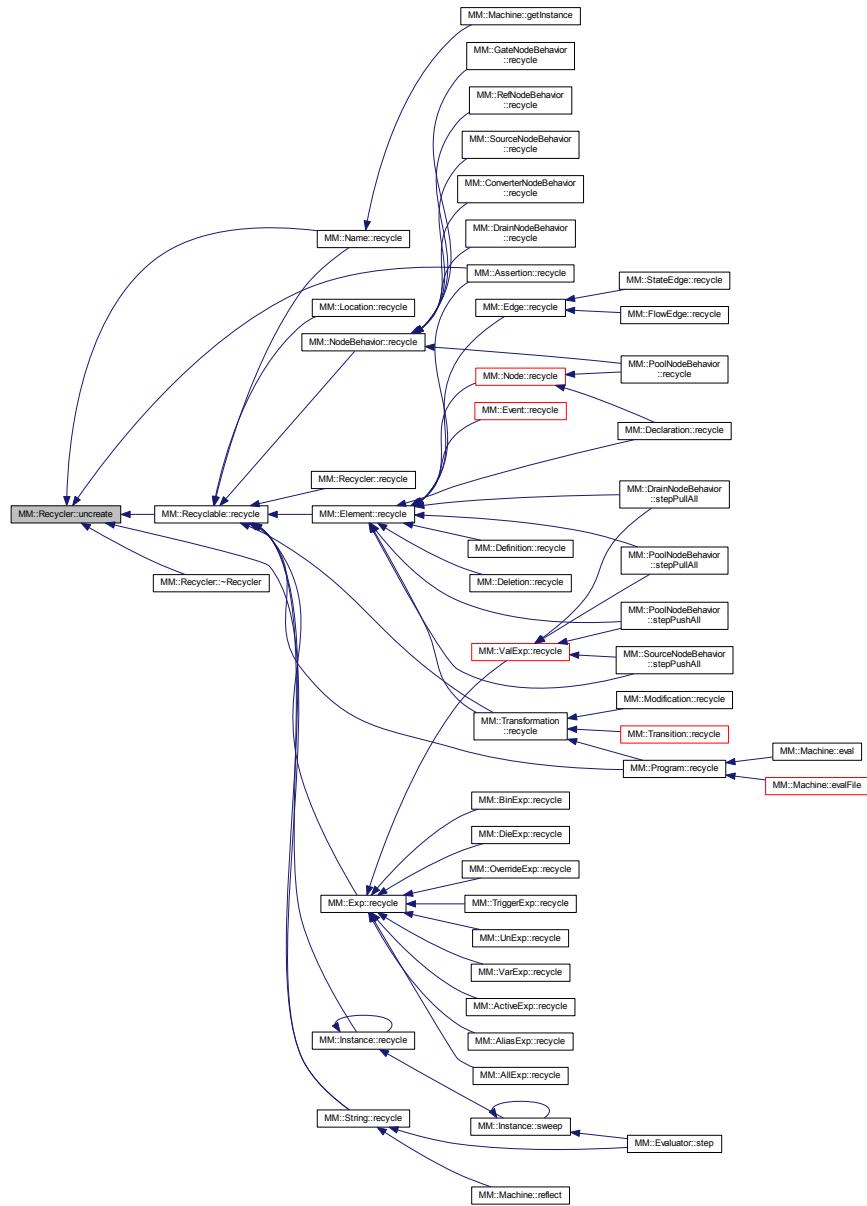
6.96.2.5 MM::VOID Recycler::recycle (MM::Recyclable * r)

Here is the call graph for this function:



6.96.2.6 MM::VOID Recycler::uncreate (MM::Recyclable * r)

Here is the caller graph for this function:



6.96.2.7 MM::VOID Recycler::uncreate (MM::CHAR * str) [static]

6.96.3 Member Data Documentation

6.96.3.1 const MM::CHAR * Recycler::TYPE_STR [static]

The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Recycler.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Recycler.cpp](#)

6.97 Recycler Class Reference

The [Recycler](#) abstraction is used to recycle recyclable objects.

```
#include <Recycler.h>
```

6.97.1 Detailed Description

The [Recycler](#) abstraction is used to recycle recyclable objects.

The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Recycler.h](#)

6.98 Reflector Class Reference

The [Reflector](#) reflects program changes to definitions and instances.

```
#include <Reflector.h>
```

6.98.1 Detailed Description

The [Reflector](#) reflects program changes to definitions and instances.

The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Reflector.h](#)

6.99 MM::Reflector Class Reference

```
#include <Reflector.h>
```

Public Member Functions

- [Reflector \(MM::Machine *m\)](#)
- [~Reflector \(\)](#)
- [MM::TID getTypeld \(\)](#)
- [MM::BOOLEAN instanceof \(MM::TID tid\)](#)
- [MM::Definition * getDefinition \(\)](#)
- [MM::Instance * getInstance \(\)](#)
- [MM::VOID addElement \(MM::Definition *def, MM::Element *element\)](#)
- [MM::VOID removeElement \(MM::Definition *def, MM::Element *element\)](#)
- [MM::VOID merge \(MM::Modification *modification\)](#)
- [MM::VOID init \(MM::Definition *def\)](#)
- [MM::VOID init \(MM::Node *node\)](#)
- [MM::VOID_deinit \(MM::Definition *def, MM::Edge *edge\)](#)

6.99.1 Constructor & Destructor Documentation

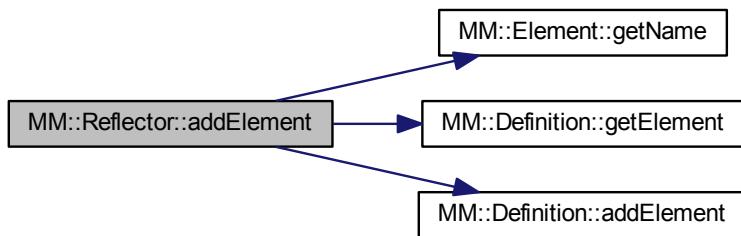
6.99.1.1 Reflector::Reflector (`MM::Machine * m`)

6.99.1.2 Reflector::~Reflector ()

6.99.2 Member Function Documentation

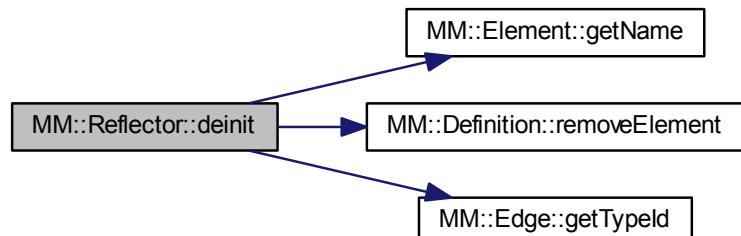
6.99.2.1 MM::VOID Reflector::addElement (`MM::Definition * def, MM::Element * element`)

Here is the call graph for this function:



6.99.2.2 MM::VOID Reflector::deinit (`MM::Definition * def, MM::Edge * edge`)

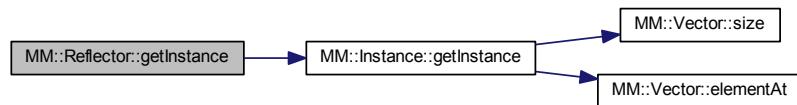
Here is the call graph for this function:



6.99.2.3 `MM::Definition * Reflector::getDefinition ()`

6.99.2.4 MM::Instance * Reflector::getInstance()

Here is the call graph for this function:



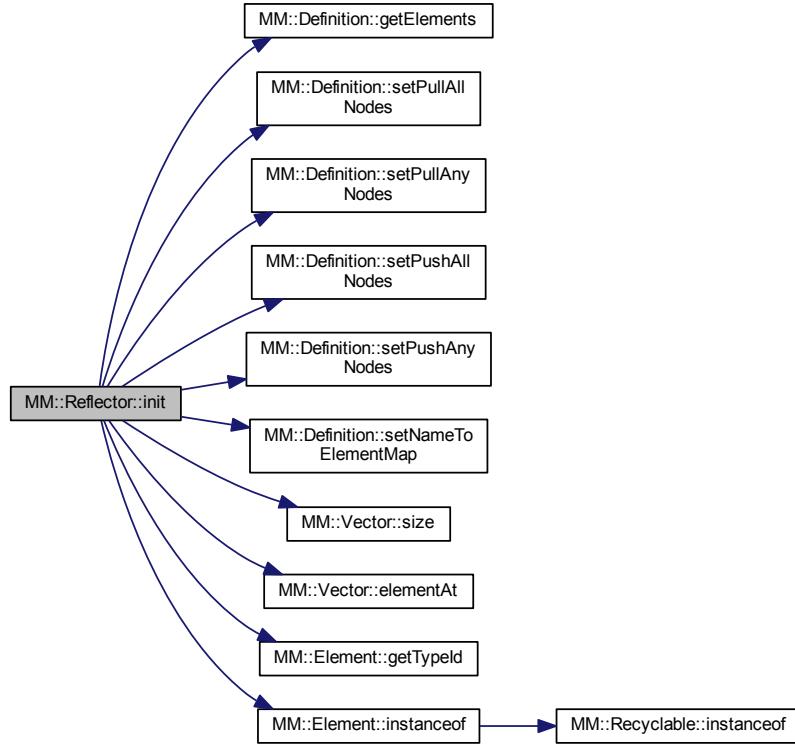
Here is the caller graph for this function:



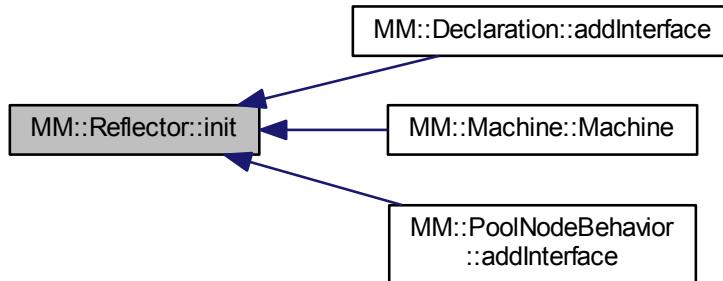
6.99.2.5 MM::TID Reflector::getTypeId()

6.99.2.6 MM::VOID Reflector::init (MM::Definition * def)

Here is the call graph for this function:

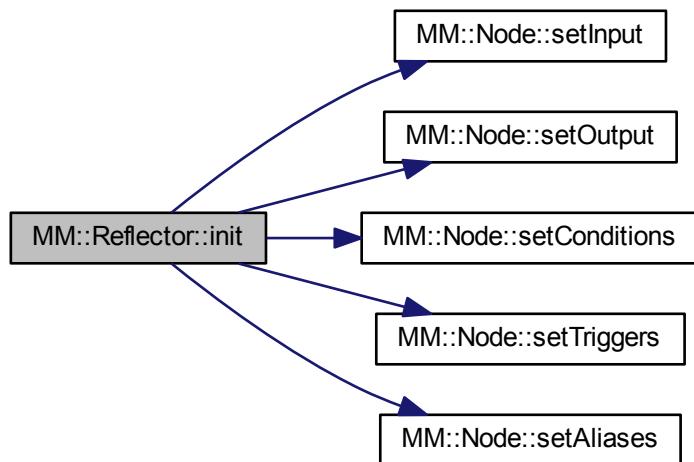


Here is the caller graph for this function:



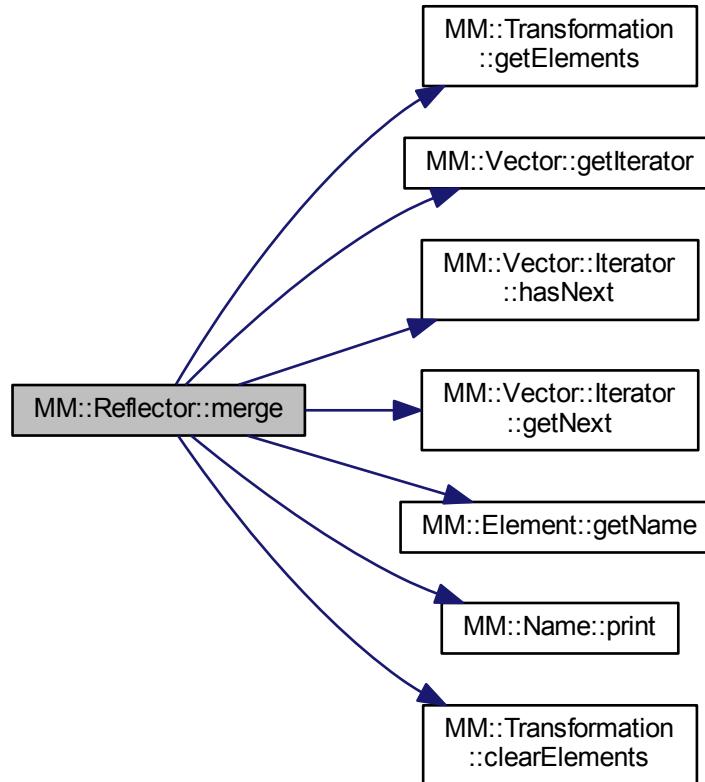
6.99.2.7 MM::VOID Reflector::init (MM::Node * node)

Here is the call graph for this function:

**6.99.2.8 MM::BOOLEAN Reflector::instanceof (MM::TID tid)**

6.99.2.9 MM::VOID Reflector::merge (MM::Modification * *modification*)

Here is the call graph for this function:

6.99.2.10 MM::VOID Reflector::removeElement (MM::Definition * *def*, MM::Element * *element*)

The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Reflector.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Reflector.cpp](#)

6.100 RefNodeBehavior Class Reference

The [RefNodeBehavior](#) abstraction expresses that the behavior of a node is defined elsewhere.

```
#include <RefNodeBehavior.h>
```

6.100.1 Detailed Description

The [RefNodeBehavior](#) abstraction expresses that the behavior of a node is defined elsewhere.

Note

Strategy

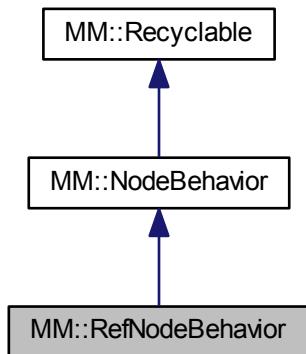
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[RefNodeBehavior.h](#)

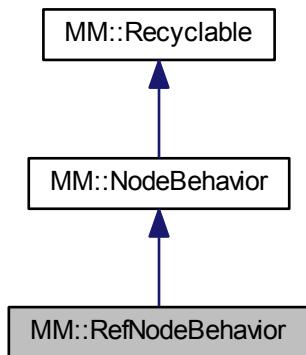
6.101 MM::RefNodeBehavior Class Reference

```
#include <RefNodeBehavior.h>
```

Inheritance diagram for MM::RefNodeBehavior:



Collaboration diagram for MM::RefNodeBehavior:



Public Member Functions

- `RefNodeBehavior (MM::NodeBehavior::IO io)`
- `~RefNodeBehavior ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::Edge * getAlias ()`
- `MM::VOID setAlias (MM::Edge *edge)`
- `MM::Node * getReference ()`
- `MM::UINT32 getCreateMessage ()`
- `MM::UINT32 getUpdateMessage ()`
- `MM::UINT32 getDeleteMessage ()`
- `MM::VOID step (MM::Node *n, MM::Instance *i, MM::Machine *m, MM::Transition *t)`
- `MM::VOID stepPullAny (MM::Node *node, MM::Instance *i, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)`
- `MM::VOID stepPushAny (MM::Node *node, MM::Instance *i, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)`
- `MM::VOID stepPullAll (MM::Node *node, MM::Instance *i, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)`
- `MM::VOID begin (MM::Instance *i, MM::Machine *m, MM::Node *n)`
- `MM::VOID end (MM::Instance *i, MM::Machine *m, MM::Node *n)`
- `MM::VOID change (MM::Instance *i, MM::Machine *m, MM::Node *n)`
- `MM::VOID add (MM::Instance *i, MM::Machine *m, MM::Node *n, MM::UINT32 amount)`
- `MM::VOID sub (MM::Instance *i, MM::Machine *m, MM::Node *n, MM::UINT32 amount)`
- `MM::UINT32 getCapacity (MM::Instance *i, MM::Node *n)`
- `MM::UINT32 getResources (MM::Instance *i, MM::Node *n)`
- `MM::BOOLEAN hasCapacity (MM::Instance *i, MM::Node *n, MM::UINT32 amount)`
- `MM::BOOLEAN hasResources (MM::Instance *i, MM::Node *n, MM::UINT32 amount)`
- `MM::VOID doTriggers (MM::Instance *i, MM::Node *n)`
- `MM::VOID toString (MM::String *buf)`
- `MM::VOID toString (MM::String *buf, MM::Name *name)`

Additional Inherited Members

6.101.1 Constructor & Destructor Documentation

6.101.1.1 RefNodeBehavior::RefNodeBehavior (`MM::NodeBehavior::IO io`)

a RefNode must have exactly one alias edge that defines its behavior

6.101.1.2 RefNodeBehavior::~RefNodeBehavior ()

6.101.2 Member Function Documentation

6.101.2.1 MM::VOID RefNodeBehavior::add (`MM::Instance * i, MM::Machine * m, MM::Node * n, MM::UINT32 amount`) [virtual]

Implements [MM::NodeBehavior](#).

6.101.2.2 **MM::VOID RefNodeBehavior::begin (MM::Instance * i, MM::Machine * m, MM::Node * n)**
 [virtual]

Implements [MM::NodeBehavior](#).

6.101.2.3 **MM::VOID RefNodeBehavior::change (MM::Instance * i, MM::Machine * m, MM::Node * n)**
 [virtual]

Implements [MM::NodeBehavior](#).

6.101.2.4 **MM::VOID RefNodeBehavior::doTriggers (MM::Instance * i, MM::Node * n)**

6.101.2.5 **MM::VOID RefNodeBehavior::end (MM::Instance * i, MM::Machine * m, MM::Node * n)** [virtual]

Implements [MM::NodeBehavior](#).

6.101.2.6 **MM::Edge * RefNodeBehavior::getAlias ()**

6.101.2.7 **MM::UINT32 RefNodeBehavior::getCapacity (MM::Instance * i, MM::Node * n)** [virtual]

Implements [MM::NodeBehavior](#).

6.101.2.8 **MM::UINT32 RefNodeBehavior::getCreateMessage ()** [virtual]

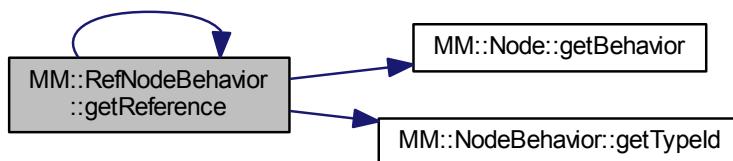
Implements [MM::NodeBehavior](#).

6.101.2.9 **MM::UINT32 RefNodeBehavior::getDeleteMessage ()** [virtual]

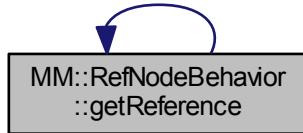
Implements [MM::NodeBehavior](#).

6.101.2.10 **MM::Node * RefNodeBehavior::getReference ()**

Here is the call graph for this function:



Here is the caller graph for this function:



6.101.2.11 **MM::UINT32 RefNodeBehavior::getResources (MM::Instance * *i*, MM::Node * *n*) [virtual]**

Implements [MM::NodeBehavior](#).

6.101.2.12 **MM::TID RefNodeBehavior::getTypeld() [virtual]**

Reimplemented from [MM::NodeBehavior](#).

6.101.2.13 **MM::UINT32 RefNodeBehavior::getUpdateMessage() [virtual]**

Implements [MM::NodeBehavior](#).

6.101.2.14 **MM::BOOLEAN RefNodeBehavior::hasCapacity (MM::Instance * *i*, MM::Node * *n*, MM::UINT32 *amount*) [virtual]**

Implements [MM::NodeBehavior](#).

6.101.2.15 **MM::BOOLEAN RefNodeBehavior::hasResources (MM::Instance * *i*, MM::Node * *n*, MM::UINT32 *amount*) [virtual]**

Implements [MM::NodeBehavior](#).

6.101.2.16 **MM::BOOLEAN RefNodeBehavior::instanceof (MM::TID *tid*) [virtual]**

Reimplemented from [MM::NodeBehavior](#).

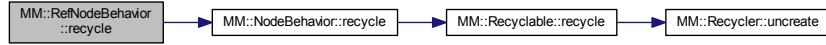
Here is the call graph for this function:



6.101.2.17 MM::VOID RefNodeBehavior::recycle (MM::Recycler * r) [virtual]

Reimplemented from [MM::NodeBehavior](#).

Here is the call graph for this function:



6.101.2.18 MM::VOID RefNodeBehavior::setAlias (MM::Edge * edge)

6.101.2.19 MM::VOID RefNodeBehavior::step (MM::Node * n, MM::Instance * i, MM::Machine * m, MM::Transition * t) [virtual]

Reimplemented from [MM::NodeBehavior](#).

6.101.2.20 MM::VOID RefNodeBehavior::stepPullAll (MM::Node * node, MM::Instance * i, MM::Vector<MM::NodeWorkItem * > * work, MM::Machine * m, MM::Transition * tr) [virtual]

Implements [MM::NodeBehavior](#).

6.101.2.21 MM::VOID RefNodeBehavior::stepPullAny (MM::Node * node, MM::Instance * i, MM::Vector<MM::NodeWorkItem * > * work, MM::Machine * m, MM::Transition * tr) [virtual]

Reimplemented from [MM::NodeBehavior](#).

6.101.2.22 MM::VOID RefNodeBehavior::stepPushAll (MM::Node * node, MM::Instance * i, MM::Vector<MM::NodeWorkItem * > * work, MM::Machine * m, MM::Transition * tr) [virtual]

Implements [MM::NodeBehavior](#).

6.101.2.23 MM::VOID RefNodeBehavior::stepPushAny (MM::Node * node, MM::Instance * i, MM::Vector<MM::NodeWorkItem * > * work, MM::Machine * m, MM::Transition * tr) [virtual]

Reimplemented from [MM::NodeBehavior](#).

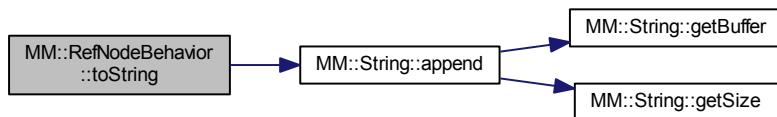
6.101.2.24 MM::VOID RefNodeBehavior::sub (MM::Instance * i, MM::Machine * m, MM::Node * n, MM::UINT32 amount) [virtual]

Implements [MM::NodeBehavior](#).

6.101.2.25 MM::VOID RefNodeBehavior::toString (MM::String * buf) [virtual]

Implements [MM::Recyclable](#).

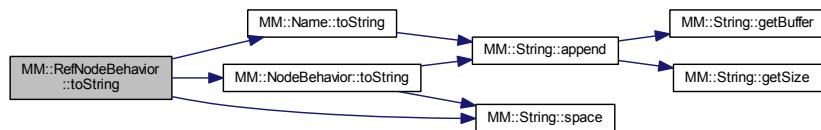
Here is the call graph for this function:



6.101.2.26 MM::VOID RefNodeBehavior::toString (MM::String * buf, MM::Name * name) [virtual]

Reimplemented from [MM::NodeBehavior](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[RefNodeBehavior.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[RefNodeBehavior.cpp](#)

6.102 SourceNodeBehavior Class Reference

The [SourceNodeBehavior](#) abstraction defines the behavior of source nodes.

```
#include <SourceNodeBehavior.h>
```

6.102.1 Detailed Description

The [SourceNodeBehavior](#) abstraction defines the behavior of source nodes.

Note

Strategy

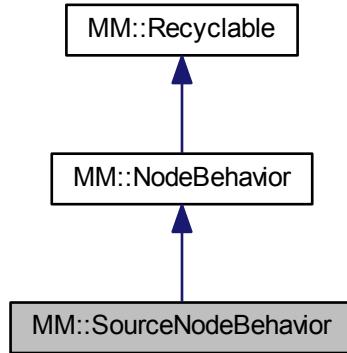
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[SourceNodeBehavior.h](#)

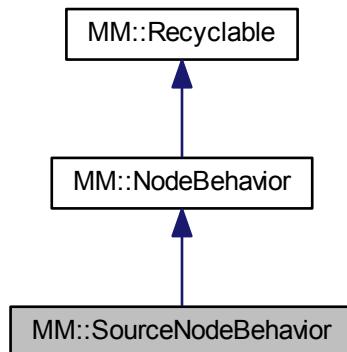
6.103 MM::SourceNodeBehavior Class Reference

```
#include <SourceNodeBehavior.h>
```

Inheritance diagram for MM::SourceNodeBehavior:



Collaboration diagram for MM::SourceNodeBehavior:



Public Member Functions

- [SourceNodeBehavior \(MM::NodeBehavior::IO io, MM::NodeBehavior::When when\)](#)
- [~SourceNodeBehavior \(\)](#)
- [MM::VOID recycle \(MM::Recycler *r\)](#)
- [MM::TID getTypeld \(\)](#)
- [MM::BOOLEAN instanceof \(MM::TID tid\)](#)
- [MM::UINT32 getCreateMessage \(\)](#)
- [MM::UINT32 getUpdateMessage \(\)](#)
- [MM::UINT32 getDeleteMessage \(\)](#)
- [MM::VOID stepPullAny \(MM::Node *node, MM::Instance *i, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr\)](#)

- MM::VOID stepPullAll (MM::Node *node, MM::Instance *i, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)
- MM::VOID stepPushAll (MM::Node *node, MM::Instance *i, MM::Vector< MM::NodeWorkItem * > *work, MM::Machine *m, MM::Transition *tr)
- MM::VOID begin (MM::Instance *i, MM::Machine *m, MM::Node *n)
- MM::VOID end (MM::Instance *i, MM::Machine *m, MM::Node *n)
- MM::VOID change (MM::Instance *i, MM::Machine *m, MM::Node *n)
- MM::VOID add (MM::Instance *i, MM::Machine *m, MM::Node *n, MM::UINT32 amount)
- MM::VOID sub (MM::Instance *i, MM::Machine *m, MM::Node *n, MM::UINT32 amount)
- MM::UINT32 getCapacity (MM::Instance *i, MM::Node *n)
- MM::UINT32 getResources (MM::Instance *i, MM::Node *n)
- MM::BOOLEAN hasCapacity (MM::Instance *i, MM::Node *n, MM::UINT32 amount)
- MM::BOOLEAN hasResources (MM::Instance *i, MM::Node *n, MM::UINT32 amount)
- MM::VOID toString (MM::String *buf)
- MM::VOID toString (MM::String *buf, MM::Name *name)

Additional Inherited Members

6.103.1 Constructor & Destructor Documentation

6.103.1.1 SourceNodeBehavior::SourceNodeBehavior (MM::NodeBehavior::IO *io*, MM::NodeBehavior::When *when*)

6.103.1.2 SourceNodeBehavior::~SourceNodeBehavior ()

6.103.2 Member Function Documentation

6.103.2.1 MM::VOID SourceNodeBehavior::add (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*, MM::UINT32 *amount*) [virtual]

Implements [MM::NodeBehavior](#).

6.103.2.2 MM::VOID SourceNodeBehavior::begin (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*) [virtual]

Implements [MM::NodeBehavior](#).

6.103.2.3 MM::VOID SourceNodeBehavior::change (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*) [virtual]

Implements [MM::NodeBehavior](#).

6.103.2.4 MM::VOID SourceNodeBehavior::end (MM::Instance * *i*, MM::Machine * *m*, MM::Node * *n*) [virtual]

Implements [MM::NodeBehavior](#).

6.103.2.5 MM::UINT32 SourceNodeBehavior::getCapacity (MM::Instance * *i*, MM::Node * *n*) [virtual]

Implements [MM::NodeBehavior](#).

6.103.2.6 MM::UINT32 SourceNodeBehavior::getCreateMessage () [virtual]

Implements [MM::NodeBehavior](#).

6.103.2.7 **MM::UINT32 SourceNodeBehavior::getDeleteMessage() [virtual]**

Implements [MM::NodeBehavior](#).

6.103.2.8 **MM::UINT32 SourceNodeBehavior::getResources(MM::Instance * i, MM::Node * n) [virtual]**

Implements [MM::NodeBehavior](#).

6.103.2.9 **MM::TID SourceNodeBehavior::getTypeId() [virtual]**

Reimplemented from [MM::NodeBehavior](#).

6.103.2.10 **MM::UINT32 SourceNodeBehavior::getUpdateMessage() [virtual]**

Implements [MM::NodeBehavior](#).

6.103.2.11 **MM::BOOLEAN SourceNodeBehavior::hasCapacity(MM::Instance * i, MM::Node * n, MM::UINT32 amount) [virtual]**

Implements [MM::NodeBehavior](#).

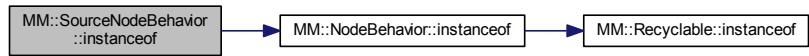
6.103.2.12 **MM::BOOLEAN SourceNodeBehavior::hasResources(MM::Instance * i, MM::Node * n, MM::UINT32 amount) [virtual]**

Implements [MM::NodeBehavior](#).

6.103.2.13 **MM::BOOLEAN SourceNodeBehavior::instanceof(MM::TID tid) [virtual]**

Reimplemented from [MM::NodeBehavior](#).

Here is the call graph for this function:



6.103.2.14 **MM::VOID SourceNodeBehavior::recycle(MM::Recycler * r) [virtual]**

Reimplemented from [MM::NodeBehavior](#).

Here is the call graph for this function:



6.103.2.15 **MM::VOID** SourceNodeBehavior::stepPullAll (**MM::Node** * *node*, **MM::Instance** * *i*, **MM::Vector<**
MM::NodeWorkItem * > * *work*, **MM::Machine** * *m*, **MM::Transition** * *tr*) [virtual]

Implements [MM::NodeBehavior](#).

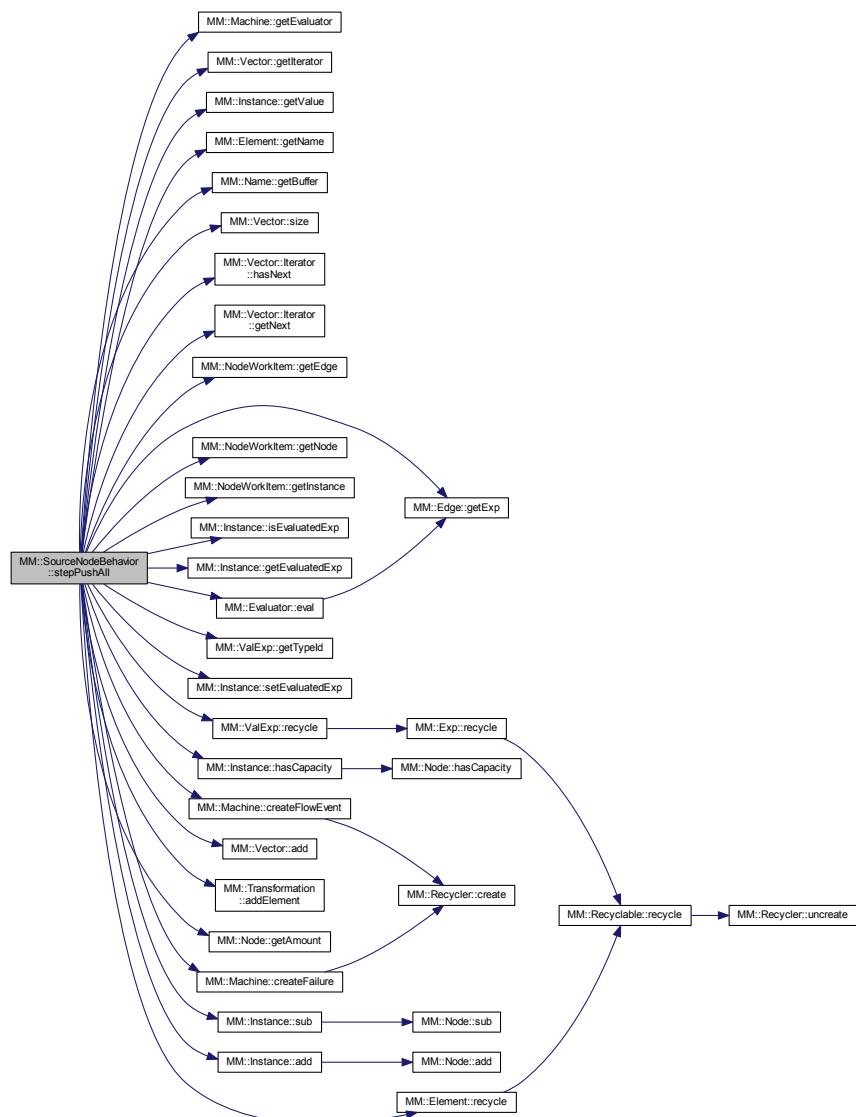
6.103.2.16 **MM::VOID** SourceNodeBehavior::stepPullAny (**MM::Node** * *node*, **MM::Instance** * *i*, **MM::Vector<**
MM::NodeWorkItem * > * *work*, **MM::Machine** * *m*, **MM::Transition** * *tr*) [virtual]

Reimplemented from [MM::NodeBehavior](#).

6.103.2.17 **MM::VOID** SourceNodeBehavior::stepPushAll (**MM::Node** * *node*, **MM::Instance** * *i*, **MM::Vector<**
MM::NodeWorkItem * > * *work*, **MM::Machine** * *m*, **MM::Transition** * *tr*) [virtual]

Implements [MM::NodeBehavior](#).

Here is the call graph for this function:



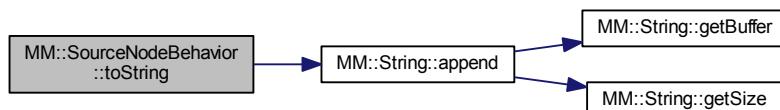
6.103.2.18 **MM::VOID** `SourceNodeBehavior::sub (MM::Instance * i, MM::Machine * m, MM::Node * n, MM::UINT32 amount) [virtual]`

Implements [MM::NodeBehavior](#).

6.103.2.19 **MM::VOID** `SourceNodeBehavior::toString (MM::String * buf) [virtual]`

Implements [MM::Recyclable](#).

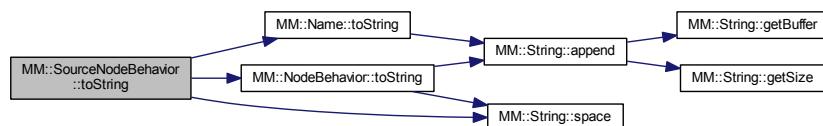
Here is the call graph for this function:



6.103.2.20 **MM::VOID** `SourceNodeBehavior::toString (MM::String * buf, MM::Name * name) [virtual]`

Reimplemented from [MM::NodeBehavior](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[SourceNodeBehavior.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[SourceNodeBehavior.cpp](#)

6.104 StateEdge Class Reference

The [StateEdge](#) abstraction defines state edges, which are conditions, aliases or triggers.

```
#include <StateEdge.h>
```

6.104.1 Detailed Description

The [StateEdge](#) abstraction defines state edges, which are conditions, aliases or triggers.

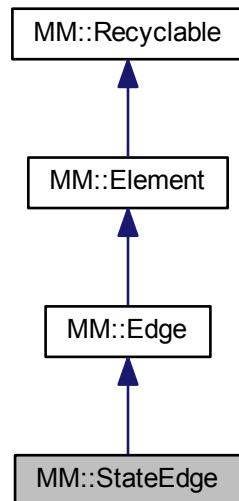
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[StateEdge.h](#)

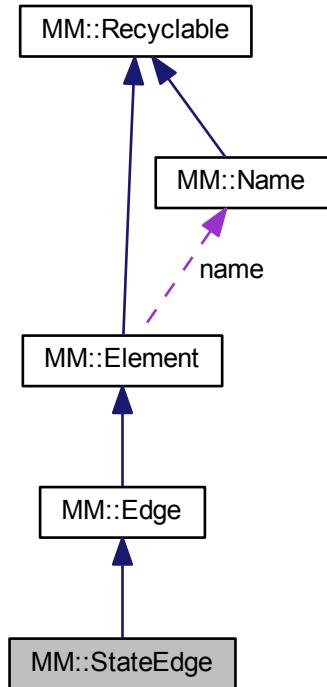
6.105 MM::StateEdge Class Reference

```
#include <StateEdge.h>
```

Inheritance diagram for MM::StateEdge:



Collaboration diagram for MM::StateEdge:



Public Member Functions

- `StateEdge (MM::Name *name, MM::Name *src, MM::Exp *exp, MM::Name *tgt)`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeId ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::BOOLEAN isTrigger ()`
- `MM::BOOLEAN isAlias ()`
- `MM::BOOLEAN isCondition ()`
- `MM::VOID toString (MM::String *buf)`
- `MM::VOID toString (MM::String *buf, MM::UINT32 indent)`

Protected Member Functions

- `~StateEdge ()`

Additional Inherited Members

6.105.1 Constructor & Destructor Documentation

6.105.1.1 `StateEdge::~StateEdge ()` [protected]

6.105.1.2 StateEdge::StateEdge (**MM::Name** * *name*, **MM::Name** * *src*, **MM::Exp** * *exp*, **MM::Name** * *tgt*)

6.105.2 Member Function Documentation

6.105.2.1 **MM::TID** StateEdge::getTypeId () [virtual]

Reimplemented from [MM::Edge](#).

6.105.2.2 **MM::BOOLEAN** StateEdge::instanceof (**MM::TID** *tid*) [virtual]

Reimplemented from [MM::Edge](#).

Here is the call graph for this function:



6.105.2.3 **MM::BOOLEAN** StateEdge::isAlias ()

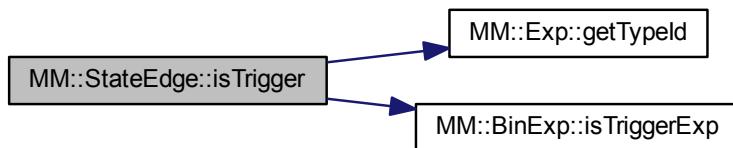
Here is the call graph for this function:



6.105.2.4 **MM::BOOLEAN** StateEdge::isCondition ()

6.105.2.5 **MM::BOOLEAN** StateEdge::isTrigger ()

Here is the call graph for this function:



6.105.2.6 MM::VOID StateEdge::recycle (MM::Recycler * r) [virtual]

Reimplemented from [MM::Edge](#).

Here is the call graph for this function:



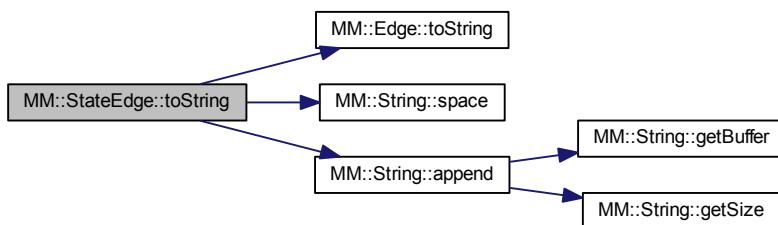
6.105.2.7 MM::VOID StateEdge::toString (MM::String * buf) [virtual]

Implements [MM::Edge](#).

6.105.2.8 MM::VOID StateEdge::toString (MM::String * buf, MM::UINT32 indent) [virtual]

Reimplemented from [MM::Edge](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[StateEdge.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[StateEdge.cpp](#)

6.106 String Class Reference

The [String](#) abstraction defines bounded [String](#) buffers.

```
#include <String.h>
```

6.106.1 Detailed Description

The [String](#) abstraction defines bounded [String](#) buffers.

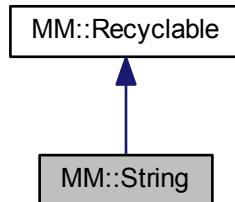
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[String.h](#)

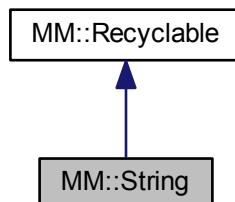
6.107 MM::String Class Reference

```
#include <String.h>
```

Inheritance diagram for MM::String:



Collaboration diagram for MM::String:



Public Member Functions

- [String \(MM::CHAR *buf, MM::UINT32 size\)](#)
- [~String \(\)](#)
- [MM::VOID recycle \(MM::Recycler *r\)](#)
- [MM::TID getTypeld \(\)](#)
- [MM::BOOLEAN instanceof \(TID tid\)](#)
- [MM::UINT32 getSize \(\)](#)
- [MM::UINT32 getUsed \(\)](#)
- [MM::CHAR * getBuffer \(\)](#)
- [MM::VOID append \(MM::String *str\)](#)
- [MM::VOID append \(MM::CHAR *buf, MM::UINT32 len\)](#)
- [MM::VOID append \(MM::CHAR c\)](#)
- [MM::VOID appendInt \(MM::INT32 val\)](#)
- [MM::VOID space \(\)](#)
- [MM::VOID space \(MM::UINT32 amount\)](#)
- [MM::VOID linebreak \(\)](#)
- [MM::VOID clear \(\)](#)

- MM::VOID print ()

- MM::VOID toString (MM::String *buf)

6.107.1 Constructor & Destructor Documentation

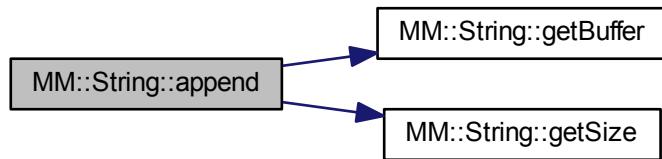
6.107.1.1 String::String (MM::CHAR * buf, MM::UINT32 size)

6.107.1.2 String::~String ()

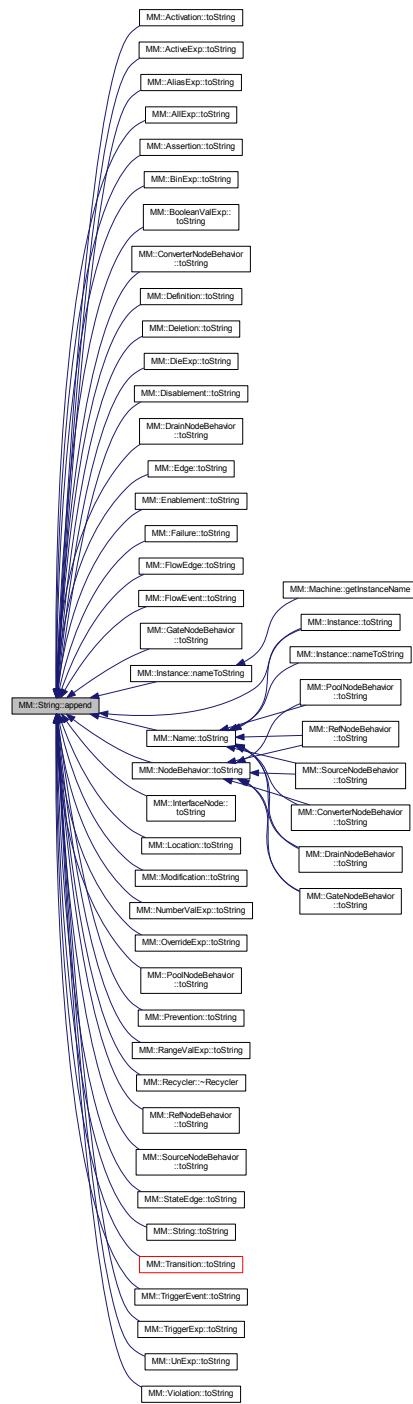
6.107.2 Member Function Documentation

6.107.2.1 MM::VOID String::append (MM::String * str)

Here is the call graph for this function:



Here is the caller graph for this function:

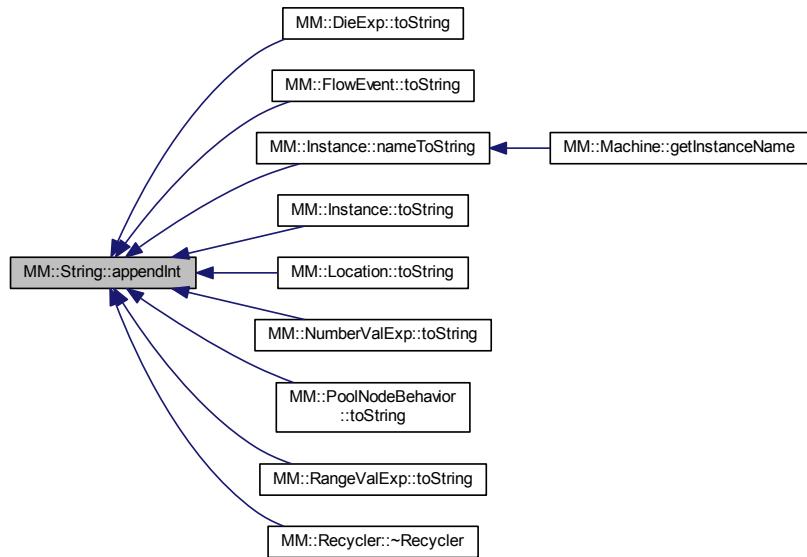


6.107.2.2 MM::VOID String::append (`MM::CHAR * buf, MM::UINT32 len`)

6.107.2.3 MM::VOID String::append (`MM::CHAR c`)

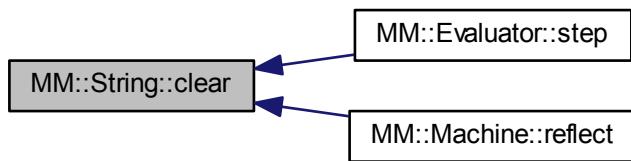
6.107.2.4 MM::VOID String::appendInt (MM::INT32 val)

Here is the caller graph for this function:



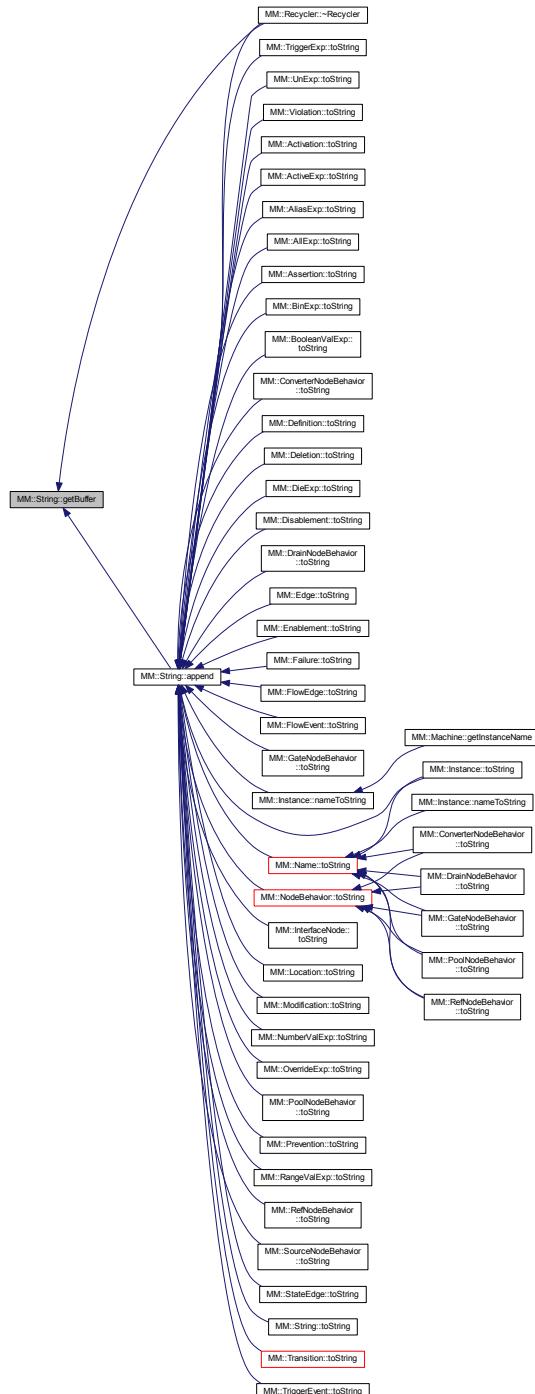
6.107.2.5 MM::VOID String::clear ()

Here is the caller graph for this function:



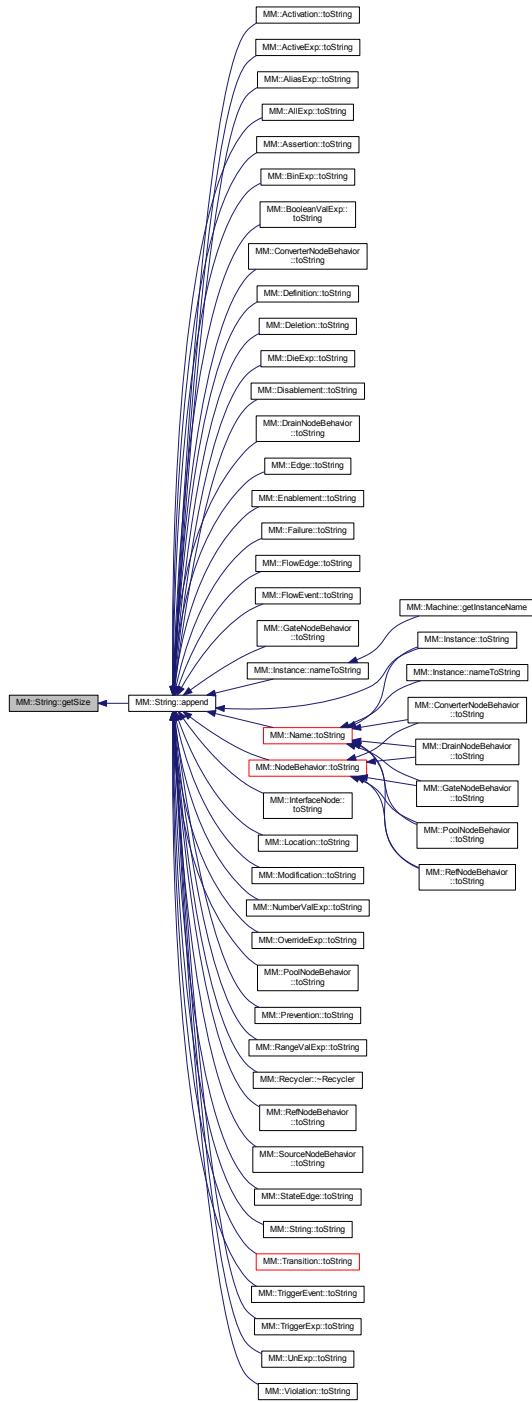
6.107.2.6 MM::CHAR * String::getBuffer()

Here is the caller graph for this function:



6.107.2.7 MM::UINT32 String::getSize ()

Here is the caller graph for this function:



6.107.2.8 MM::TID String::getTypeld() [virtual]

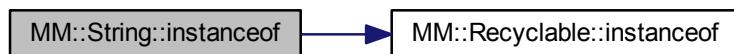
Reimplemented from [MM::Recyclable](#).

6.107.2.9 MM::UINT32 String::getUsed()

6.107.2.10 MM::BOOLEAN String::instanceof(TID *tid*) [virtual]

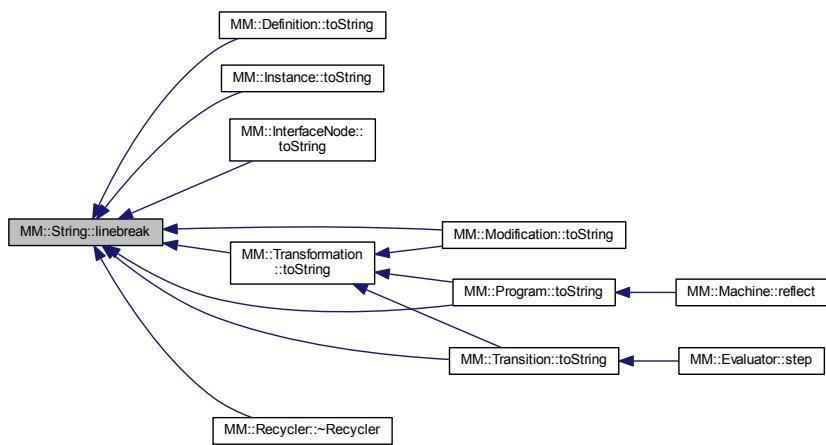
Reimplemented from [MM::Recyclable](#).

Here is the call graph for this function:



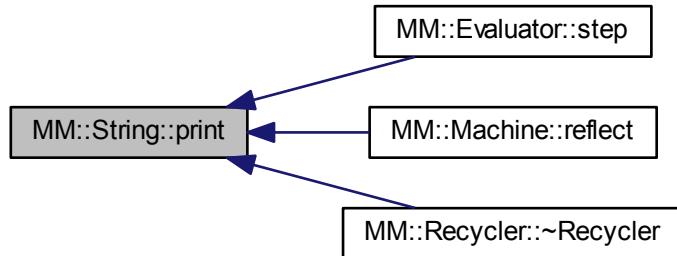
6.107.2.11 MM::VOID String::linebreak()

Here is the caller graph for this function:



6.107.2.12 MM::VOID String::print ()

Here is the caller graph for this function:



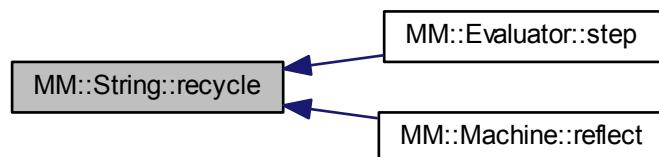
6.107.2.13 MM::VOID String::recycle(MM::Recycler * r) [virtual]

Reimplemented from [MM::Recyclable](#).

Here is the call graph for this function:

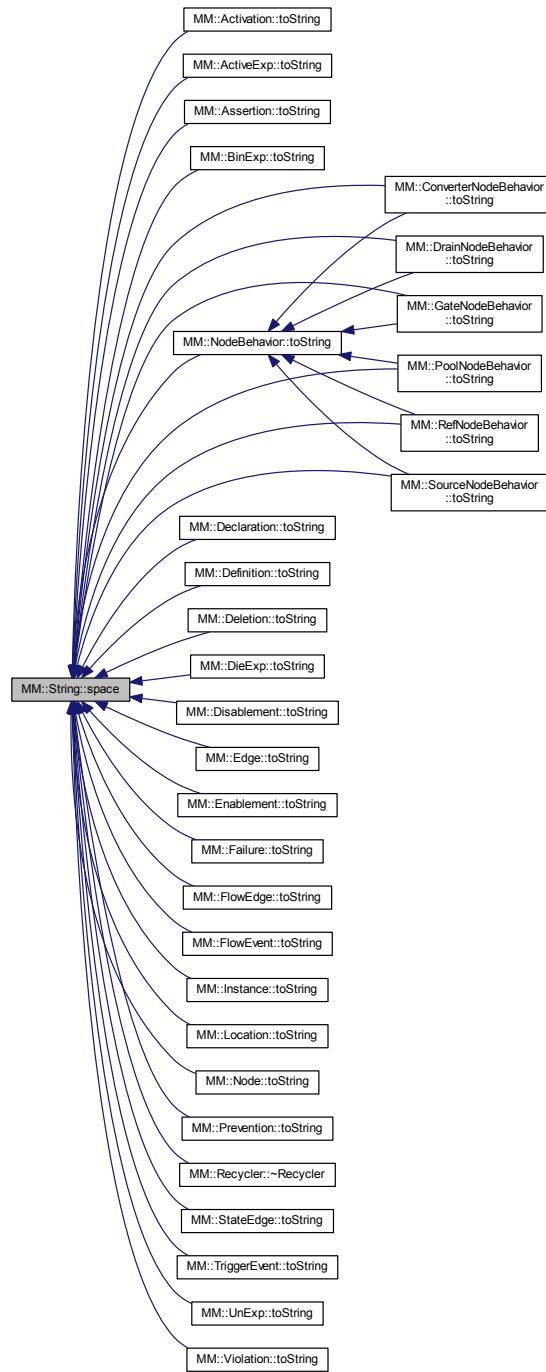


Here is the caller graph for this function:



6.107.2.14 MM::VOID String::space()

Here is the caller graph for this function:

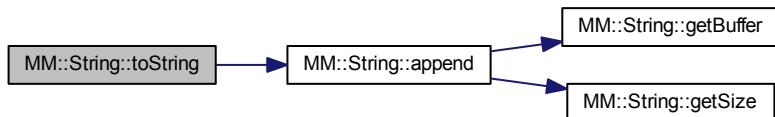


6.107.2.15 MM::VOID String::space(MM::UINT32 amount)

6.107.2.16 MM::VOID String::toString(MM::String * buf) [virtual]

Implements [MM::Recyclable](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[String.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[String.cpp](#)

6.108 Transformation Class Reference

The [Transformation](#) abstraction is the abstract super class of all model transformations.

```
#include <Transformation.h>
```

6.108.1 Detailed Description

The [Transformation](#) abstraction is the abstract super class of all model transformations.

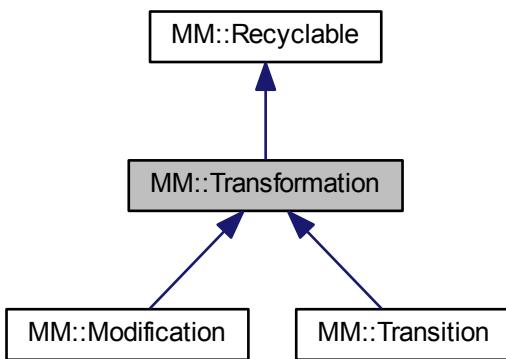
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Transformation.h](#)

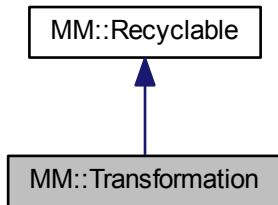
6.109 MM::Transformation Class Reference

```
#include <Transformation.h>
```

Inheritance diagram for MM::Transformation:



Collaboration diagram for MM::Transformation:



Public Member Functions

- `~Transformation ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()=0`
- `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::VOID addElement (MM::Element *element)`
- `MM::Vector< MM::Element * > * getElements ()`
- `MM::VOID clearElements ()`
- `MM::VOID toString (MM::String *buf)`

Protected Member Functions

- `Transformation (MM::Vector< MM::Element * > *elements)`

6.109.1 Constructor & Destructor Documentation

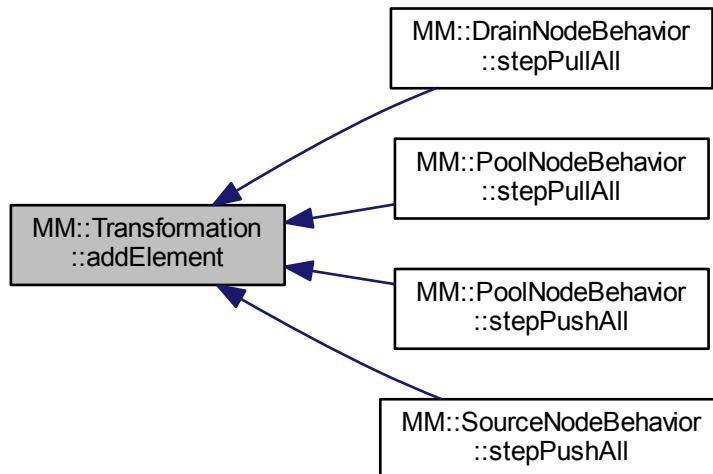
6.109.1.1 `Transformation::Transformation (MM::Vector< MM::Element * > *elements) [protected]`

6.109.1.2 `Transformation::~Transformation ()`

6.109.2 Member Function Documentation

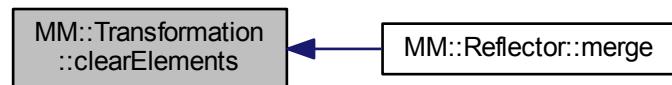
6.109.2.1 MM::VOID Transformation::addElement (MM::Element * element)

Here is the caller graph for this function:



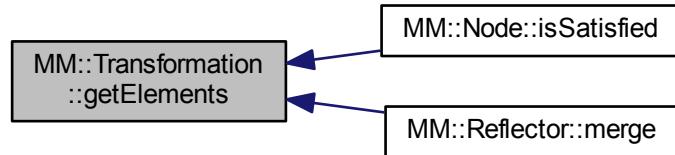
6.109.2.2 MM::VOID Transformation::clearElements ()

Here is the caller graph for this function:



6.109.2.3 MM::Vector< MM::Element * > * Transformation::getElements()

Here is the caller graph for this function:



6.109.2.4 MM::TID MM::Transformation::getTypeId() [pure virtual]

Reimplemented from [MM::Recyclable](#).

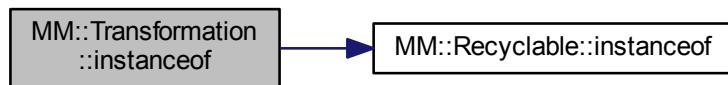
Implemented in [MM::Modification](#), and [MM::Transition](#).

6.109.2.5 MM::BOOLEAN Transformation::instanceof(MM::TID tid) [virtual]

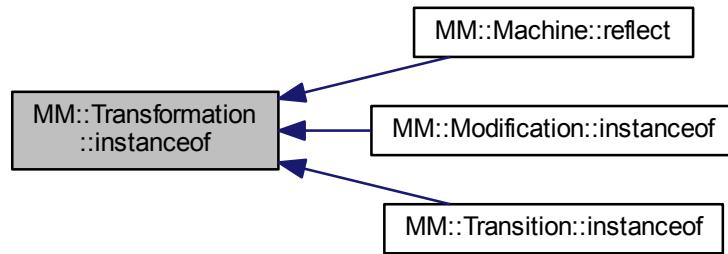
Reimplemented from [MM::Recyclable](#).

Reimplemented in [MM::Transition](#).

Here is the call graph for this function:



Here is the caller graph for this function:

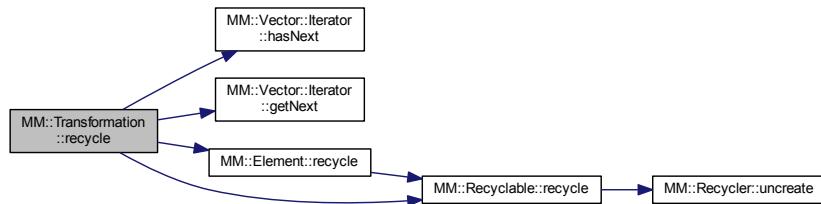


6.109.2.6 MM::VOID Transformation::recycle (MM::Recycler * r) [virtual]

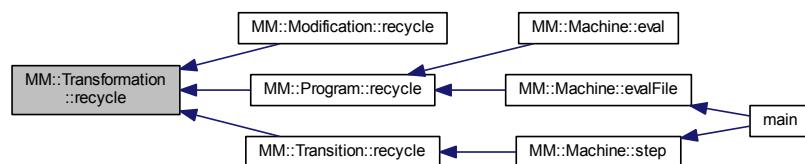
Reimplemented from [MM::Recyclable](#).

Reimplemented in [MM::Transition](#).

Here is the call graph for this function:



Here is the caller graph for this function:

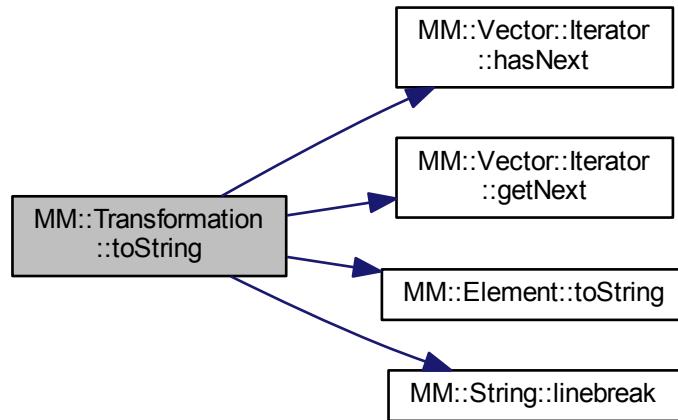


6.109.2.7 MM::VOID Transformation::toString (MM::String * buf) [virtual]

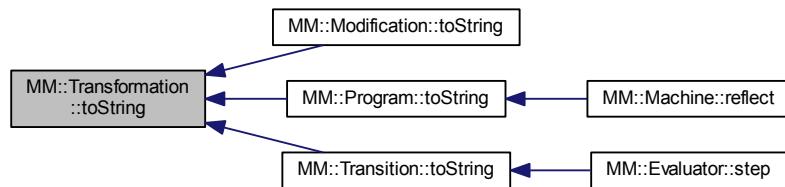
Implements [MM::Recyclable](#).

Reimplemented in [MM::Transition](#).

Here is the call graph for this function:



Here is the caller graph for this function:



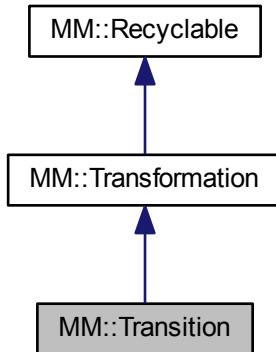
The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Transformation.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Transformation.cpp](#)

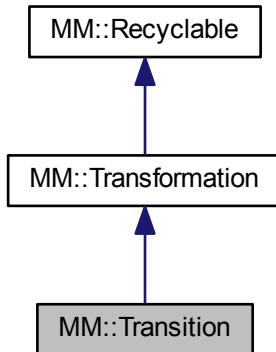
6.110 MM::Transition Class Reference

```
#include <Transition.h>
```

Inheritance diagram for MM::Transition:



Collaboration diagram for MM::Transition:



Public Member Functions

- `Transition (MM::Vector< MM::Element * > *elements)`
- `Transition (MM::Vector< MM::Element * > *elements, MM::Location *loc)`
- `~Transition ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::VOID toString (MM::String *str)`

Additional Inherited Members

6.110.1 Constructor & Destructor Documentation

6.110.1.1 `Transition::Transition (MM::Vector< MM::Element * > * elements)`

we must stringify before instances are deleted

6.110.1.2 `Transition::Transition (MM::Vector< MM::Element * > * elements, MM::Location * loc)`

6.110.1.3 `Transition::~Transition ()`

6.110.2 Member Function Documentation

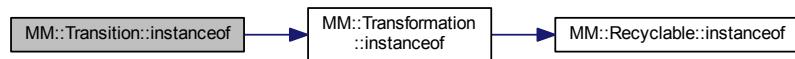
6.110.2.1 `MM::TID Transition::getTypeId () [virtual]`

Implements [MM::Transformation](#).

6.110.2.2 `MM::BOOLEAN Transition::instanceof (MM::TID tid) [virtual]`

Reimplemented from [MM::Transformation](#).

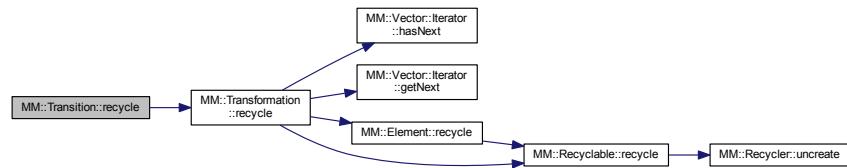
Here is the call graph for this function:



6.110.2.3 `MM::VOID Transition::recycle (MM::Recycler * r) [virtual]`

Reimplemented from [MM::Transformation](#).

Here is the call graph for this function:



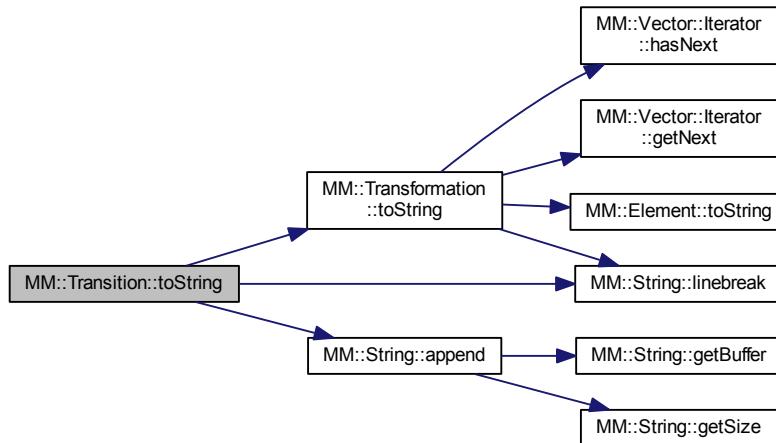
Here is the caller graph for this function:



6.110.2.4 MM::VOID Transition::toString (MM::String * str) [virtual]

Reimplemented from [MM::Transformation](#).

Here is the call graph for this function:



Here is the caller graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Transition.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Transition.cpp](#)

6.111 Transition Class Reference

The [Transition](#) abstraction trace elements that define steps.

```
#include <Transition.h>
```

6.111.1 Detailed Description

The [Transition](#) abstraction trace elements that define steps.

The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Transition.h](#)

6.112 TriggerEvent Class Reference

The [TriggerEvent](#) abstraction defines a trigger fired.

```
#include <Prevention.h>
```

6.112.1 Detailed Description

The [TriggerEvent](#) abstraction defines a trigger fired.

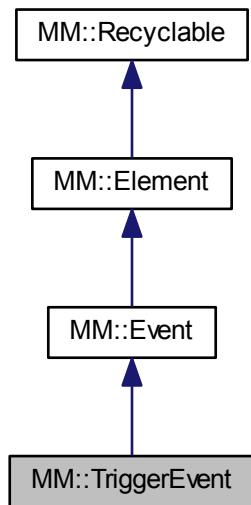
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Prevention.h](#)

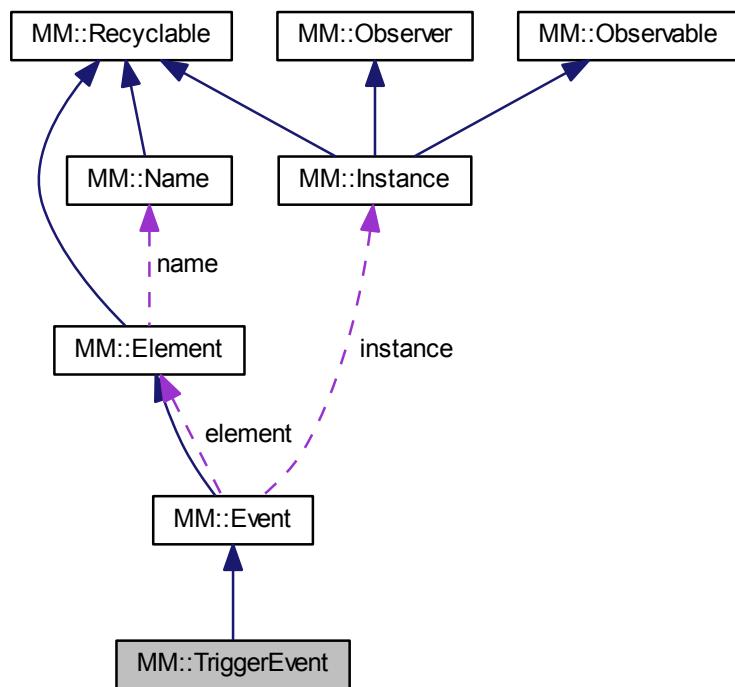
6.113 MM::TriggerEvent Class Reference

```
#include <TriggerEvent.h>
```

Inheritance diagram for MM::TriggerEvent:



Collaboration diagram for MM::TriggerEvent:



Public Member Functions

- [TriggerEvent \(MM::Name *name\)](#)
- [TriggerEvent \(MM::Location *loc, MM::Name *name\)](#)
- [TriggerEvent \(MM::Instance *instance, MM::Edge *edge\)](#)
- [~TriggerEvent \(\)](#)
- [MM::VOID recycle \(MM::Recycler *r\)](#)
- [MM::TID getTypeld \(\)](#)
- [MM::BOOLEAN instanceof \(MM::TID tid\)](#)
- [MM::Location * getLocation \(\)](#)
- [MM::MESSAGE getMessage \(\)](#)
- [MM::VOID toString \(MM::String *buf\)](#)
- [MM::VOID toString \(MM::String *buf, MM::UINT32 indent\)](#)

Additional Inherited Members

6.113.1 Constructor & Destructor Documentation

6.113.1.1 TriggerEvent::TriggerEvent (MM::Name * *name*)

trigger keyword location (parsed)

6.113.1.2 TriggerEvent::TriggerEvent (MM::Location * *loc*, MM::Name * *name*)

6.113.1.3 TriggerEvent::TriggerEvent (MM::Instance * *instance*, MM::Edge * *edge*)

6.113.1.4 TriggerEvent::~TriggerEvent ()

6.113.2 Member Function Documentation

6.113.2.1 MM::Location * TriggerEvent::getLocation ()

6.113.2.2 MM::MESSAGE TriggerEvent::getMessage () [virtual]

Implements [MM::Event](#).

6.113.2.3 MM::TID TriggerEvent::getTypeld () [virtual]

Reimplemented from [MM::Event](#).

6.113.2.4 MM::BOOLEAN TriggerEvent::instanceof (MM::TID *tid*) [virtual]

Reimplemented from [MM::Event](#).

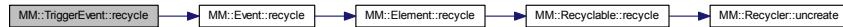
Here is the call graph for this function:



6.113.2.5 MM::VOID TriggerEvent::recycle (MM::Recycler * r) [virtual]

Reimplemented from [MM::Event](#).

Here is the call graph for this function:



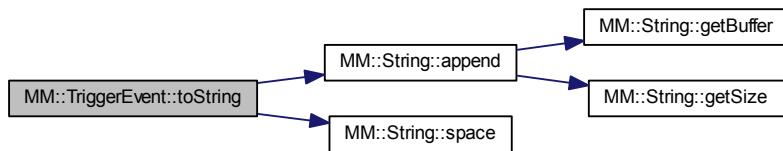
6.113.2.6 MM::VOID TriggerEvent::toString (MM::String * buf) [virtual]

Implements [MM::Event](#).

6.113.2.7 MM::VOID TriggerEvent::toString (MM::String * buf, MM::UINT32 indent) [virtual]

Implements [MM::Event](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[TriggerEvent.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[TriggerEvent.cpp](#)

6.114 TriggerExp Class Reference

The [TriggerExp](#) abstraction expresses that a source of a state edge activates the target of that edge if all of the edge the source operates on are satisfied (meaning a flow of one or more exists for that edge).

```
#include <TriggerExp.h>
```

6.114.1 Detailed Description

The [TriggerExp](#) abstraction expresses that a source of a state edge activates the target of that edge if all of the edge the source operates on are satisfied (meaning a flow of one or more exists for that edge).

Note

Currently not used because no desugaring is applied.
The notion of satisfied does not respect all or any modifiers of nodes.

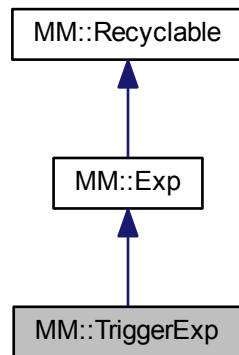
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[TriggerExp.h](#)

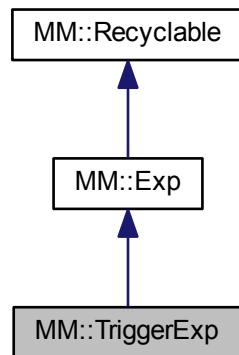
6.115 MM::TriggerExp Class Reference

```
#include <TriggerExp.h>
```

Inheritance diagram for MM::TriggerExp:



Collaboration diagram for MM::TriggerExp:



Public Member Functions

- [TriggerExp \(\)](#)
- [TriggerExp \(MM::Location *loc\)](#)
- [~TriggerExp \(\)](#)
- [MM::VOID recycle \(MM::Recycler *r\)](#)
- [MM::TID getTypeId \(\)](#)
Retrieves the type id of a [Exp](#) object.
- [MM::BOOLEAN instanceof \(MM::TID tid\)](#)
Assesses if an object is an instance of a type tid.
- [MM::VOID toString \(MM::String *buf\)](#)
Serializes a [TriggerExp](#) object into a [String](#) buffer.

Static Public Attributes

- static const [MM::CHAR TRIGGER_CHAR = '*'](#)

Additional Inherited Members

6.115.1 Constructor & Destructor Documentation

6.115.1.1 TriggerExp::TriggerExp ()

trigger string

6.115.1.2 TriggerExp::TriggerExp (MM::Location * loc)

Constructs a [TriggerExp](#) object.

Parameters

<i>loc</i>	source location
------------	-----------------

Returns

new [TriggerExp](#) object

6.115.1.3 TriggerExp::~TriggerExp ()

Destructs a [TriggerExp](#) object.

6.115.2 Member Function Documentation

6.115.2.1 MM::TID TriggerExp::getTypeId () [virtual]

Retrieves the type id of a [Exp](#) object.

Retrieves the type id of a [TriggerExp](#) object.

Returns

type id

Reimplemented from [MM::Exp](#).

6.115.2.2 MM::BOOLEAN TriggerExp::instanceof(MM::TID *tid*) [virtual]

Assesses if an object is an instance of a type tid.

Parameters

<i>tid</i>	type id
------------	---------

Returns

MM_TRUE if this object is instance of tid, MM_FALSE otherwise

Reimplemented from [MM::Exp](#).

Here is the call graph for this function:

**6.115.2.3 MM::VOID TriggerExp::recycle (MM::Recycler * r) [virtual]**

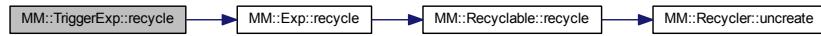
Recycles a [TriggerExp](#) object in a [Recycler](#).

Parameters

<i>r</i>	Recycler object
----------	---------------------------------

Implements [MM::Exp](#).

Here is the call graph for this function:

**6.115.2.4 MM::VOID TriggerExp::toString (MM::String * buf) [virtual]**

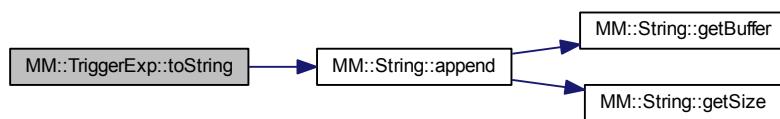
Serializes a [TriggerExp](#) object into a [String](#) buffer.

Parameters

<i>buf</i>	String buffer to serialize this object into
------------	---

Implements [MM::Exp](#).

Here is the call graph for this function:



6.115.3 Member Data Documentation

6.115.3.1 const MM::CHAR TriggerExp::TRIGGER_CHAR = '*' [static]

trigger source location

The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[TriggerExp.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[TriggerExp.cpp](#)

6.116 UnExp Class Reference

The [UnExp](#) abstraction defines unary expressions.

```
#include <UnExp.h>
```

6.116.1 Detailed Description

The [UnExp](#) abstraction defines unary expressions.

The documentation for this class was generated from the following file:

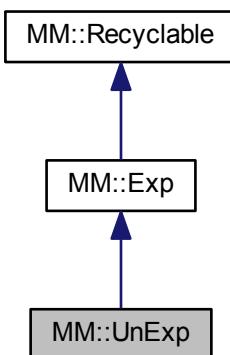
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[UnExp.h](#)

6.117 MM::UnExp Class Reference

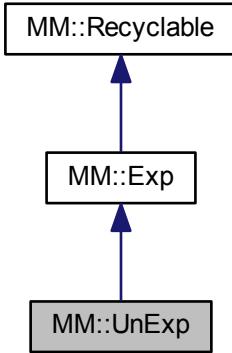
Defines the [UnExp](#) class.

```
#include <UnExp.h>
```

Inheritance diagram for MM::UnExp:



Collaboration diagram for MM::UnExp:



Public Member Functions

- `UnExp (MM::Operator::OP op, MM::Exp *exp)`
- `UnExp (MM::Operator::OP op, MM::Exp *exp, MM::Location *loc)`
- `~UnExp ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`
 - Retrieves the type id of a `Exp` object.*
- `MM::BOOLEAN instanceof (MM::TID tid)`
 - Assesses if an object is an instance of a type tid.*
- `MM::Exp * getExp ()`
- `MM::Operator::OP getOperator ()`
- `MM::VOID toString (MM::String *buf)`

Additional Inherited Members

6.117.1 Detailed Description

Defines the [UnExp](#) class.

6.117.2 Constructor & Destructor Documentation

6.117.2.1 UnExp::UnExp (MM::Operator::OP op, MM::Exp * exp)

operator source location

6.117.2.2 UnExp::UnExp (MM::Operator::OP op, MM::Exp * exp, MM::Location * loc)

6.117.2.3 UnExp::~UnExp ()

6.117.3 Member Function Documentation

6.117.3.1 MM::Exp * UnExp::getExp()

Here is the caller graph for this function:



6.117.3.2 MM::Operator::OP UnExp::getOperator()

Here is the caller graph for this function:



6.117.3.3 MM::TID UnExp::getTypeId() [virtual]

Retrieves the type id of a [Exp](#) object.

Retrieves the type id of a [TriggerExp](#) object.

Returns

type id

Reimplemented from [MM::Exp](#).

6.117.3.4 MM::BOOLEAN UnExp::instanceof(MM::TID *tid*) [virtual]

Assesses if an object is an instance of a type tid.

Parameters

<i>tid</i>	type id
------------	---------

Returns

MM_TRUE if this object is instance of tid, MM_FALSE otherwise

Reimplemented from [MM::Exp](#).

Here is the call graph for this function:



6.117.3.5 MM::VOID UnExp::recycle(MM::Recycler * r) [virtual]

Recycles an [Exp](#) object in a [Recycler](#).

Parameters

<code>r</code>	Recycler object
----------------	---------------------------------

Implements [MM::Exp](#).

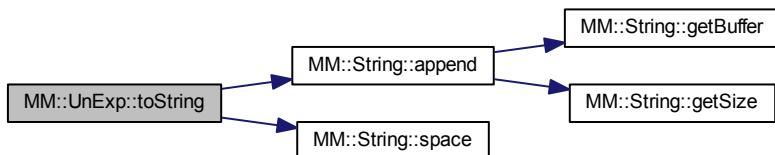
Here is the call graph for this function:



6.117.3.6 MM::VOID UnExp::toString(MM::String * buf) [virtual]

Implements [MM::Exp](#).

Here is the call graph for this function:



The documentation for this class was generated from the following files:

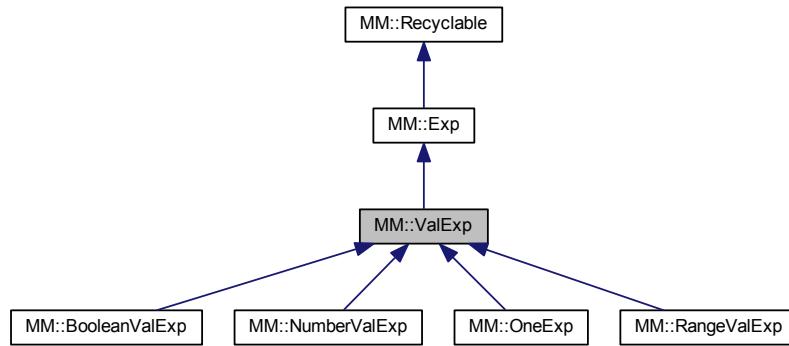
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[UnExp.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[UnExp.cpp](#)

6.118 MM::ValExp Class Reference

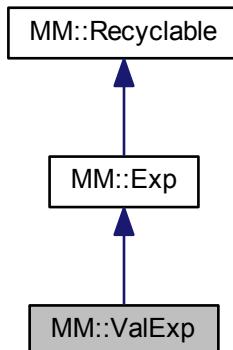
Defines the [ValExp](#) class.

```
#include <ValExp.h>
```

Inheritance diagram for MM::ValExp:



Collaboration diagram for MM::ValExp:



Public Member Functions

- virtual [~ValExp \(\)](#)
- [MM::VOID recycle \(MM::Recycler *r\)](#)
- virtual [MM::TID getTypeld \(\)](#)
Retrieves the type id of a [Exp](#) object.
- virtual [MM::BOOLEAN instanceof \(MM::TID tid\)](#)
Assesses if an object is an instance of a type tid.
- virtual [MM::BOOLEAN greaterEquals \(MM::UINT32 val\)=0](#)
- virtual [MM::VOID toString \(MM::String *buf\)=0](#)

Protected Member Functions

- [ValExp \(\)](#)

6.118.1 Detailed Description

Defines the [ValExp](#) class.

6.118.2 Constructor & Destructor Documentation

6.118.2.1 ValExp::ValExp() [protected]

6.118.2.2 ValExp::~ValExp() [virtual]

6.118.3 Member Function Documentation

6.118.3.1 MM::TID ValExp::getTypeId() [virtual]

Retrieves the type id of a [Exp](#) object.

Retrieves the type id of a [TriggerExp](#) object.

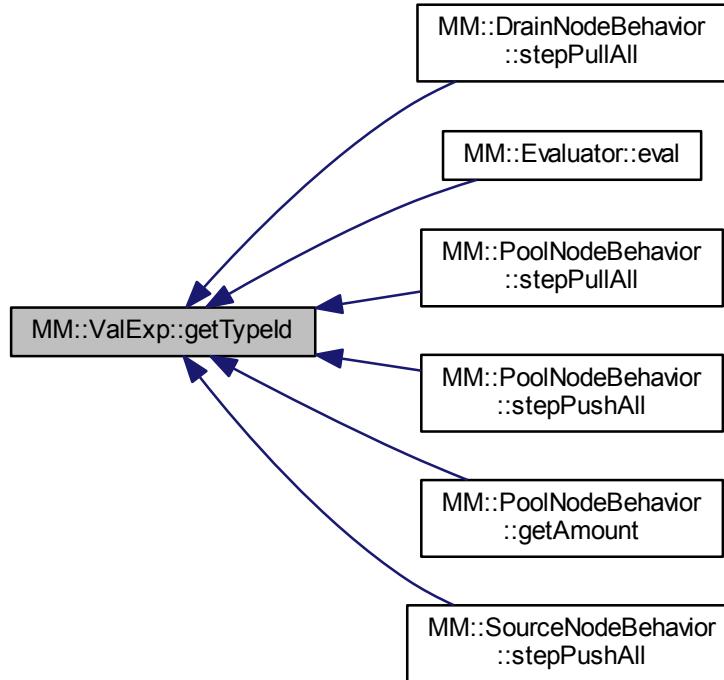
Returns

type id

Reimplemented from [MM::Exp](#).

Reimplemented in [MM::RangeValExp](#), [MM::OneExp](#), [MM::NumberValExp](#), and [MM::BooleanValExp](#).

Here is the caller graph for this function:



6.118.3.2 virtual MM::BOOLEAN MM::ValExp::greaterEquals (MM::UINT32 *val*) [pure virtual]

Implemented in [MM::RangeValExp](#), [MM::NumberValExp](#), [MM::BooleanValExp](#), and [MM::OneExp](#).

6.118.3.3 MM::BOOLEAN ValExp::instanceof (MM::TID *tid*) [virtual]

Assesses if an object is an instance of a type tid.

Parameters

<i>tid</i>	type id
------------	---------

Returns

MM_TRUE if this object is instance of tid, MM_FALSE otherwise

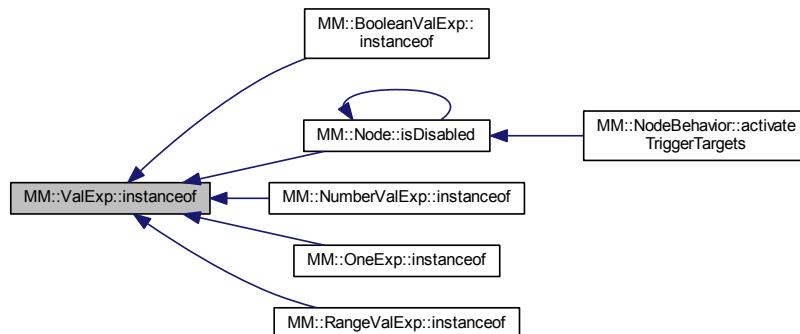
Reimplemented from [MM::Exp](#).

Reimplemented in [MM::RangeValExp](#), [MM::OneExp](#), [MM::NumberValExp](#), and [MM::BooleanValExp](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.118.3.4 MM::VOID ValExp::recycle (MM::Recycler * r) [virtual]

Recycles an [Exp](#) object in a [Recycler](#).

Parameters

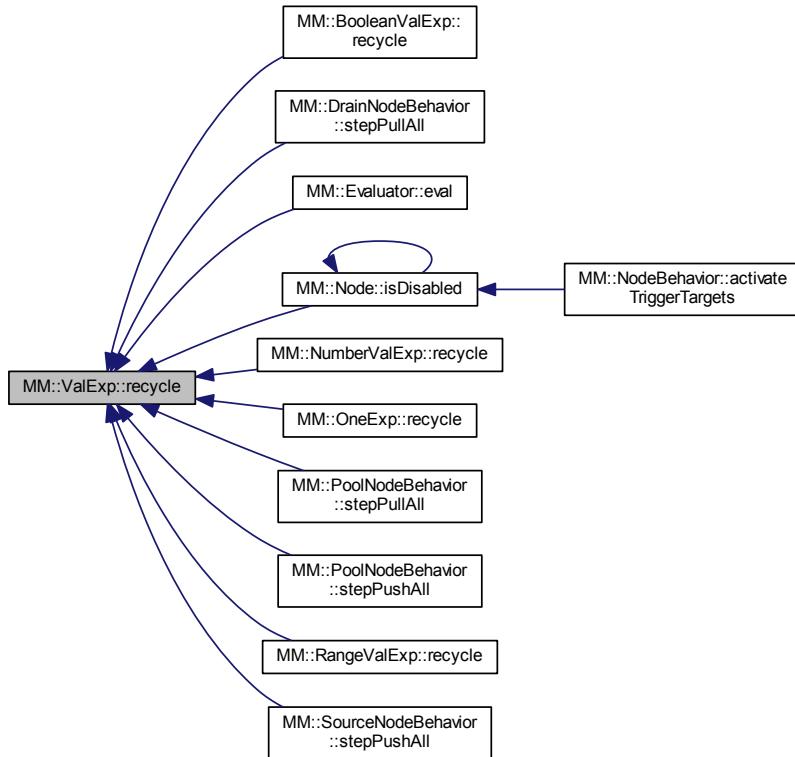
<i>r</i>	Recycler object
----------	---------------------------------

Implements [MM::Exp](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.118.3.5 virtual MM::VOID MM::ValExp::toString(MM::String * buf) [pure virtual]

Implements [MM::Exp](#).

Implemented in [MM::RangeValExp](#), [MM::NumberValExp](#), [MM::BooleanValExp](#), and [MM::OneExp](#).

The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[ValExp.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[ValExp.cpp](#)

6.119 ValExp Class Reference

The [ValExp](#) abstraction defines the abstract super class of value expressions.

```
#include <ValExp.h>
```

6.119.1 Detailed Description

The [ValExp](#) abstraction defines the abstract super class of value expressions.

The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[ValExp.h](#)

6.120 VarExp Class Reference

The [VarExp](#) abstraction defines expressions referencing pools.

```
#include <VarExp.h>
```

6.120.1 Detailed Description

The [VarExp](#) abstraction defines expressions referencing pools.

The documentation for this class was generated from the following file:

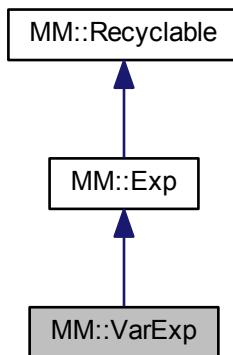
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[VarExp.h](#)

6.121 MM::VarExp Class Reference

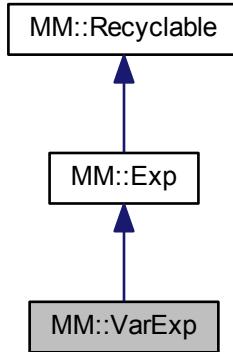
Defines the [VarExp](#) class.

```
#include <VarExp.h>
```

Inheritance diagram for MM::VarExp:



Collaboration diagram for MM::VarExp:



Public Member Functions

- [VarExp \(MM::Name *name\)](#)
- [~VarExp \(\)](#)
- [MM::VOID recycle \(MM::Recycler *r\)](#)
- [MM::TID getTypeId \(\)](#)
 Retrieves the type id of a VarExp object.
- [MM::BOOLEAN instanceof \(MM::TID tid\)](#)
 Assesses if an object is an instance of a type tid.
- [MM::Name * getName \(\)](#)
- [MM::VOID toString \(MM::String *buf\)](#)
 Serializes an VarExp object into a String buffer.

Additional Inherited Members

6.121.1 Detailed Description

Defines the [VarExp](#) class.

6.121.2 Constructor & Destructor Documentation

6.121.2.1 VarExp::VarExp (MM::Name * name)

variable name

Constructs an [VarExp](#) object.

Parameters

<i>name</i>	Name of the pool this expression refers to.
-------------	---

Returns

new [VarExp](#) object

6.121.2.2 VarExp::~VarExp()

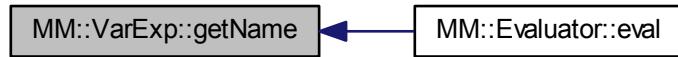
Deconstructs a [VarExp](#) object.

Note

Only called by Recycler.

6.121.3 Member Function Documentation**6.121.3.1 MM::Name * VarExp::getName()**

Here is the caller graph for this function:

**6.121.3.2 MM::TID VarExp::getTypeId() [virtual]**

Retrieves the type id of a [VarExp](#) object.

Returns

type id

Reimplemented from [MM::Exp](#).

6.121.3.3 MM::BOOLEAN VarExp::instanceof(MM::TID tid) [virtual]

Assesses if an object is an instance of a type tid.

Parameters

<i>tid</i>	type id
------------	---------

Returns

MM_TRUE if this object is instance of tid, MM_FALSE otherwise

Reimplemented from [MM::Exp](#).

Here is the call graph for this function:

**6.121.3.4 MM::VOID VarExp::recycle(MM::Recycler * r) [virtual]**

Recycles an [VarExp](#) object in a [Recycler](#).

Parameters

<i>r</i>	Recycler
----------	--------------------------

Implements [MM::Exp](#).

Here is the call graph for this function:

**6.121.3.5 MM::VOID VarExp::toString(MM::String * buf) [virtual]**

Serializes an [VarExp](#) object into a [String](#) buffer.

Parameters

<i>buf</i>	String buffer to serialize this object into
------------	---

Implements [MM::Exp](#).

The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[VarExp.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[VarExp.cpp](#)

6.122 MM::Vector< T > Class Template Reference

```
#include <Vector.h>
```

Classes

- class [Iterator](#)

Public Member Functions

- `Vector ()`
- `~Vector ()`
- virtual `MM::TID getTypeId ()`
- virtual `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::VOID add (T element)`
- `MM::VOID addAll (MM::Vector< T > *other)`
- `T pop ()`
- `MM::BOOLEAN isEmpty ()`
- `T elementAt (MM::UINT32 pos)`
- `MM::INT32 getPosition (T element)`
- `MM::VOID remove (MM::UINT32 pos)`
- `MM::VOID remove (T element)`
- `MM::VOID clear ()`
- `T at (MM::UINT32 pos)`
- `MM::BOOLEAN contains (T element)`
- `MM::UINT32 size ()`
- `MM::Vector< T >::Iterator getIterator ()`
- `MM::Vector< T >::Iterator * getNewIterator ()`

6.122.1 Constructor & Destructor Documentation

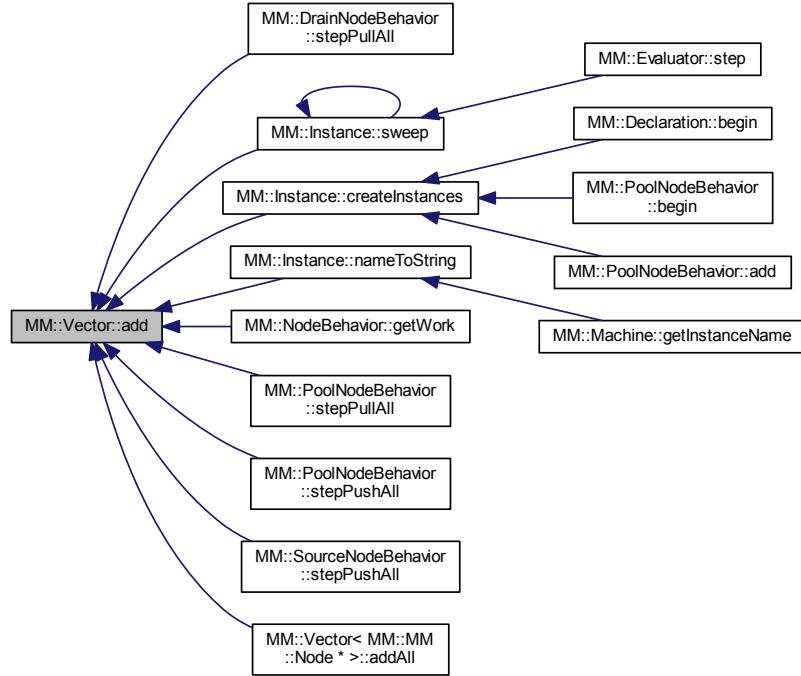
6.122.1.1 `template<class T> MM::Vector< T >::Vector () [inline]`

6.122.1.2 `template<class T> MM::Vector< T >::~Vector () [inline]`

6.122.2 Member Function Documentation

6.122.2.1 template<class T> MM::VOID MM::Vector< T >::add (T element) [inline]

Here is the caller graph for this function:



6.122.2.2 template<class T> MM::VOID MM::Vector< T >::addAll (MM::Vector< T > * other) [inline]

6.122.2.3 template<class T> T MM::Vector< T >::at (MM::UINT32 pos) [inline]

6.122.2.4 template<class T> MM::VOID MM::Vector< T >::clear () [inline]

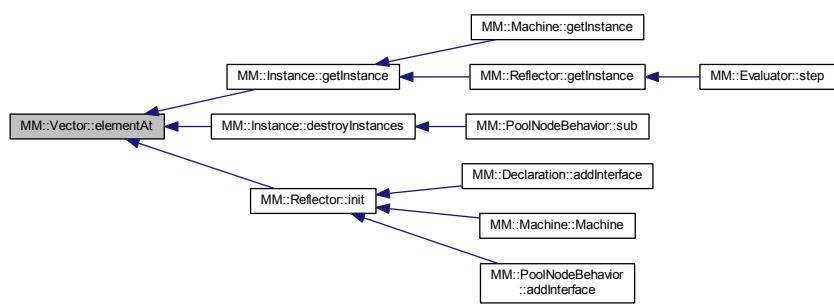
6.122.2.5 template<class T> MM::BOOLEAN MM::Vector< T >::contains (T element) [inline]

Here is the caller graph for this function:



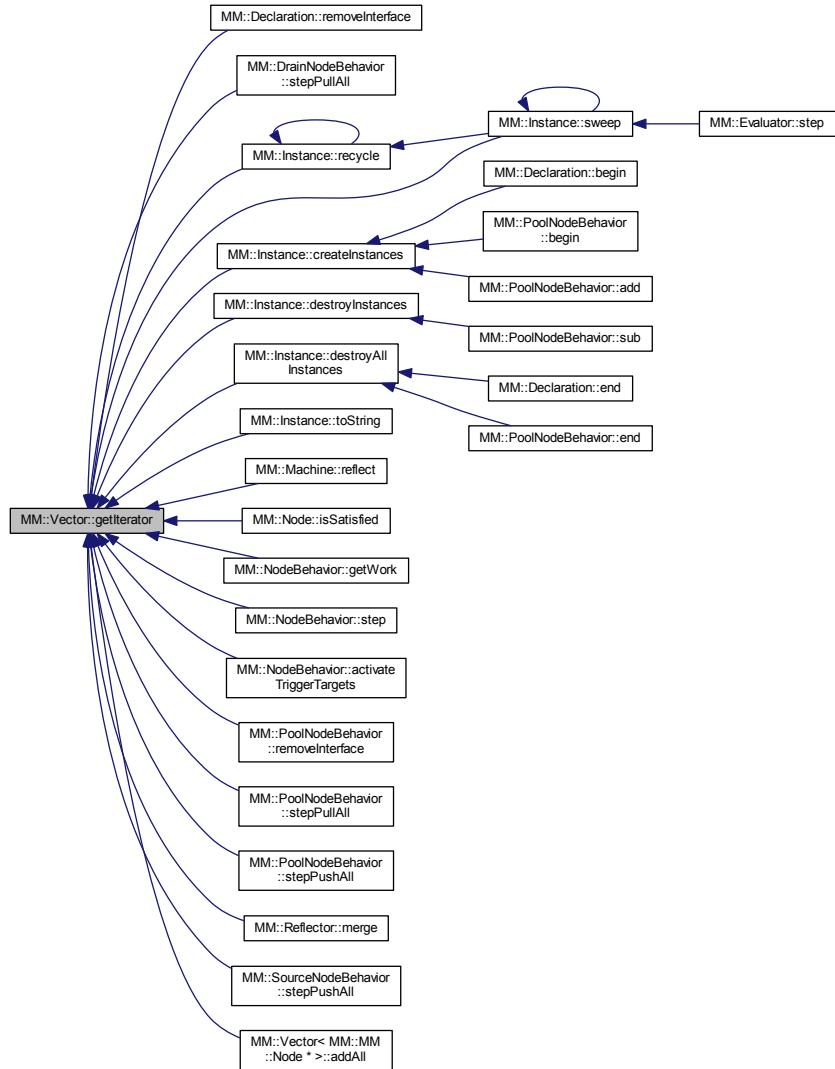
6.122.2.6 template<class T> T MM::Vector< T >::elementAt(MM::UINT32 pos) [inline]

Here is the caller graph for this function:



6.122.2.7 template<class T> MM::Vector<T>::Iterator MM::Vector< T >::getIterator() [inline]

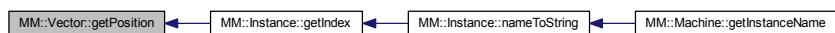
Here is the caller graph for this function:



6.122.2.8 template<class T> MM::Vector<T>::Iterator* MM::Vector< T >::getNewIterator() [inline]

6.122.2.9 template<class T> MM::INT32 MM::Vector< T >::getPosition(T element) [inline]

Here is the caller graph for this function:

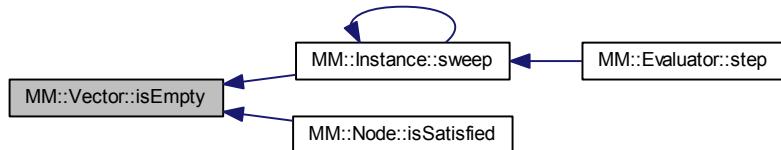


6.122.2.10 template<class T> virtual MM::TID MM::Vector< T >::getTypeId() [inline], [virtual]

6.122.2.11 template<class T> virtual MM::BOOLEAN MM::Vector< T >::instanceof(MM::TID *tid*) [inline], [virtual]

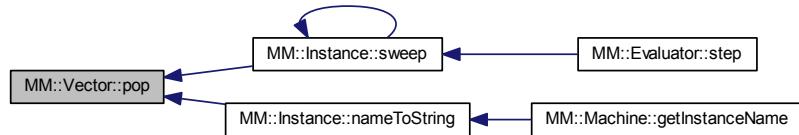
6.122.2.12 template<class T> MM::BOOLEAN MM::Vector< T >::isEmpty() [inline]

Here is the caller graph for this function:



6.122.2.13 template<class T> T MM::Vector< T >::pop() [inline]

Here is the caller graph for this function:



6.122.2.14 template<class T> MM::VOID MM::Vector< T >::remove(MM::UINT32 *pos*) [inline]

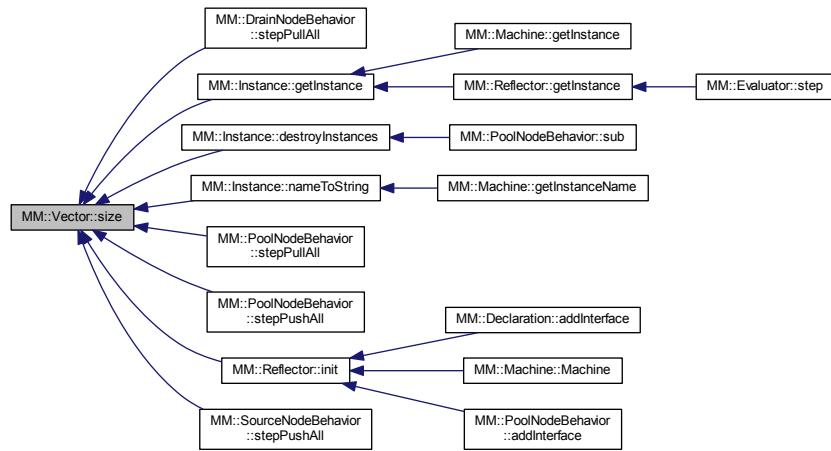
Here is the caller graph for this function:



6.122.2.15 template<class T> MM::VOID MM::Vector< T >::remove(T *element*) [inline]

6.122.2.16 template<class T> MM::UINT32 MM::Vector< T >::size() [inline]

Here is the caller graph for this function:



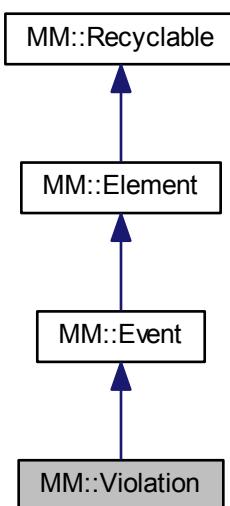
The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Vector.h](#)

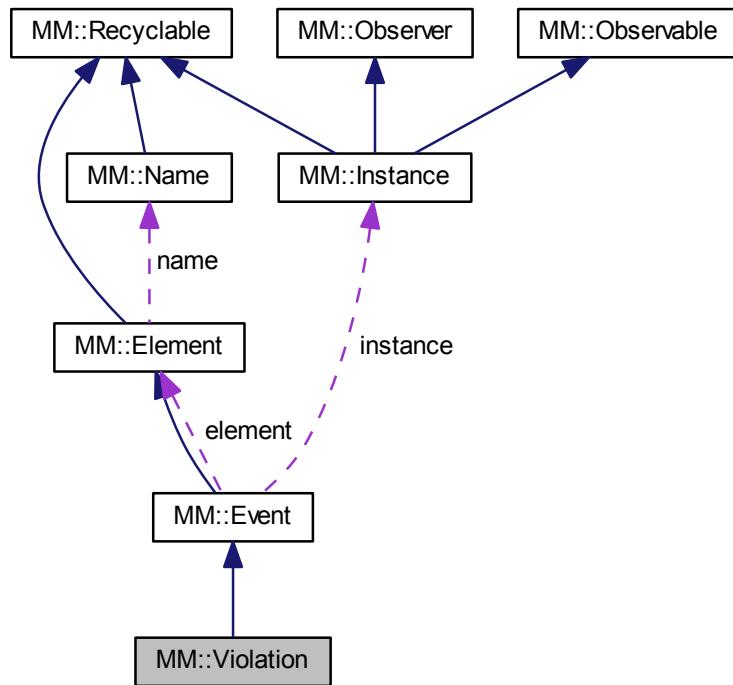
6.123 MM::Violation Class Reference

```
#include <Violation.h>
```

Inheritance diagram for MM::Violation:



Collaboration diagram for MM::Violation:



Public Member Functions

- `Violation (MM::Location *loc, MM::Name *name)`
- `Violation (MM::Name *name)`
- `Violation (MM::Instance *instance, MM::Assertion *element)`
- `~Violation ()`
- `MM::VOID recycle (MM::Recycler *r)`
- `MM::TID getTypeld ()`
- `MM::BOOLEAN instanceof (MM::TID tid)`
- `MM::Location * getLocation ()`
- `MM::MESSAGE getMessage ()`
- `MM::VOID toString (MM::String *buf)`
- `MM::VOID toString (MM::String *buf, MM::UINT32 indent)`

Static Public Attributes

- static const `MM::CHAR * VIOLATE_STR = "violate"`
- static const `MM::UINT32 VIOLATE_LEN = strlen(MM::Violation::VIOLATE_STR)`

Additional Inherited Members

6.123.1 Constructor & Destructor Documentation

6.123.1.1 `Violation::Violation (MM::Location * loc, MM::Name * name)`

6.123.1.2 `Violation::Violation (MM::Name * name)`

6.123.1.3 `Violation::Violation (MM::Instance * instance, MM::Assertion * element)`

6.123.1.4 `Violation::~Violation ()`

6.123.2 Member Function Documentation

6.123.2.1 `MM::Location * Violation::getLocation ()`

6.123.2.2 `MM::MESSAGE Violation::getMessage () [virtual]`

Implements [MM::Event](#).

6.123.2.3 `MM::TID Violation::getTypeID () [virtual]`

Reimplemented from [MM::Event](#).

6.123.2.4 `MM::BOOLEAN Violation::instanceof (MM::TID tid) [virtual]`

Reimplemented from [MM::Event](#).

Here is the call graph for this function:



6.123.2.5 `MM::VOID Violation::recycle (MM::Recycler * r) [virtual]`

Reimplemented from [MM::Event](#).

Here is the call graph for this function:



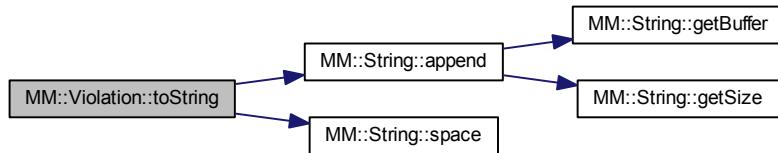
6.123.2.6 `MM::VOID Violation::toString (MM::String * buf) [virtual]`

Implements [MM::Event](#).

6.123.2.7 `MM::VOID Violation::toString (MM::String * buf, MM::UINT32 indent) [virtual]`

Implements [MM::Event](#).

Here is the call graph for this function:



6.123.3 Member Data Documentation

6.123.3.1 `const MM::UINT32 Violation::VIOLATE_LEN = strlen(MM::Violation::VIOLATE_STR) [static]`

6.123.3.2 `const MM::CHAR * Violation::VIOLATE_STR = "violate" [static]`

The documentation for this class was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Violation.h](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Violation.cpp](#)

6.124 Violation Class Reference

The `Violation` abstraction is a transition event that expresses an `Assertion` has been violated.

```
#include <Violation.h>
```

6.124.1 Detailed Description

The `Violation` abstraction is a transition event that expresses an `Assertion` has been violated.

The documentation for this class was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[Violation.h](#)

6.125 yy_buffer_state Struct Reference

Public Attributes

- `FILE * yy_input_file`
- `char * yy_ch_buf`
- `char * yy_buf_pos`
- `yy_size_t yy_buf_size`
- `int yy_n_chars`
- `int yy_is_our_buffer`
- `int yy_is_interactive`
- `int yy_at_bol`
- `int yy_bs_lineno`
- `int yy_bs_column`
- `int yy_fill_buffer`
- `int yy_buffer_status`

6.125.1 Member Data Documentation

6.125.1.1 int yy_buffer_state::yy_at_bol

6.125.1.2 int yy_buffer_state::yy_bs_column

The column count.

6.125.1.3 int yy_buffer_state::yy_bs_lineno

The line count.

6.125.1.4 char* yy_buffer_state::yy_buf_pos

6.125.1.5 yy_size_t yy_buffer_state::yy_buf_size

6.125.1.6 int yy_buffer_state::yy_buffer_status

6.125.1.7 char* yy_buffer_state::yy_ch_buf

6.125.1.8 int yy_buffer_state::yy_fill_buffer

6.125.1.9 FILE* yy_buffer_state::yy_input_file

6.125.1.10 int yy_buffer_state::yy_is_interactive

6.125.1.11 int yy_buffer_state::yy_is_our_buffer

6.125.1.12 int yy_buffer_state::yy_n_chars

The documentation for this struct was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[lex.mm.cpp](#)

6.126 yy_trans_info Struct Reference

Public Attributes

- [flex_int32_t yy_verify](#)
- [flex_int32_t yy_nxt](#)

6.126.1 Member Data Documentation

6.126.1.1 [flex_int32_t yy_trans_info::yy_nxt](#)

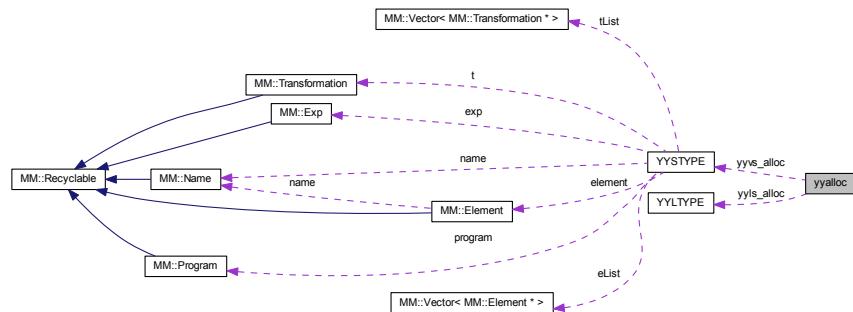
6.126.1.2 [flex_int32_t yy_trans_info::yy_verify](#)

The documentation for this struct was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[lex.mm.cpp](#)

6.127 yyalloc Union Reference

Collaboration diagram for yyalloc:



Public Attributes

- `yytype_int16 yyss_alloc`
- `YYSTYPE yyvs_alloc`
- `YYLTYPE yyls_alloc`

6.127.1 Member Data Documentation

6.127.1.1 YYLTYPE yyalloc::yyls_alloc

6.127.1.2 yytype_int16 yyalloc::yyss_alloc

6.127.1.3 YYSTYPE yyalloc::yyvs_alloc

The documentation for this union was generated from the following file:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[mm.tab.cpp](#)

6.128 YYLTYPE Struct Reference

```
#include <mm.tab.hpp>
```

Public Attributes

- `int first_line`
- `int first_column`
- `int last_line`
- `int last_column`

6.128.1 Member Data Documentation

6.128.1.1 int YYLTYPE::first_column

6.128.1.2 int YYLTYPE::first_line

6.128.1.3 int YYLTYPE::last_column

6.128.1.4 int YYLTYPE::last_line

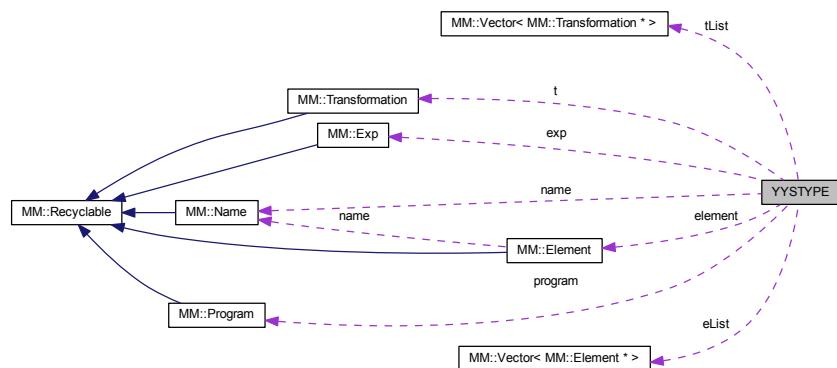
The documentation for this struct was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/mm.tab.hpp
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/YYLTYPE.h

6.129 YYSTYPE Union Reference

```
#include <mm.tab.hpp>
```

Collaboration diagram for YYSTYPE:



Public Attributes

- MM::UINT32 val
- MM::CHAR * str
- MM::Name * name
- MM::Program * program
- MM::Transformation * t
- MM::Vector<MM::Transformation * > * tList
- MM::Element * element
- MM::Vector<MM::Element*> * eList
- MM::Exp * exp
- MM::NodeBehavior::IO io
- MM::NodeBehavior::When when
- MM::NodeBehavior::Act act
- MM::NodeBehavior::How how

6.129.1 Member Data Documentation

6.129.1.1 MM::NodeBehavior::Act YYSTYPE::act

- 6.129.1.2 **MM::Element * YYSTYPE::element**
- 6.129.1.3 **MM::Vector< MM::Element * > * YYSTYPE::eList**
- 6.129.1.4 **MM::Exp * YYSTYPE::exp**
- 6.129.1.5 **MM::NodeBehavior::How YYSTYPE::how**
- 6.129.1.6 **MM::NodeBehavior::IO YYSTYPE::io**
- 6.129.1.7 **MM::Name * YYSTYPE::name**
- 6.129.1.8 **MM::Program * YYSTYPE::program**
- 6.129.1.9 **MM::CHAR * YYSTYPE::str**
- 6.129.1.10 **MM::Transformation * YYSTYPE::t**
- 6.129.1.11 **MM::Vector< MM::Transformation * > * YYSTYPE::tList**
- 6.129.1.12 **MM::UINT32 YYSTYPE::val**
- 6.129.1.13 **MM::NodeBehavior::When YYSTYPE::when**

The documentation for this union was generated from the following files:

- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[mm.tab.cpp](#)
- C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/[mm.tab.hpp](#)

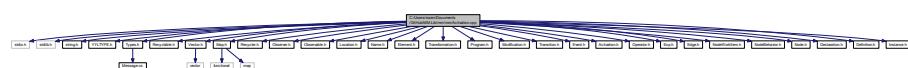
Chapter 7

File Documentation

7.1 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Activation.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Transformation.h"
#include "Program.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "Activation.h"
#include "Operator.h"
#include "Exp.h"
#include "Edge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "Declaration.h"
#include "Definition.h"
#include "Instance.h"
```

Include dependency graph for Activation.cpp:



7.2 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Activation.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Activation](#)

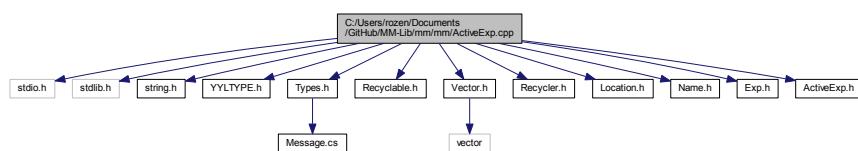
Namespaces

- [MM](#)

7.3 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/ActiveExp.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "Name.h"
#include "Exp.h"
#include "ActiveExp.h"
```

Include dependency graph for ActiveExp.cpp:



Namespaces

- [MM](#)

7.4 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/ActiveExp.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::ActiveExp](#)

Namespaces

- [MM](#)

7.4.1 Detailed Description

Author

Riemer van Rozen

Date

November 22nd 2013

Author

Riemer van Rozen

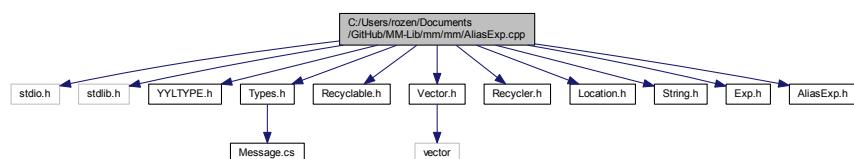
Date

July 21th 2013

7.5 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/AliasExp.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "String.h"
#include "Exp.h"
#include "AliasExp.h"
```

Include dependency graph for AliasExp.cpp:



Namespaces

- [MM](#)

7.5.1 Detailed Description

Author

Riemer van Rozen

Date

July 20th 2013

7.6 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/AliasExp.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class MM::AliasExp

Namespaces

- MM

7.6.1 Detailed Description

Author

Riemer van Rozen

Date

July 20th 2013

Author

Riemer van Rozen

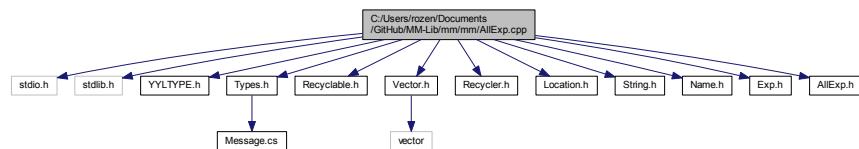
Date _____

July 20th 2013

7.7 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/AllExp.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "String.h"
#include "Name.h"
#include "Exp.h"
#include "AllExp.h"
```

Include dependency graph for AllExp.cpp:



Namespaces

- MM

7.7.1 Detailed Description

Author

Riemer van Rozen

Date

July 19th 2013

7.8 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/AllExp.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class MM::AllExp

Namespaces

- MM

7.8.1 Detailed Description

Author

Riemer van Rozen

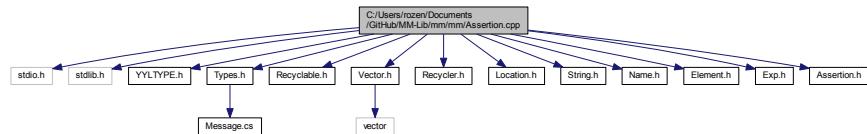
Date

July 19th 2013

7.9 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Assertion.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "String.h"
#include "Name.h"
#include "Element.h"
#include "Exp.h"
#include "Assertion.h"
```

Include dependency graph for Assertion.cpp:



7.10 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Assertion.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class MM::Assertion

Namespaces

- MM

7.10.1 Detailed Description

Author

Riemer van Rozen

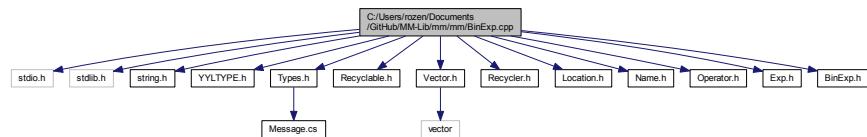
Date

July 29th 2013

7.11 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/BinExp.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "Name.h"
#include "Operator.h"
#include "Exp.h"
#include "BinExp.h"
```

Include dependency graph for BinExp.cpp:



Namespaces

- MM

7.12 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/BinExp.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class MM::BinExp

Namespaces

- MM

7.12.1 Detailed Description

Author

Riemer van Rozen

Date

July 18th 2013

Author

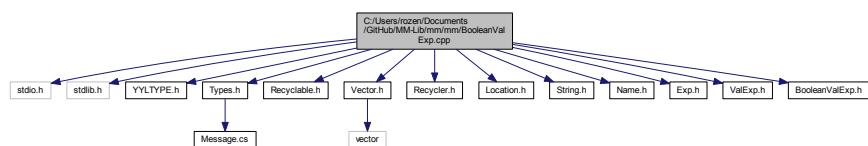
Riemer van Rozen

Date

July 18th 2013

7.13 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/BooleanValExp.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "String.h"
#include "Name.h"
#include "Exp.h"
#include "ValExp.h"
#include "BooleanValExp.h"
Include dependency graph for BooleanValExp.cpp:
```



Namespaces

- MM

7.13.1 Detailed Description

Author

Riemer van Rozen

Date

July 19th 2013

7.14 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/BooleanValExp.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::BooleanValExp](#)

Namespaces

- MM

7.14.1 Detailed Description

Author

Riemer van Rozen

Date

July 19th 2013

7.15 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/ConverterNodeBehavior.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Operator.h"
#include "Exp.h"
#include "Assertion.h"
#include "Deletion.h"
#include "Edge.h"
#include "StateEdge.h"
#include "FlowEdge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "Transformation.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "FlowEvent.h"
#include "TriggerEvent.h"
#include "Failure.h"
#include "Enablement.h"
#include "Disablement.h"
#include "Prevention.h"
#include "Violation.h"
#include "Activation.h"
#include "Program.h"
#include "PoolNodeBehavior.h"
#include "SourceNodeBehavior.h"
#include "DrainNodeBehavior.h"
#include "RefNodeBehavior.h"
#include "GateNodeBehavior.h"
#include "ConverterNodeBehavior.h"
#include "Declaration.h"
#include "InterfaceNode.h"
#include "Definition.h"
#include "Instance.h"
#include "ValExp.h"
#include "UnExp.h"
#include "BinExp.h"
#include "RangeValExp.h"
#include "BooleanValExp.h"
#include "NumberValExp.h"
#include "OverrideExp.h"
#include "ActiveExp.h"
#include "AllExp.h"
```

Generated on Wed Apr 16 2014 13:49 for MM by Doxygen

```
#include "AliasExp.h"
#include "OneExp.h"
#include "VarExp.h"
#include "Reflector.h"
```

7.16 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/ConverterNodeBehavior.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::ConverterNodeBehavior](#)

Namespaces

- [MM](#)

7.16.1 Detailed Description

Author

Riemer van Rozen

Date

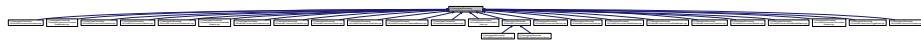
November 21st 2013

7.17 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Declaration.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Transformation.h"
#include "Program.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "FlowEvent.h"
#include "TriggerEvent.h"
#include "Failure.h"
#include "Enablement.h"
#include "Disablement.h"
#include "Violation.h"
#include "Prevention.h"
#include "Operator.h"
#include "Exp.h"
#include "Assertion.h"
#include "Deletion.h"
#include "Activation.h"
#include "Edge.h"
#include "StateEdge.h"
#include "FlowEdge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "PoolNodeBehavior.h"
#include "SourceNodeBehavior.h"
#include "DrainNodeBehavior.h"
#include "GateNodeBehavior.h"
#include "RefNodeBehavior.h"
#include "ConverterNodeBehavior.h"
#include "Declaration.h"
#include "InterfaceNode.h"
#include "Definition.h"
#include "Instance.h"
#include "ValExp.h"
#include "UnExp.h"
#include "BinExp.h"
#include "DieExp.h"
#include "RangeValExp.h"
#include "BooleanValExp.h"
#include "NumberValExp.h"
#include "OverrideExp.h"
#include "ActiveExp.h"
#include "AllExp.h"
#include "AliasExp.h"
#include "OneExp.h"
#include "VarExp.h"
#include "Reflector.h"
#include "Evaluator.h"
```

7.18 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Declaration.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Declaration](#)

Namespaces

- [MM](#)

7.18.1 Detailed Description

Author

Riemer van Rozen

Date

July 23rd 2013

7.19 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Definition.cpp File Reference

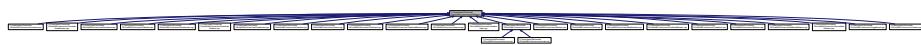
```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "String.h"
#include "Name.h"
#include "Element.h"
#include "Exp.h"
#include "Edge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "PoolNodeBehavior.h"
#include "Node.h"
#include "Declaration.h"
#include "Definition.h"
#include "InterfaceNode.h"
```

Include dependency graph for Definition.cpp:



7.20 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Definition.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Definition](#)

Namespaces

- [MM](#)

7.20.1 Detailed Description

Author

Riemer van Rozen

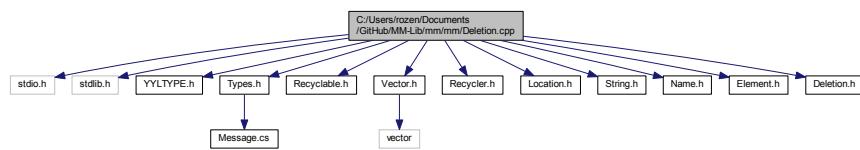
Date

July 21rd 2013

7.21 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Deletion.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "String.h"
#include "Name.h"
#include "Element.h"
#include "Deletion.h"
```

Include dependency graph for Deletion.cpp:



7.22 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Deletion.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Deletion](#)

Namespaces

- [MM](#)

7.22.1 Detailed Description

Author

Riemer van Rozen

Date

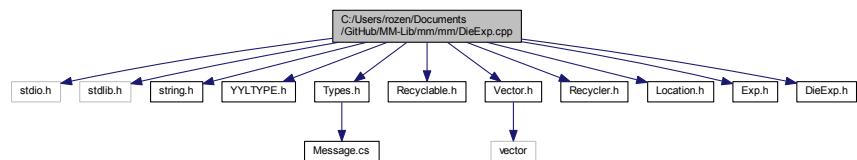
September 29th 2013

7.23 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/DieExp.cpp File Reference

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "Exp.h"
#include "DieExp.h"
  
```

Include dependency graph for DieExp.cpp:



Namespaces

- MM

7.23.1 Detailed Description

Author

Riemer van Rozen

Date

July 20th 2013

7.24 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/DieExp.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::DieExp](#)
Defines the DieExp class.

Namespaces

- MM

7.24.1 Detailed Description

Author

Riemer van Rozen

Date

July 20th 2013

7.25 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Disablement.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Transformation.h"
#include "Program.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "Disablement.h"
#include "Operator.h"
#include "Exp.h"
#include "Edge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "Declaration.h"
#include "Definition.h"
#include "Instance.h"
```

Include dependency graph for Disablement.cpp:



7.26 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Disablement.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Disablement](#)

Namespaces

- [MM](#)

7.26.1 Detailed Description

Author

Riemer van Rozen

Date

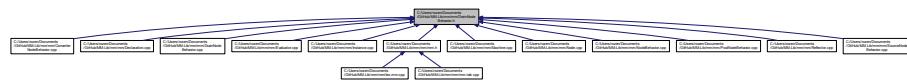
March 26th 2014

7.27 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/DrainNodeBehavior.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Transformation.h"
#include "Program.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "FlowEvent.h"
#include "TriggerEvent.h"
#include "Failure.h"
#include "Enablement.h"
#include "Disablement.h"
#include "Violation.h"
#include "Prevention.h"
#include "Operator.h"
#include "Exp.h"
#include "Assertion.h"
#include "Deletion.h"
#include "Activation.h"
#include "Edge.h"
#include "StateEdge.h"
#include "FlowEdge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "PoolNodeBehavior.h"
#include "SourceNodeBehavior.h"
#include "DrainNodeBehavior.h"
#include "GateNodeBehavior.h"
#include "RefNodeBehavior.h"
#include "ConverterNodeBehavior.h"
#include "Declaration.h"
#include "InterfaceNode.h"
#include "Definition.h"
#include "Instance.h"
#include "ValExp.h"
#include "UnExp.h"
#include "BinExp.h"
#include "DieExp.h"
#include "RangeValExp.h"
#include "BooleanValExp.h"
#include "NumberValExp.h"
#include "OverrideExp.h"
#include "ActiveExp.h"
Generated on Wed Apr 16 2014 13:09:49 for MM by Doxygen
#include "AliasExp.h"
#include "OneExp.h"
#include "VarExp.h"
#include "Reflector.h"
```

7.28 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/DrainNodeBehavior.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::DrainNodeBehavior](#)

Namespaces

- [MM](#)

7.28.1 Detailed Description

Author

Riemer van Rozen

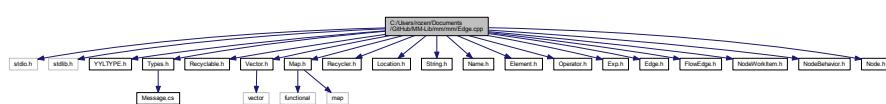
Date

July 11th 2013

7.29 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Edge.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Location.h"
#include "String.h"
#include "Name.h"
#include "Element.h"
#include "Operator.h"
#include "Exp.h"
#include "Edge.h"
#include "FlowEdge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
```

Include dependency graph for Edge.cpp:



7.30 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Edge.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Edge](#)

Namespaces

- [MM](#)

7.30.1 Detailed Description

Author

Riemer van Rozen

Date

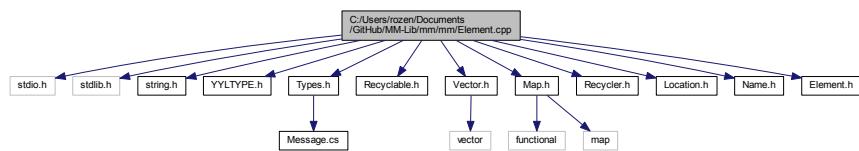
July 10th 2013

7.31 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Element.cpp File Reference

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
  
```

Include dependency graph for Element.cpp:



Namespaces

- MM

7.31.1 Detailed Description

Author

Riemer van Rozen

Date

July 10th 2013

7.32 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Element.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class MM::Element

Namespaces

- MM

7.32.1 Detailed Description

Author

Riemer van Rozen

Date

July 10th 2013

7.33 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Enablement.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Transformation.h"
#include "Program.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "Enablement.h"
#include "Operator.h"
#include "Exp.h"
#include "Assertion.h"
#include "Deletion.h"
#include "Activation.h"
#include "Edge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "Declaration.h"
#include "Definition.h"
#include "Instance.h"
```

Include dependency graph for Enablement.cpp:



7.34 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Enablement.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Enablement](#)

Namespaces

- [MM](#)

7.34.1 Detailed Description

Author

Riemer van Rozen

Date

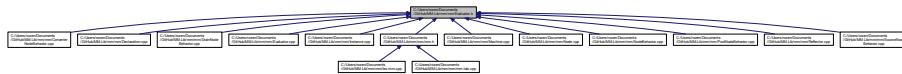
March 27th 2014

7.35 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Evaluator.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Operator.h"
#include "Exp.h"
#include "Assertion.h"
#include "Deletion.h"
#include "Edge.h"
#include "StateEdge.h"
#include "FlowEdge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "Transformation.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "FlowEvent.h"
#include "TriggerEvent.h"
#include "Failure.h"
#include "Activation.h"
#include "Enablement.h"
#include "Disablement.h"
#include "Prevention.h"
#include "Violation.h"
#include "Program.h"
#include "PoolNodeBehavior.h"
#include "SourceNodeBehavior.h"
#include "DrainNodeBehavior.h"
#include "RefNodeBehavior.h"
#include "GateNodeBehavior.h"
#include "ConverterNodeBehavior.h"
#include "Declaration.h"
#include "InterfaceNode.h"
#include "Definition.h"
#include "Instance.h"
#include "ValExp.h"
#include "UnExp.h"
#include "BinExp.h"
#include "RangeValExp.h"
#include "BooleanValExp.h"
#include "NumberValExp.h"
#include "OverrideExp.h"
#include "ActiveExp.h"
#include "AllExp.h"
#include "DieExp.h"
#include "AliasExp.h"
Generated on Wed Apr 16 2014 13:09:49 for MM by Doxygen
#include "OneExp.h"
#include "VarExp.h"
#include "Reflector.h"
#include "Evaluator.h"
```

7.36 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Evaluator.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Evaluator](#)

Namespaces

- [MM](#)

7.36.1 Detailed Description

Author

Riemer van Rozen

Date

September 26th 2013

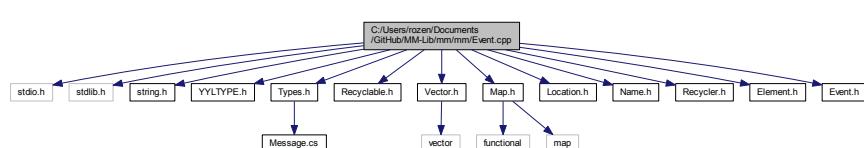
7.37 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Event.cpp File Reference

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Location.h"
#include "Name.h"
#include "Recycler.h"
#include "Element.h"
#include "Event.h"

```

Include dependency graph for Event.cpp:



Namespaces

- [MM](#)

7.37.1 Detailed Description

Author

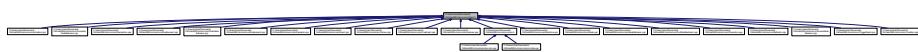
Riemer van Rozen

Date

March 26th 2013

7.38 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Event.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Event](#)

Namespaces

- [MM](#)

7.38.1 Detailed Description

Author

Riemer van Rozen

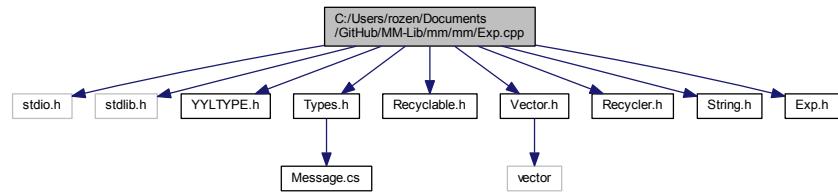
Date

March 26th 2013

7.39 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Exp.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "String.h"
#include "Exp.h"
```

Include dependency graph for Exp.cpp:



Namespaces

- MM

7.39.1 Detailed Description

Author

Riemer van Rozen

Date

July 11th 2013

7.40 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Exp.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class MM::Exp

Namespaces

- MM

7.40.1 Detailed Description

Author

Riemer van Rozen

Date

July 11th 2013

7.41 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Failure.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Transformation.h"
#include "Program.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "Failure.h"
#include "Operator.h"
#include "Exp.h"
#include "Edge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "Declaration.h"
#include "Definition.h"
#include "Instance.h"
```

Include dependency graph for Failure.cpp:



7.42 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Failure.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Failure](#)

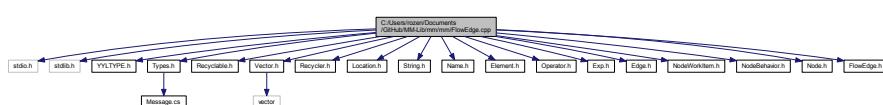
Namespaces

- [MM](#)

7.43 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/FlowEdge.cpp File Reference

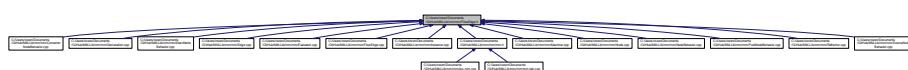
```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "String.h"
#include "Name.h"
#include "Element.h"
#include "Operator.h"
#include "Exp.h"
#include "Edge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "FlowEdge.h"
```

Include dependency graph for FlowEdge.cpp:



7.44 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/FlowEdge.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::FlowEdge](#)

Namespaces

- [MM](#)

7.44.1 Detailed Description

Author

Riemer van Rozen

Date

July 11th 2013

7.45 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/FlowEvent.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Transformation.h"
#include "Program.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "FlowEvent.h"
#include "Operator.h"
#include "Exp.h"
#include "Edge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "Declaration.h"
#include "Definition.h"
#include "Instance.h"
```

Include dependency graph for FlowEvent.cpp:



7.46 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/FlowEvent.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::FlowEvent](#)

Namespaces

- [MM](#)

7.46.1 Detailed Description

Author

Riemer van Rozen

Date

January 30th 2014

7.47 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/GateNodeBehavior.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPEn.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "String.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Exp.h"
#include "Edge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "GateNodeBehavior.h"
#include "Declaration.h"
#include "Definition.h"
#include "Instance.h"
```

Include dependency graph for GateNodeBehavior.cpp:



7.48 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/GateNodeBehavior.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::GateNodeBehavior](#)

Namespaces

- [MM](#)

7.48.1 Detailed Description

Author

Riemer van Rozen

Date

October 16th 2013

7.49 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Instance.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Transformation.h"
#include "Program.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "FlowEvent.h"
#include "TriggerEvent.h"
#include "Failure.h"
#include "Enablement.h"
#include "Disablement.h"
#include "Violation.h"
#include "Prevention.h"
#include "Operator.h"
#include "Exp.h"
#include "Assertion.h"
#include "Deletion.h"
#include "Activation.h"
#include "Edge.h"
#include "StateEdge.h"
#include "FlowEdge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "PoolNodeBehavior.h"
#include "SourceNodeBehavior.h"
#include "DrainNodeBehavior.h"
#include "GateNodeBehavior.h"
#include "RefNodeBehavior.h"
#include "ConverterNodeBehavior.h"
#include "Declaration.h"
#include "InterfaceNode.h"
#include "Definition.h"
#include "Instance.h"
#include "ValExp.h"
#include "UnExp.h"
#include "BinExp.h"
#include "DieExp.h"
#include "RangeValExp.h"
#include "BooleanValExp.h"
#include "NumberValExp.h"
#include "OverrideExp.h"
#include "ActiveExp.h"
#include "AllExp.h"
#include "AliasExp.h"
#include "OneExp.h"
#include "VarExp.h"
#include "Reflector.h"
#include "Evaluator.h"
```

Namespaces

- [MM](#)

7.49.1 Detailed Description

Author

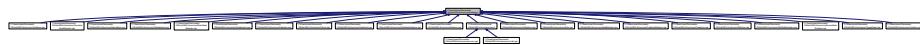
Riemer van Rozen

Date

September 11 2013

7.50 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Instance.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Instance](#)

Namespaces

- [MM](#)

7.50.1 Detailed Description

Author

Riemer van Rozen

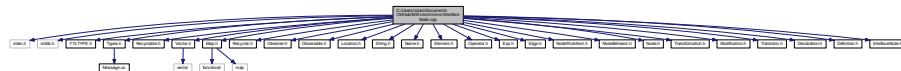
Date

September 11th 2013

7.51 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/InterfaceNode.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "String.h"
#include "Name.h"
#include "Element.h"
#include "Operator.h"
#include "Exp.h"
#include "Edge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "Transformation.h"
#include "Modification.h"
#include "Transition.h"
#include "Declaration.h"
#include "Definition.h"
#include "InterfaceNode.h"
```

Include dependency graph for InterfaceNode.cpp:



7.52 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/InterfaceNode.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::InterfaceNode](#)

Namespaces

- MM

7.52.1 Detailed Description

Author

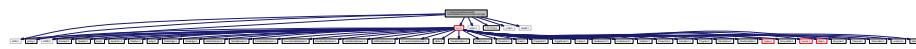
Riemer van Rozen

Date

October 8th 2013

7.53 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/lex.mm.cpp File Reference

```
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "mm.h"
#include "y.tab.h"
#include <unistd.h>
Include dependency graph for lex.mm.cpp:
```



Classes

- struct yy_buffer_state
- struct yy_trans_info

Macros

- #define YY_INT_ALIGNED short int
- #define FLEX_SCANNER
- #define YY_FLEX_MAJOR_VERSION 2
- #define YY_FLEX_MINOR_VERSION 5
- #define YY_FLEX_SUBMINOR_VERSION 35
- #define FLEX_BETA
- #define FLEXINT_H
- #define INT8_MIN (-128)
- #define INT16_MIN (-32767-1)
- #define INT32_MIN (-2147483647-1)
- #define INT8_MAX (127)
- #define INT16_MAX (32767)
- #define INT32_MAX (2147483647)
- #define UINT8_MAX (255U)
- #define UINT16_MAX (65535U)
- #define UINT32_MAX (4294967295U)
- #define yyconst

- #define YY_NULL 0
- #define YY_SC_TO_UI(c) ((unsigned int) (unsigned char) c)
- #define BEGIN (yy_start) = 1 + 2 *
- #define YY_START (((yy_start) - 1) / 2)
- #define YYSTATE YY_START
- #define YY_STATE_EOF(state) (YY_END_OF_BUFFER + state + 1)
- #define YY_NEW_FILE yyrestart(yyin)
- #define YY_END_OF_BUFFER_CHAR 0
- #define YY_BUF_SIZE 16384
- #define YY_STATE_BUF_SIZE ((YY_BUF_SIZE + 2) * sizeof(yy_state_type))
- #define YY_TYPEDEF_YY_BUFFER_STATE
- #define EOB_ACT_CONTINUE_SCAN 0
- #define EOB_ACT_END_OF_FILE 1
- #define EOB_ACT_LAST_MATCH 2
- #define YY_LESS_LINENO(n)
- #define yyless(n)
- #define unput(c) yyunput(c, (yytext_ptr))
- #define YY_TYPEDEF_YY_SIZE_T
- #define YY_STRUCT_YY_BUFFER_STATE
- #define YY_BUFFER_NEW 0
- #define YY_BUFFER_NORMAL 1
- #define YY_BUFFER_EOF_PENDING 2
- #define YY_CURRENT_BUFFER
- #define YY_CURRENT_BUFFER_LVALUE (yy_buffer_stack)[(yy_buffer_stack_top)]
- #define YY_FLUSH_BUFFER yy_flush_buffer(YY_CURRENT_BUFFER)
- #define yy_new_buffer yy_create_buffer
- #define yy_set_interactive(is_interactive)
- #define yy_set_bol(at_bol)
- #define YY_AT_BOL() (YY_CURRENT_BUFFER_LVALUE->yy_at_bol)
- #define yywrap(n) 1
- #define YY_SKIP_YYWRAP
- #define yytext_ptr yytext
- #define YY_DO_BEFORE_ACTION
- #define YY_NUM_RULES 87
- #define YY_END_OF_BUFFER 88
- #define REJECT reject_used_but_not_detected
- #define yymore() yymore_used_but_not_detected
- #define YY_MORE_ADJ 0
- #define YY_RESTORE_YY_MORE_OFFSET
- #define LEX_ID_SIZE 256
- #define YY_USER_ACTION
- #define INITIAL 0
- #define ERROR 1
- #define IDPART 2
- #define IDPART2 3
- #define IN_S_COMMENT 4
- #define IN_M_COMMENT 5
- #define YY_EXTRA_TYPE void *
- #define YY_READ_BUF_SIZE 8192
- #define ECHO do { if (fwrite(yytext, yyleng, 1, yyout)) {} } while (0)
- #define YY_INPUT(buf, result, max_size)
- #define yyterminate() return YY_NULL
- #define YY_START_STACK_INCR 25
- #define YY_FATAL_ERROR(msg) yy_fatal_error(msg)
- #define YY_DECL_IS_OURS 1

- #define YY_DECL int yylex (void)
- #define YY_BREAK break;
- #define YY_RULE_SETUP YY_USER_ACTION
- #define YY_EXIT_FAILURE 2
- #define yyless(n)
- #define YYTABLES_NAME "yytables"

Typedefs

- typedef signed char flex_int8_t
- typedef short int flex_int16_t
- typedef int flex_int32_t
- typedef unsigned char flex_uint8_t
- typedef unsigned short int flex_uint16_t
- typedef unsigned int flex_uint32_t
- typedef struct yy_buffer_state * YY_BUFFER_STATE
- typedef size_t yy_size_t
- typedef unsigned char YY_CHAR
- typedef int yy_state_type

Functions

- void yyrestart (FILE *input_file)
- void yy_switch_to_buffer (YY_BUFFER_STATE new_buffer)
- YY_BUFFER_STATE yy_create_buffer (FILE *file, int size)
- void yy_delete_buffer (YY_BUFFER_STATE b)
- void yy_flush_buffer (YY_BUFFER_STATE b)
- void yppush_buffer_state (YY_BUFFER_STATE new_buffer)
- void yypop_buffer_state (void)
- YY_BUFFER_STATE yy_scan_buffer (char *base, yy_size_t size)
- YY_BUFFER_STATE yy_scan_string (yyconst char *yy_str)
- YY_BUFFER_STATE yy_scan_bytes (yyconst char *bytes, int len)
- void * yyalloc (yy_size_t)
- void * yyrealloc (void *, yy_size_t)
- void yyfree (void *)
- int yylex_destroy (void)
- int yyget_debug (void)
- void yyset_debug (int debug_flag)
- YY_EXTRA_TYPE yyget_extra (void)
- void yyset_extra (YY_EXTRA_TYPE user_defined)
- FILE * yyget_in (void)
- void yyset_in (FILE *in_str)
- FILE * yyget_out (void)
- void yyset_out (FILE *out_str)
- int yyget_leng (void)
- char * yyget_text (void)
- int yyget_lineno (void)
- void yyset_lineno (int line_number)
- int yylex (void)
- if (!(yy_init))
- while (1)
- int isatty (int)

Variables

- int `yylen`
- FILE * `yyin` = (FILE *) 0
- FILE * `yyout` = (FILE *) 0
- int `yylineno` = 1
- char * `yytext`
- int `yy flex_debug` = 0
- int `yycolumn` = 1
- char `yyid` [256] = { 0 }
- char `yystr` [256] = { 0 }
- int `yyidpos` = 0
- bool `inID` = false
- YY_DECL
- register char * `yy_cp`
- register char * `yy_bp`
- register int `yy_act`

7.53.1 Macro Definition Documentation

7.53.1.1 #define BEGIN (yy_start) = 1 + 2 *

7.53.1.2 #define ECHO do { if (fwrite(yytext, yylen, 1, yyout)) {} } while (0)

7.53.1.3 #define EOB_ACT_CONTINUE_SCAN 0

7.53.1.4 #define EOB_ACT_END_OF_FILE 1

7.53.1.5 #define EOB_ACT_LAST_MATCH 2

7.53.1.6 #define ERROR 1

7.53.1.7 #define FLEX_BETA

7.53.1.8 #define FLEX_SCANNER

7.53.1.9 #define FLEXINT_H

7.53.1.10 #define IDPART 2

7.53.1.11 #define IDPART2 3

7.53.1.12 #define IN_M_COMMENT 5

7.53.1.13 #define IN_S_COMMENT 4

7.53.1.14 #define INITIAL 0

7.53.1.15 #define INT16_MAX (32767)

7.53.1.16 #define INT16_MIN (-32767-1)

7.53.1.17 #define INT32_MAX (2147483647)

7.53.1.18 #define INT32_MIN (-2147483647-1)

```

7.53.1.19 #define INT8_MAX (127)
7.53.1.20 #define INT8_MIN (-128)
7.53.1.21 #define LEX_ID_SIZE 256
7.53.1.22 #define REJECT reject_used_but_not_detected
7.53.1.23 #define UINT16_MAX (65535U)
7.53.1.24 #define UINT32_MAX (4294967295U)
7.53.1.25 #define UINT8_MAX (255U)
7.53.1.26 #define unput( c ) yyunput( c, (yytext_ptr) )
7.53.1.27 #define YY_AT_BOL( ) (YY_CURRENT_BUFFER_LVALUE->yy_at_bol)
7.53.1.28 #define YY_BREAK break;
7.53.1.29 #define YY_BUF_SIZE 16384
7.53.1.30 #define YY_BUFFER_EOF_PENDING 2
7.53.1.31 #define YY_BUFFER_NEW 0
7.53.1.32 #define YY_BUFFER_NORMAL 1
7.53.1.33 #define YY_CURRENT_BUFFER

```

Value:

```
( (yy_buffer_stack) \
    ? (yy_buffer_stack)[(yy_buffer_stack_top)] \
    : NULL)
```

```

7.53.1.34 #define YY_CURRENT_BUFFER_LVALUE (yy_buffer_stack)[(yy_buffer_stack_top)]
7.53.1.35 #define YY_DECL int yylex (void)
7.53.1.36 #define YY_DECL_IS_OURS 1
7.53.1.37 #define YY_DO_BEFORE_ACTION

```

Value:

```
(yytext_ptr) = yy_bp; \
    yylen = (size_t) (yy_cp - yy_bp); \
    (yy_hold_char) = *yy_cp; \
    *yy_cp = '\0'; \
    (yy_c_buf_p) = yy_cp;
```

```

7.53.1.38 #define YY_END_OF_BUFFER 88
7.53.1.39 #define YY_END_OF_BUFFER_CHAR 0

```

```

7.53.1.40 #define YY_EXIT_FAILURE 2

7.53.1.41 #define YY_EXTRA_TYPE void *

7.53.1.42 #define YY_FATAL_ERROR( msg ) yy_fatal_error( msg )

7.53.1.43 #define YY_FLEX_MAJOR_VERSION 2

7.53.1.44 #define YY_FLEX_MINOR_VERSION 5

7.53.1.45 #define YY_FLEX_SUBMINOR_VERSION 35

7.53.1.46 #define YY_FLUSH_BUFFER yy_flush_buffer(YY_CURRENT_BUFFER)

7.53.1.47 #define YY_INPUT( buf, result, max_size )

```

Value:

```

if ( YY_CURRENT_BUFFER_LVALUE->yy_is_interactive ) \
{ \
    int c = '*' ; \
    size_t n; \
    for ( n = 0; n < max_size && \
          (c = getc( yyin )) != EOF && c != '\n'; ++n ) \
        buf[n] = (char) c; \
    if ( c == '\n' ) \
        buf[n++] = (char) c; \
    if ( c == EOF && ferror( yyin ) ) \
        YY_FATAL_ERROR( "input in flex scanner failed" ); \
    result = n; \
} \
else \
{ \
    errno=0; \
    while ( (result = fread(buf, 1, max_size, yyin))==0 && ferror(
yyin)) \
    { \
        if( errno != EINTR) \
        { \
            YY_FATAL_ERROR( "input in flex scanner failed" ); \
            break; \
        } \
        errno=0; \
        clearerr(yyin); \
    } \
}

```

```

7.53.1.48 #define YY_INT_ALIGNED short int

7.53.1.49 #define YY_LESS_FILENO( n )

7.53.1.50 #define YY_MORE_ADJ 0

7.53.1.51 #define yy_new_buffer yy_create_buffer

7.53.1.52 #define YY_NEW_FILE yyrestart(yyin)

7.53.1.53 #define YY_NULL 0

7.53.1.54 #define YY_NUM_RULES 87

7.53.1.55 #define YY_READ_BUF_SIZE 8192

7.53.1.56 #define YY_RESTORE_YY_MORE_OFFSET

```

7.53.1.57 #define YY_RULE_SETUP YY_USER_ACTION

7.53.1.58 #define YY_SC_TO_UI(c)((unsigned int)(unsigned char)c)

7.53.1.59 #define yy_set_bol(at_bol)

Value:

```
{ \
    if ( ! YY_CURRENT_BUFFER ){ \
        yyensure_buffer_stack (); \
        YY_CURRENT_BUFFER_LVALUE = \
            yy_create_buffer(yyin,YY_BUF_SIZE ); \
    } \
    YY_CURRENT_BUFFER_LVALUE->yy_at_bol = at_bol; \
}
```

7.53.1.60 #define yy_set_interactive(is_interactive)

Value:

```
{ \
    if ( ! YY_CURRENT_BUFFER ){ \
        yyensure_buffer_stack (); \
        YY_CURRENT_BUFFER_LVALUE = \
            yy_create_buffer(yyin,YY_BUF_SIZE ); \
    } \
    YY_CURRENT_BUFFER_LVALUE->yy_is_interactive = is_interactive; \
}
```

7.53.1.61 #define YY_SKIP_YYWRAP

7.53.1.62 #define YY_START (((yy_start) - 1) / 2)

7.53.1.63 #define YY_START_STACK_INCR 25

7.53.1.64 #define YY_STATE_BUF_SIZE ((YY_BUF_SIZE + 2) * sizeof(yy_state_type))

7.53.1.65 #define YY_STATE_EOF(state)(YY_END_OF_BUFFER + state + 1)

7.53.1.66 #define YY_STRUCT_YY_BUFFER_STATE

7.53.1.67 #define YY_TYPEDEF_YY_BUFFER_STATE

7.53.1.68 #define YY_TYPEDEF_YY_SIZE_T

7.53.1.69 #define YY_USER_ACTION

Value:

```
if(inID == false) \
{ \
    yylloc.first_line = (int) yylineno; \
    yylloc.first_column = (int) yycolumn; \
    yylloc.last_line = (int) yylineno; \
} \
yylloc.last_column = (int) yycolumn + (int) yyleng - 1; \
yycolumn += yyleng;
```

7.53.1.70 #define yyconst

7.53.1.71 #define yyless(*n*)

Value:

```
do \
{ \
/* Undo effects of setting up yytext. */ \
int yyless_macro_arg = (n); \
YY_LESS_LINENO(yyless_macro_arg); \
*yy_cp = (yy_hold_char); \
YY_RESTORE YY_MORE_OFFSET \
(yy_c_buf_p) = yy_cp = yy_bp + yyless_macro_arg - YY_MORE_ADJ; \
} \
YY_DO_BEFORE_ACTION; /* set up yytext again */ \
} \
while ( 0 )
```

7.53.1.72 #define yyless(*n*)

Value:

```
do \
{ \
/* Undo effects of setting up yytext. */ \
int yyless_macro_arg = (n); \
YY_LESS_LINENO(yyless_macro_arg); \
yytext[yylen] = (yy_hold_char); \
(yy_c_buf_p) = yytext + yyless_macro_arg; \
(yy_hold_char) = *(yy_c_buf_p); \
*(yy_c_buf_p) = '\0'; \
yylen = yyless_macro_arg; \
} \
while ( 0 )
```

7.53.1.73 #define yymore() yymore_used_but_not_detected

7.53.1.74 #define YYSTATE YY_START

7.53.1.75 #define YYTABLES_NAME "yytables"

7.53.1.76 #define yyterminate() return YY_NULL

7.53.1.77 #define yytext_ptr yytext

7.53.1.78 #define yywrap(*n*) 1

7.53.2 Typedef Documentation

7.53.2.1 typedef short int flex_int16_t

7.53.2.2 typedef int flex_int32_t

7.53.2.3 typedef signed char flex_int8_t

7.53.2.4 typedef unsigned short int flex_uint16_t

7.53.2.5 typedef unsigned int flex_uint32_t

7.53.2.6 typedef unsigned char flex_uint8_t

7.53.2.7 `typedef struct yy_buffer_state* YY_BUFFER_STATE`

7.53.2.8 `typedef unsigned char YY_CHAR`

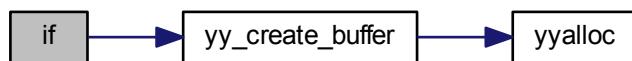
7.53.2.9 `typedef size_t yy_size_t`

7.53.2.10 `typedef int yy_state_type`

7.53.3 Function Documentation

7.53.3.1 `if (! yy_init)`

Here is the call graph for this function:



7.53.3.2 `int isatty (int)`

7.53.3.3 `while (1)`

[^.]

7.53.3.4 `YY_BUFFER_STATE yy_create_buffer (FILE * file, int size)`

Allocate and initialize an input buffer state.

Parameters

<code>file</code>	A readable stream.
<code>size</code>	The character buffer size in bytes. When in doubt, use YY_BUF_SIZE.

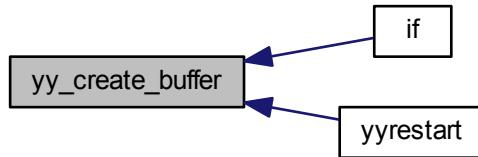
Returns

the allocated buffer state.

Here is the call graph for this function:



Here is the caller graph for this function:



7.53.3.5 void yy_delete_buffer (YY_BUFFER_STATE b)

Destroy the buffer.

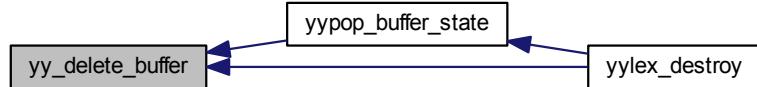
Parameters

<i>b</i>	a buffer created with yy_create_buffer()
----------	--

Here is the call graph for this function:



Here is the caller graph for this function:



7.53.3.6 void yy_flush_buffer (YY_BUFFER_STATE b)

Discard all buffered characters. On the next scan, YY_INPUT will be called.

Parameters

<i>b</i>	the buffer state to be flushed, usually YY_CURRENT_BUFFER.
----------	--

7.53.3.7 YY_BUFFER_STATE yy_scan_buffer (char * *base*, yy_size_t *size*)

Setup the input buffer state to scan directly from a user-specified character buffer.

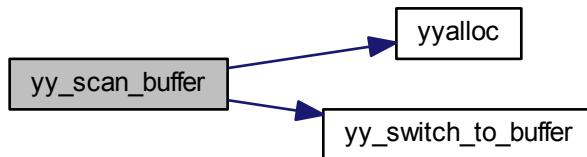
Parameters

<i>base</i>	the character buffer
<i>size</i>	the size in bytes of the character buffer

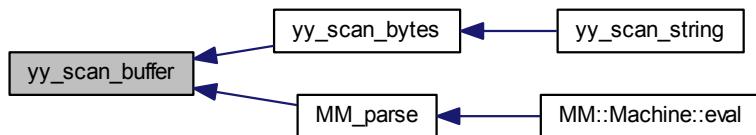
Returns

the newly allocated buffer state object.

Here is the call graph for this function:



Here is the caller graph for this function:

**7.53.3.8 YY_BUFFER_STATE yy_scan_bytes (yyconst char * *yybytes*, int _yybytes_len)**

Setup the input buffer state to scan the given bytes. The next call to [yylex\(\)](#) will scan from a *copy* of *bytes*.

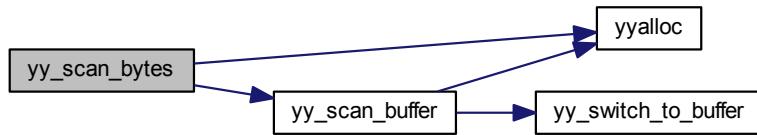
Parameters

<code>yybytes</code>	the byte buffer to scan
<code>_yybytes_len</code>	the number of bytes in the buffer pointed to by <code>bytes</code> .

Returns

the newly allocated buffer state object.

Here is the call graph for this function:



Here is the caller graph for this function:



7.53.3.9 YY_BUFFER_STATE `yy_scan_string (yyconst char * yystr)`

Setup the input buffer state to scan a string. The next call to [yylex\(\)](#) will scan from a *copy* of `str`.

Parameters

<code>yystr</code>	a NUL-terminated string to scan
--------------------	---------------------------------

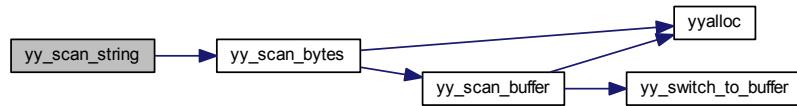
Returns

the newly allocated buffer state object.

Note

If you want to scan bytes that may contain NUL values, then use [yy_scan_bytes\(\)](#) instead.

Here is the call graph for this function:



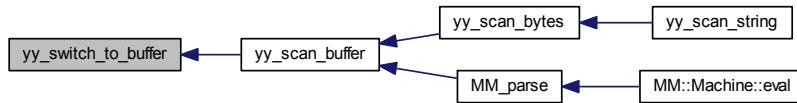
7.53.3.10 void yy_switch_to_buffer (YY_BUFFER_STATE new_buffer)

Switch to a different input buffer.

Parameters

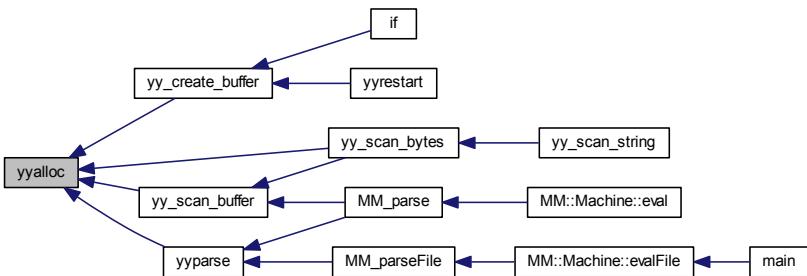
<i>new_buffer</i>	The new input buffer.
-------------------	-----------------------

Here is the caller graph for this function:



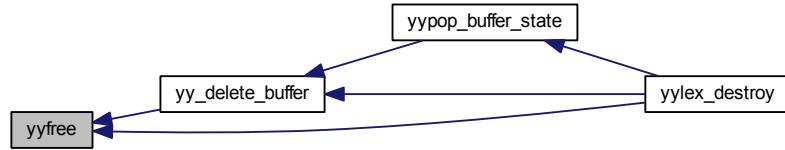
7.53.3.11 void * yyalloc (yy_size_t size)

Here is the caller graph for this function:



7.53.3.12 void yyfree (void * *ptr*)

Here is the caller graph for this function:

**7.53.3.13 int yyget_debug (void)****7.53.3.14 YY_EXTRA_TYPE yyget_extra (void)****7.53.3.15 FILE * yyget_in (void)**

Get the input stream.

7.53.3.16 int yyget_leng (void)

Get the length of the current token.

7.53.3.17 int yyget_lineno (void)

Get the current line number.

7.53.3.18 FILE * yyget_out (void)

Get the output stream.

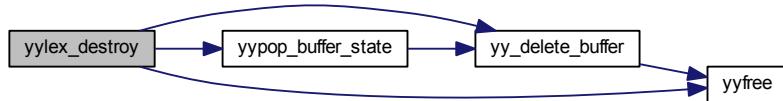
7.53.3.19 char * yyget_text (void)

Get the current token.

7.53.3.20 int yylex (void)

7.53.3.21 int yylex_destroy (void)

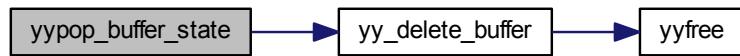
Here is the call graph for this function:



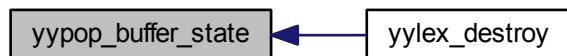
7.53.3.22 void yypop_buffer_state (void)

Removes and deletes the top of the stack, if present. The next element becomes the new top.

Here is the call graph for this function:



Here is the caller graph for this function:



7.53.3.23 void yypush_buffer_state (YY_BUFFER_STATE new_buffer)

Pushes the new state onto the stack. The new state becomes the current state. This function will allocate the stack if necessary.

Parameters

<i>new_buffer</i>	The new state.
-------------------	----------------

7.53.3.24 void * yyrealloc (void * ptr, yy_size_t size)

7.53.3.25 void yyrestart (FILE * *input_file*)

Immediately switch to a different input stream.

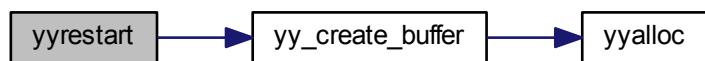
Parameters

<i>input_file</i>	A readable stream.
-------------------	--------------------

Note

This function does not reset the start condition to INITIAL .

Here is the call graph for this function:



7.53.3.26 void yyset_debug (int *debug_flag*)

7.53.3.27 void yyset_extra (YY_EXTRA_TYPE *user_defined*)

7.53.3.28 void yyset_in (FILE * *in_str*)

Set the input stream. This does not discard the current input buffer.

Parameters

<i>in_str</i>	A readable stream.
---------------	--------------------

See Also

[yy_switch_to_buffer](#)

7.53.3.29 void yyset_lineno (int *line_number*)

Set the current line number.

Parameters

<i>line_number</i>	
--------------------	--

7.53.3.30 void yyset_out (FILE * *out_str*)

7.53.4 Variable Documentation

7.53.4.1 bool inID = false

7.53.4.2 register int yy_act

7.53.4.3 register char * yy_bp

7.53.4.4 register char* yy_cp

7.53.4.5 YY_DECL

Initial value:

```
{
    register yy_state_type yy_current_state
```

The main scanner function which does all the work.

7.53.4.6 int yy flex_debug = 0

7.53.4.7 int yycolumn = 1

7.53.4.8 char yyid[256] = { 0 }

7.53.4.9 int yyidpos = 0

7.53.4.10 FILE * yyin = (FILE *) 0

7.53.4.11 int yyleng

7.53.4.12 int yylineno = 1

7.53.4.13 FILE * yyout = (FILE *) 0

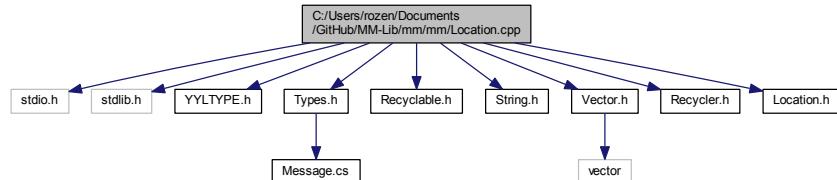
7.53.4.14 char yystr[256] = { 0 }

7.53.4.15 char * yytext

7.54 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Location.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "String.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
```

Include dependency graph for Location.cpp:



Namespaces

- [MM](#)

7.54.1 Detailed Description

Author

Riemer van Rozen

Date

July 27th 2013

7.55 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Location.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Location](#)

Namespaces

- [MM](#)

7.55.1 Detailed Description

Author

Riemer van Rozen

Date

July 27th 2013

7.56 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Machine.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Transformation.h"
#include "Program.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "FlowEvent.h"
#include "TriggerEvent.h"
#include "Failure.h"
#include "Enablement.h"
#include "Disablement.h"
#include "Activation.h"
#include "Violation.h"
#include "Prevention.h"
#include "Operator.h"
#include "Exp.h"
#include "Assertion.h"
#include "Deletion.h"
#include "Edge.h"
#include "StateEdge.h"
#include "FlowEdge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "PoolNodeBehavior.h"
#include "SourceNodeBehavior.h"
#include "DrainNodeBehavior.h"
#include "GateNodeBehavior.h"
#include "RefNodeBehavior.h"
#include "ConverterNodeBehavior.h"
#include "Declaration.h"
#include "InterfaceNode.h"
#include "Definition.h"
#include "Instance.h"
#include "ValExp.h"
#include "UnExp.h"
#include "BinExp.h"
#include "DieExp.h"
#include "RangeValExp.h"
#include "BooleanValExp.h"
#include "NumberValExp.h"
#include "OverrideExp.h"
#include "ActiveExp.h"
#include "AllExp.h"
#include "AliasExp.h"
Generated on Wed Apr 16 2014 13:09:49 for MM by Doxygen
#include "OneExp.h"
#include "VarExp.h"
#include "Reflector.h"
#include "Evaluator.h"
```

Functions

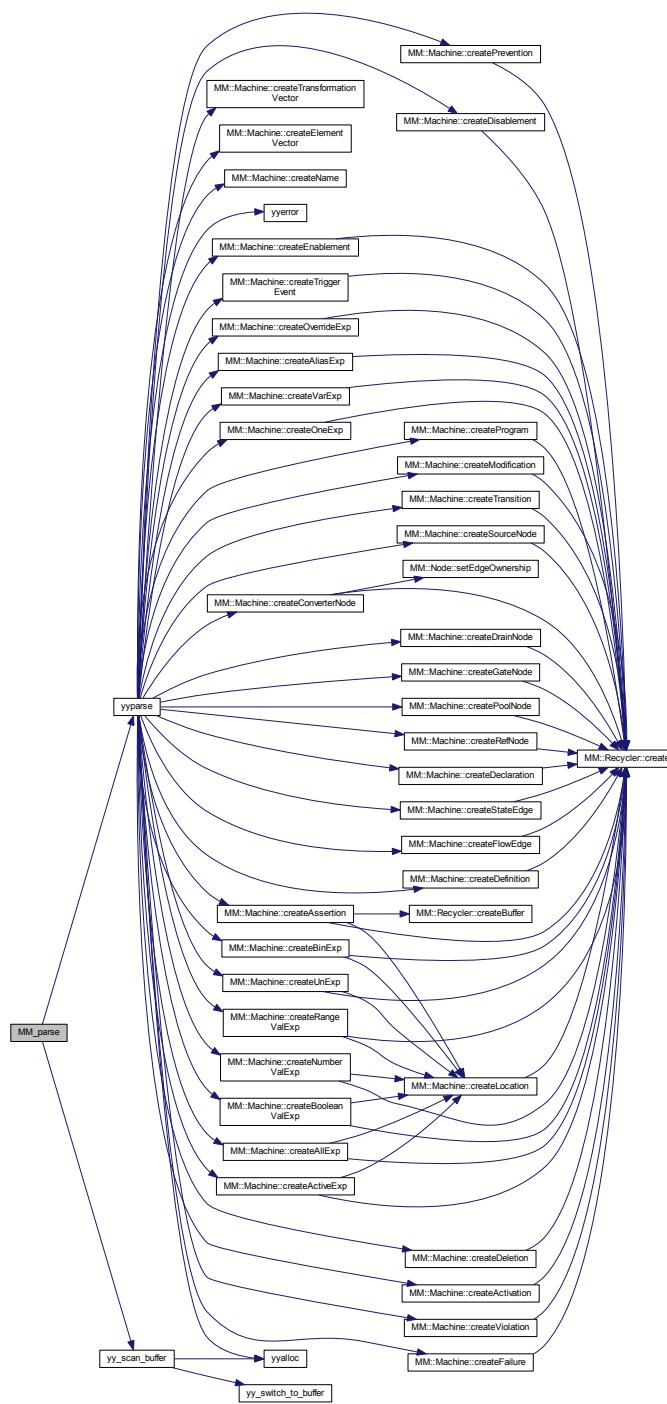
- `MM::Program * MM_parse (MM::Machine *machine, const MM::CHAR *input)`

- `MM::Program * MM_parseFile (MM::Machine *machine, const MM::CHAR *input)`

7.56.1 Function Documentation

7.56.1.1 MM::Program* MM_parse (MM::Machine * machine, const MM::CHAR * input)

Here is the call graph for this function:

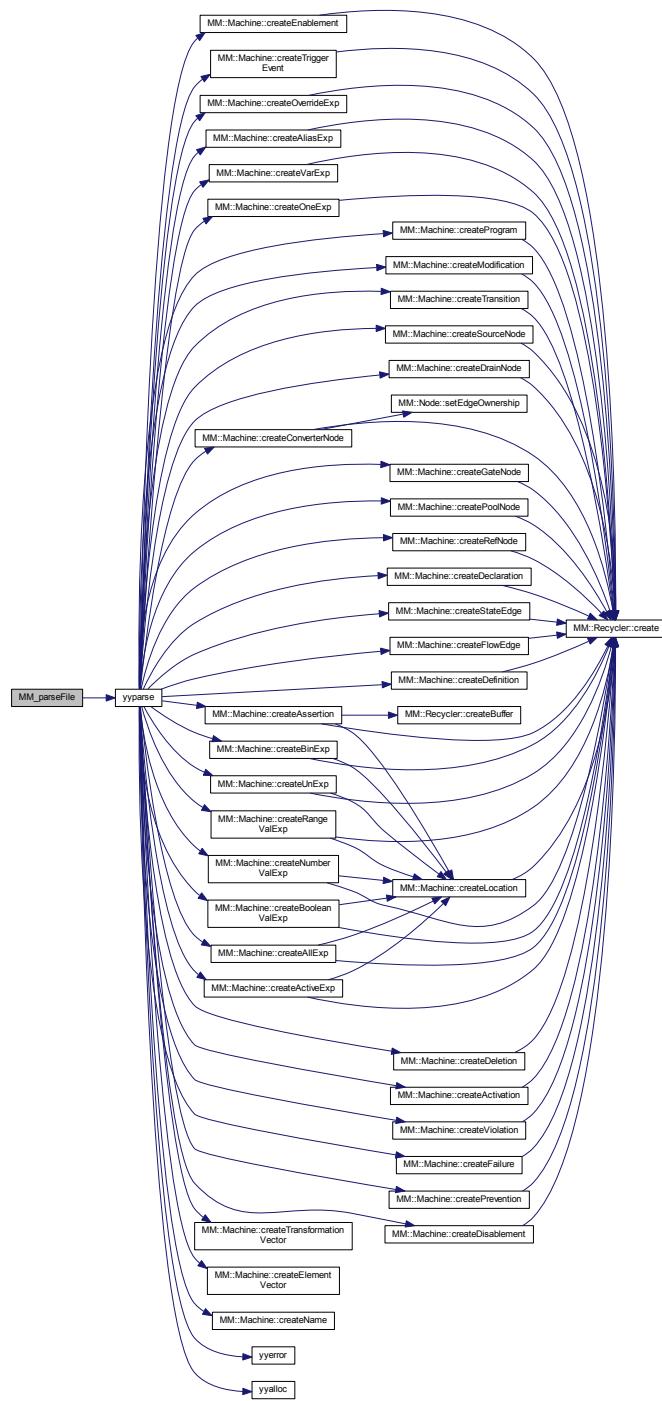


Here is the caller graph for this function:

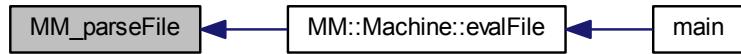


7.56.1.2 MM::Program* MM_parseFile (MM::Machine * machine, const MM::CHAR * input)

Here is the call graph for this function:

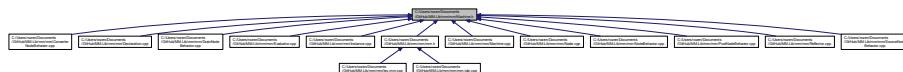


Here is the caller graph for this function:



7.57 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Machine.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Machine](#)
- class [MM::Machine::InstanceObserver](#)
- class [MM::Machine::Delegate](#)

Namespaces

- [MM](#)

7.57.1 Detailed Description

Author

Riemer van Rozen

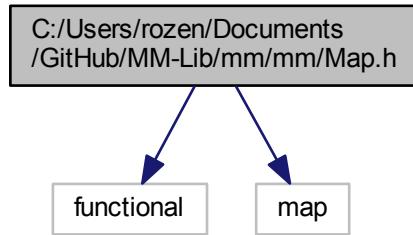
Date

July 16th 2013

7.58 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Map.h File Reference

```
#include <functional>
#include <map>
```

Include dependency graph for Map.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Map< MAP_KEY, MAP_VALUE, COMPARE >](#)
- class [MM::Map< MAP_KEY, MAP_VALUE, COMPARE >::Iterator](#)

Namespaces

- [MM](#)

7.58.1 Detailed Description

Author

Riemer van Rozen

Date

September 13th 2013

7.59 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Message.cs File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- [MM](#)
- package [LibMM](#)

Enumerations

- enum LibMM.MESSAGE {
 LibMM.MESSAGE.MSG_ERROR, LibMM.MESSAGE.MSG_NEW_TYPE, LibMM.MESSAGE.MSG_NEW_DECL,
 LibMM.MESSAGE.MSG_NEW_POOL,
 LibMM.MESSAGE.MSG_NEW_SOURCE, LibMM.MESSAGE.MSG_NEW_DRAIN, LibMM.MESSAGE.MSG_NEW_GATE,
 LibMM.MESSAGE.MSG_NEW_REF,
 LibMM.MESSAGE.MSG_NEW_CONVERTER, LibMM.MESSAGE.MSG_NEW_CONDITION, LibMM.MESSAGE.MSG_NEW_TRIGGER,
 LibMM.MESSAGE.MSG_NEW_FLOW,
 LibMM.MESSAGE.MSG_DEL_TYPE, LibMM.MESSAGE.MSG_DEL_DECL, LibMM.MESSAGE.MSG_DEL_POOL,
 LibMM.MESSAGE.MSG_DEL_SOURCE,
 LibMM.MESSAGE.MSG_DEL_DRAIN, LibMM.MESSAGE.MSG_DEL_GATE, LibMM.MESSAGE.MSG_DEL_REF,
 LibMM.MESSAGE.MSG_DEL_CONVERTER,
 LibMM.MESSAGE.MSG_DEL_CONDITION, LibMM.MESSAGE.MSG_DEL_TRIGGER, LibMM.MESSAGE.MSG_DEL_FLOW,
 LibMM.MESSAGE.MSG_UPD_TYPE,
 LibMM.MESSAGE.MSG_UPD_DECL, LibMM.MESSAGE.MSG_UPD_POOL, LibMM.MESSAGE.MSG_UPD_SOURCE,
 LibMM.MESSAGE.MSG_UPD_DRAIN,
 LibMM.MESSAGE.MSG_UPD_GATE, LibMM.MESSAGE.MSG_UPD_REF, LibMM.MESSAGE.MSG_UPD_CONVERTER,
 LibMM.MESSAGE.MSG_UPD_CONDITION,
 LibMM.MESSAGE.MSG_UPD_TRIGGER, LibMM.MESSAGE.MSG_UPD_FLOW, LibMM.MESSAGE.MSG_NEW_INST,
 LibMM.MESSAGE.MSG_DEL_INST,
 LibMM.MESSAGE.MSG_ADD_VALUE, LibMM.MESSAGE.MSG_SUB_VALUE, LibMM.MESSAGE.MSG_HAS_VALUE,
 LibMM.MESSAGE.MSG_TRIGGER,
 LibMM.MESSAGE.MSG_ACTIVATE, LibMM.MESSAGE.MSG_ENABLE, LibMM.MESSAGE.MSG_DISABLE,
 LibMM.MESSAGE.MSG_VIOLATE,
 LibMM.MESSAGE.MSG_FAIL, LibMM.MESSAGE.MSG_PREVENT }

MESSAGE enumeration defines messages sent to observers.

7.59.1 Detailed Description

Author

Riemer van Rozen

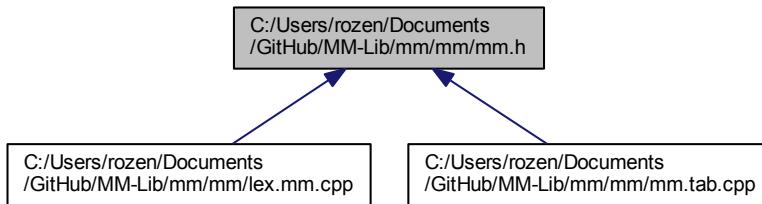
Date

September 13th 2013

7.60 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/mm.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Operator.h"
#include "Exp.h"
#include "Assertion.h"
#include "Deletion.h"
#include "Edge.h"
#include "StateEdge.h"
#include "FlowEdge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "PoolNodeBehavior.h"
#include "SourceNodeBehavior.h"
#include "DrainNodeBehavior.h"
#include "GateNodeBehavior.h"
#include "RefNodeBehavior.h"
#include "ConverterNodeBehavior.h"
#include "Node.h"
#include "Transformation.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "FlowEvent.h"
#include "TriggerEvent.h"
#include "Failure.h"
#include "Disablement.h"
#include "Enablement.h"
#include "Violation.h"
#include "Prevention.h"
#include "Activation.h"
#include "Program.h"
#include "Declaration.h"
#include "InterfaceNode.h"
#include "Definition.h"
#include "Instance.h"
#include "ValExp.h"
#include "UnExp.h"
#include "BinExp.h"
#include "DieExp.h"
#include "RangeValExp.h"
#include "BooleanValExp.h"
#include "NumberValExp.h"
#include "OverrideExp.h"
#include "ActiveExp.h"
#include "AllExp.h"
#include "AliasExp.h"
#include "OneExp.h"
Generated on Wed Apr 16 2014 13:09:49 for MM by Doxygen
#include "VarExp.h"
#include "Evaluator.h"
#include "Reflector.h"
#include "Machine.h"
```

This graph shows which files directly or indirectly include this file:



7.60.1 Detailed Description

Author

Riemer van Rozen

Date

September 23rd 2013

7.61 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/mm.tab.cpp File Reference

```
#include "YYLTYPE.h"
#include "mm.h"
Include dependency graph for mm.tab.cpp:
```



Classes

- union YYSTYPE
- union yalloc

Macros

- #define YYBISON 1
- #define YYBISON_VERSION "2.4.2"
- #define YYSKELETON_NAME "yacc.c"
- #define YYPURE 0
- #define YYPUSH 0
- #define YYPULL 1
- #define YYLSP_NEEDED 1
- #define YYDEBUG 0
- #define YYERROR_VERBOSE 1
- #define YYTOKEN_TABLE 0
- #define YYTOKENTYPE
- #define YYSTYPE_IS_TRIVIAL 1

- `#define YYSTYPE /* obsolescent; will be withdrawn */`
- `#define YYSTYPE_IS_DECLARED 1`
- `#define YYSIZE_T unsigned int`
- `#define YYSIZE_MAXIMUM ((YYSTYPE_T)-1)`
- `#define YY_(msgid) msgid`
- `#define YYUSE(e) ((void)(e))`
- `#define YYID(n) (n)`
- `#define YYSTACK_ALLOC YYMALLOC`
- `#define YYSTACK_FREE YYFREE`
- `#define YYSTACK_ALLOC_MAXIMUM YYSIZE_MAXIMUM`
- `#define YYMALLOC malloc`
- `#define YYFREE free`
- `#define YYSTACK_GAP_MAXIMUM (sizeof (union yyalloc) - 1)`
- `#define YYSTACK_BYTES(N)`
- `#define YYCOPY(To, From, Count)`
- `#define YYSTACK_RELOCATE(Stack_alloc, Stack)`
- `#define YYFINAL 3`
- `#define YYLAST 324`
- `#define YYNTOKENS 71`
- `#define YYNNTS 18`
- `#define YYNRULES 83`
- `#define YYNSTATES 140`
- `#define YYUNDEF TOK 2`
- `#define YYMAXUTOK 325`
- `#define YYTRANSLATE(YYX) ((unsigned int)(YYX) <= YYMAXUTOK ? yytranslate[YYX] : YYUNDEF TOK)`
- `#define YYPACT_NINF -94`
- `#define YYTABLE_NINF -3`
- `#define yyerrok (yyerrstatus = 0)`
- `#define yyclearin (yychar = YYEMPTY)`
- `#define YYEMPTY (-2)`
- `#define YYEOF 0`
- `#define YYACCEPT goto yyacceptlab`
- `#define YYABORT goto yyabortlab`
- `#define YYERROR goto yyerrorlab`
- `#define YYFAIL goto yyerrlab`
- `#define YYRECOVERING() (!yyerrstatus)`
- `#define YYBACKUP(Token, Value)`
- `#define YYTERROR 1`
- `#define YYERRCODE 256`
- `#define YYRHSLOC(Rhs, K) ((Rhs)[K])`
- `#define YYLLOC_DEFAULT(Current, Rhs, N)`
- `#define YY_LOCATION_PRINT(File, Loc) ((void) 0)`
- `#define YYLEX yylex ()`
- `#define YYDPRINTF(Args)`
- `#define YY_SYMBOL_PRINT(Title, Type, Value, Location)`
- `#define YY_STACK_PRINT(Bottom, Top)`
- `#define YY_REDUCE_PRINT(Rule)`
- `#define YYINITDEPTH 200`
- `#define YYMAXDEPTH 10000`
- `#define YYPOPSTACK(N) (yyvsp -= (N), yyssp -= (N), yylsp -= (N))`

Typedefs

- `typedef union YYSTYPE YYSTYPE`
- `typedef unsigned char yytype_uint8`
- `typedef short int yytype_int8`
- `typedef unsigned short int yytype_uint16`
- `typedef short int yytype_int16`
- `typedef yy_buffer_state * YY_BUFFER_STATE`

Enumerations

- `enum yytokentype {`
- `ID = 258, PRIVATE = 259, IN = 260, OUT = 261,`
- `INOUT = 262, PASSIVE = 263, AUTO = 264, USER = 265,`
- `START = 266, PUSH = 267, PULL = 268, ALL = 269,`
- `ANY = 270, FROM = 271, TO = 272, OF = 273,`
- `EQ = 274, NE = 275, LT = 276, GT = 277,`
- `LE = 278, GE = 279, AND = 280, OR = 281,`
- `NOT = 282, TRUE = 283, FALSE = 284, ACTIVE = 285,`
- `ASSERT = 286, DELETE = 287, ACTIVATE = 288, DISABLE = 289,`
- `ENABLE = 290, PREVENT = 291, TRIGGER = 292, FAIL = 293,`
- `VIOLATE = 294, MODIFY = 295, STEP = 296, AT = 297,`
- `MAX = 298, DICE = 299, ALIAS = 300, DOT = 301,`
- `PER = 302, REF = 303, DOT_GT = 304, SUB_GT = 305,`
- `LCURLY = 306, RCURLY = 307, COLON = 308, RANGE = 309,`
- `PERCENT = 310, SOURCE = 311, DRAIN = 312, POOL = 313,`
- `GATE = 314, CONVERTER = 315, SUB = 316, ADD = 317,`
- `MUL = 318, DIV = 319, FPVAL = 320, STRING = 321,`
- `ADDITION = 322, UNM = 323, RPAREN = 324, LPAREN = 325,`
- `ID = 258, PRIVATE = 259, IN = 260, OUT = 261,`
- `INOUT = 262, PASSIVE = 263, AUTO = 264, USER = 265,`
- `START = 266, PUSH = 267, PULL = 268, ALL = 269,`
- `ANY = 270, FROM = 271, TO = 272, OF = 273,`
- `EQ = 274, NE = 275, LT = 276, GT = 277,`
- `LE = 278, GE = 279, AND = 280, OR = 281,`
- `NOT = 282, TRUE = 283, FALSE = 284, ACTIVE = 285,`
- `ASSERT = 286, DELETE = 287, ACTIVATE = 288, DISABLE = 289,`
- `ENABLE = 290, PREVENT = 291, TRIGGER = 292, FAIL = 293,`
- `VIOLATE = 294, MODIFY = 295, STEP = 296, AT = 297,`
- `MAX = 298, DICE = 299, ALIAS = 300, DOT = 301,`
- `PER = 302, REF = 303, DOT_GT = 304, SUB_GT = 305,`
- `LCURLY = 306, RCURLY = 307, COLON = 308, RANGE = 309,`
- `PERCENT = 310, SOURCE = 311, DRAIN = 312, POOL = 313,`
- `GATE = 314, CONVERTER = 315, SUB = 316, ADD = 317,`
- `MUL = 318, DIV = 319, FPVAL = 320, STRING = 321,`
- `ADDITION = 322, UNM = 323, RPAREN = 324, LPAREN = 325 }`

Functions

- `int yylex ()`
- `void yyerror (char *s)`
- `for (yylen=0;yystr[yylen];yylen++) continue`
- `while ((*yyd++==*yys++)!= '\0') continue`
- `YYUSE (yylocationp)`
- `if (!yymsg) yymsg`
- `YY_SYMBOL_PRINT (yymsg, yytype, yyvaluep, yylocationp)`

- `switch (yytype)`
- `int yyparse ()`
- `YY_BUFFER_STATE yy_scan_buffer (char *, size_t)`
- `int main (int argc, const char *argv[])`
- `MM::Program * MM_parse (MM::Machine *machine, const MM::CHAR *input)`
- `MM::Program * MM_parseFile (MM::Machine *machine, const MM::CHAR *file)`

Variables

- `MM::Machine * mm`
- `MM::Program * program`
- `int yylineno`
- `int yycolumn`
- `return yylen`
- `const char * yysrc`
- `const char * yrs = yysrc`
- `return yd`
- `int yytype`
- `YYSTYPE * yyvaluep`
- `YYLTYPE * yylocationp`
- `int yychar`
- `YYSTYPE yylval`
- `YYLTYPE yylloc`
- `int yynerrs`
- `FILE * yyin`

7.61.1 Macro Definition Documentation

- 7.61.1.1 `#define YY_(msgid) msgid`
- 7.61.1.2 `#define YY_LOCATION_PRINT(File, Loc) ((void) 0)`
- 7.61.1.3 `#define YY_REDUCE_PRINT(Rule)`
- 7.61.1.4 `#define YY_STACK_PRINT(Bottom, Top)`
- 7.61.1.5 `#define YY_SYMBOL_PRINT(Title, Type, Value, Location)`
- 7.61.1.6 `#define YYABORT goto yyabortlab`
- 7.61.1.7 `#define YYACCEPT goto yyacceptlab`
- 7.61.1.8 `#define YYBACKUP(Token, Value)`

Value:

```
do
  if (yychar == YYEMPTY && yylen == 1)
  {
    yychar = (Token);
    yylval = (Value);
    yytoken = YYTRANSLATE (yychar);
    YYPOPSTACK (1);
    goto yybackup;
  }
  else
  {
    yyerror (YY_("syntax error: cannot back up")); \
    YYERROR;
  }
while (YYID (0))
```

7.61.1.9 #define YYBISON 1
 7.61.1.10 #define YYBISON_VERSION "2.4.2"
 7.61.1.11 #define yyclarin (yychar = YYEMPTY)
 7.61.1.12 #define YYCOPY(To, From, Count)

Value:

```
do          \
{           \
  YYSIZE_T yyi;           \
  for (yyi = 0; yyi < (Count); yyi++) \
    (To)[yyi] = (From)[yyi];           \
}           \
while (YYID (0))
```

7.61.1.13 #define YYDEBUG 0
 7.61.1.14 #define YYDPRINTF(Args)
 7.61.1.15 #define YYEMPTY (-2)
 7.61.1.16 #define YYEOF 0
 7.61.1.17 #define YYERRCODE 256
 7.61.1.18 #define yyerrok (yyerrstatus = 0)
 7.61.1.19 #define YYERROR goto yyerrorlab
 7.61.1.20 #define YYERROR_VERBOSE 1
 7.61.1.21 #define YYFAIL goto yyerrlab
 7.61.1.22 #define YYFINAL 3
 7.61.1.23 #define YYFREE free
 7.61.1.24 #define YYID(n)(n)
 7.61.1.25 #define YYINITDEPTH 200
 7.61.1.26 #define YYLAST 324
 7.61.1.27 #define YYLEX yylex ()
 7.61.1.28 #define YYLLOC_DEFAULT(Current, Rhs, N)

Value:

```
do          \
{           \
  if (YYID (N))           \
  {           \
    (Current).first_line   = YYRHSLOC (Rhs, 1).first_line;   \
    (Current).first_column = YYRHSLOC (Rhs, 1).first_column; \
    (Current).last_line    = YYRHSLOC (Rhs, N).last_line;    \
    (Current).last_column  = YYRHSLOC (Rhs, N).last_column; \
  }           \
}
```

```

    else
    {
        (Current).first_line  = (Current).last_line   =
            YYRHSLOC (Rhs, 0).last_line;
        (Current).first_column = (Current).last_column =
            YYRHSLOC (Rhs, 0).last_column;
    }
    while (YYID (0))

```

7.61.1.29 #define YYLSP_NEEDED 1

7.61.1.30 #define YYMALLOC malloc

7.61.1.31 #define YYMAXDEPTH 10000

7.61.1.32 #define YYMAXUTOK 325

7.61.1.33 #define YYNNTS 18

7.61.1.34 #define YYNRULES 83

7.61.1.35 #define YYNSTATES 140

7.61.1.36 #define YYNTOKENS 71

7.61.1.37 #define YYPACT_NINF -94

7.61.1.38 #define YYPOPSTACK(N) (yyvsp -= (N), yyssp -= (N), yyfsp -= (N))

7.61.1.39 #define YYPULL 1

7.61.1.40 #define YYPURE 0

7.61.1.41 #define YYPUSH 0

7.61.1.42 #define YYRECOVERING() (!yyerrstatus)

7.61.1.43 #define YYRHSLOC(Rhs, K) ((Rhs)[K])

7.61.1.44 #define YYSIZE_MAXIMUM ((YYSIZE_T)-1)

7.61.1.45 #define YYSIZE_T unsigned int

7.61.1.46 #define YYSKELETON_NAME "yacc.c"

7.61.1.47 #define YYSTACK_ALLOC YYMALLOC

7.61.1.48 #define YYSTACK_ALLOC_MAXIMUM YYSIZE_MAXIMUM

7.61.1.49 #define YYSTACK_BYTES(N)

Value:

```

((N) * (sizeof (yytype_int16) + sizeof (YYSTYPE) + sizeof (
    YYLTYPE)) \
+ 2 * YYSTACK_GAP_MAXIMUM)

```

7.61.1.50 #define YYSTACK_FREE YYFREE
 7.61.1.51 #define YYSTACK_GAP_MAXIMUM (sizeof (union yyalloc) - 1)
 7.61.1.52 #define YYSTACK_RELOCATE(Stack_alloc, Stack)

Value:

```
do
{
    YYCOPY (&yyptr->Stack_alloc, Stack, yysize);
    Stack = &yyptr->Stack_alloc;
    yynewbytes = yystacksize * sizeof (*Stack) + YYSTACK_GAP_MAXIMUM;
    yyptr += yynewbytes / sizeof (*yyptr);
}
while (YYID (0))
```

7.61.1.53 #define YYSTYPE /* obsolescent; will be withdrawn */
 7.61.1.54 #define YYSTYPE_IS_DECLARED 1
 7.61.1.55 #define YYSTYPE_IS_TRIVIAL 1
 7.61.1.56 #define YYTABLE_NINF -3
 7.61.1.57 #define YYTERROR 1
 7.61.1.58 #define YYTOKEN_TABLE 0
 7.61.1.59 #define YYTOKENTYPE
 7.61.1.60 #define YYTRANSLATE(YYX) ((unsigned int)(YYX) <= YYMAXUTOK ? yytranslate[YYX] : YYUNDEF TOK)
 7.61.1.61 #define YYUNDEF TOK 2
 7.61.1.62 #define YYUSE(e) ((void)(e))

7.61.2 Typedef Documentation

7.61.2.1 typedef yy_buffer_state* YY_BUFFER_STATE
 7.61.2.2 typedef union YYSTYPE YYSTYPE
 7.61.2.3 typedef short int yytype_int16
 7.61.2.4 typedef short int yytype_int8
 7.61.2.5 typedef unsigned short int yytype_uint16
 7.61.2.6 typedef unsigned char yytype_uint8

7.61.3 Enumeration Type Documentation

7.61.3.1 enum yytokentype

Enumerator

ID

PRIVATE
IN
OUT
INOUT
PASSIVE
AUTO
USER
START
PUSH
PULL
ALL
ANY
FROM
TO
OF
EQ
NE
LT
GT
LE
GE
AND
OR
NOT
TRUE
FALSE
ACTIVE
ASSERT
DELETE
ACTIVATE
DISABLE
ENABLE
PREVENT
TRIGGER
FAIL
VIOLATE
MODIFY
STEP
AT
MAX
DICE
ALIAS
DOT
PER
REF

DOT_GT
SUB_GT
LCURLY
RCURLY
COLON
RANGE
PERCENT
SOURCE
DRAIN
POOL
GATE
CONVERTER
SUB
ADD
MUL
DIV
FPVAL
STRING
ADDITION
UNM
RPAREN
LPAREN
ID
PRIVATE
IN
OUT
INOUT
PASSIVE
AUTO
USER
START
PUSH
PULL
ALL
ANY
FROM
TO
OF
EQ
NE
LT
GT
LE
GE
AND

OR
NOT
TRUE
FALSE
ACTIVE
ASSERT
DELETE
ACTIVATE
DISABLE
ENABLE
PREVENT
TRIGGER
FAIL
VIOLATE
MODIFY
STEP
AT
MAX
DICE
ALIAS
DOT
PER
REF
DOT_GT
SUB_GT
LCURLY
RCURLY
COLON
RANGE
PERCENT
SOURCE
DRAIN
POOL
GATE
CONVERTER
SUB
ADD
MUL
DIV
FPVAL
STRING
ADDITION
UNM
RPAREN
LPAREN

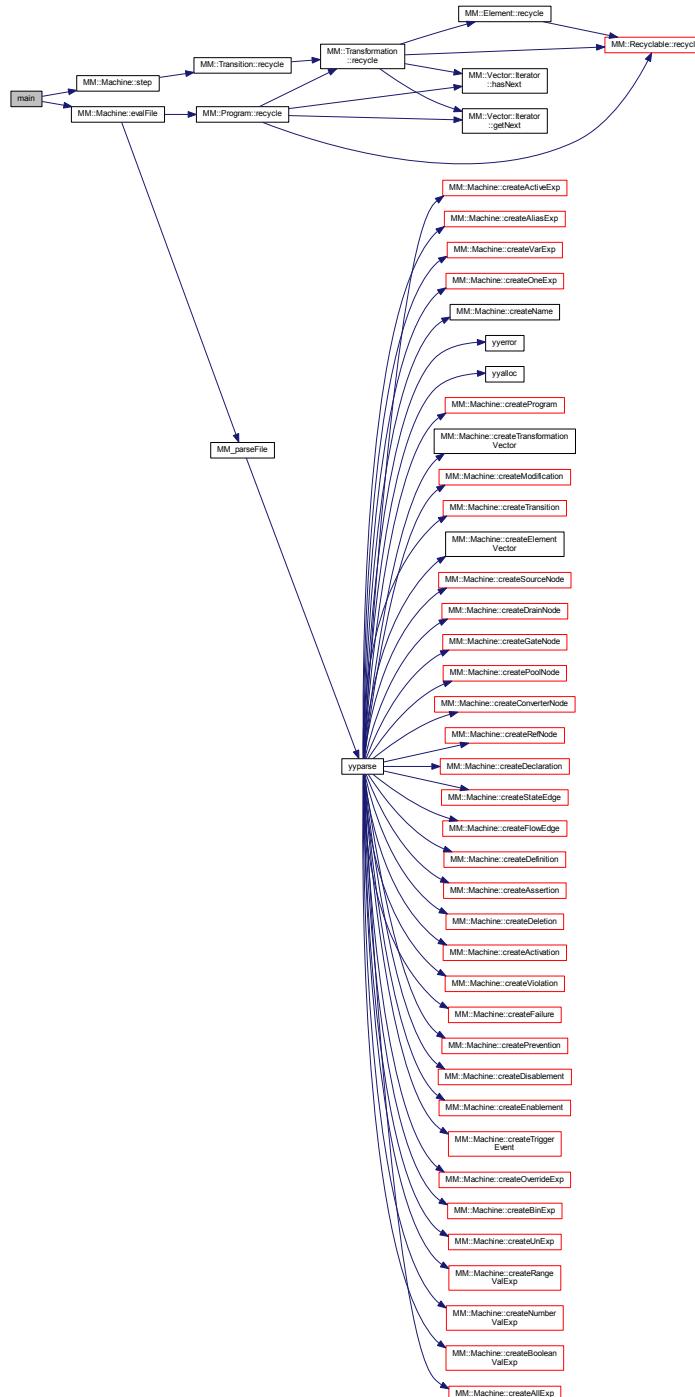
7.61.4 Function Documentation

7.61.4.1 `for(yylen = 0; yystr[yylen]; yylen++)`

7.61.4.2 `if(!yymsg)`

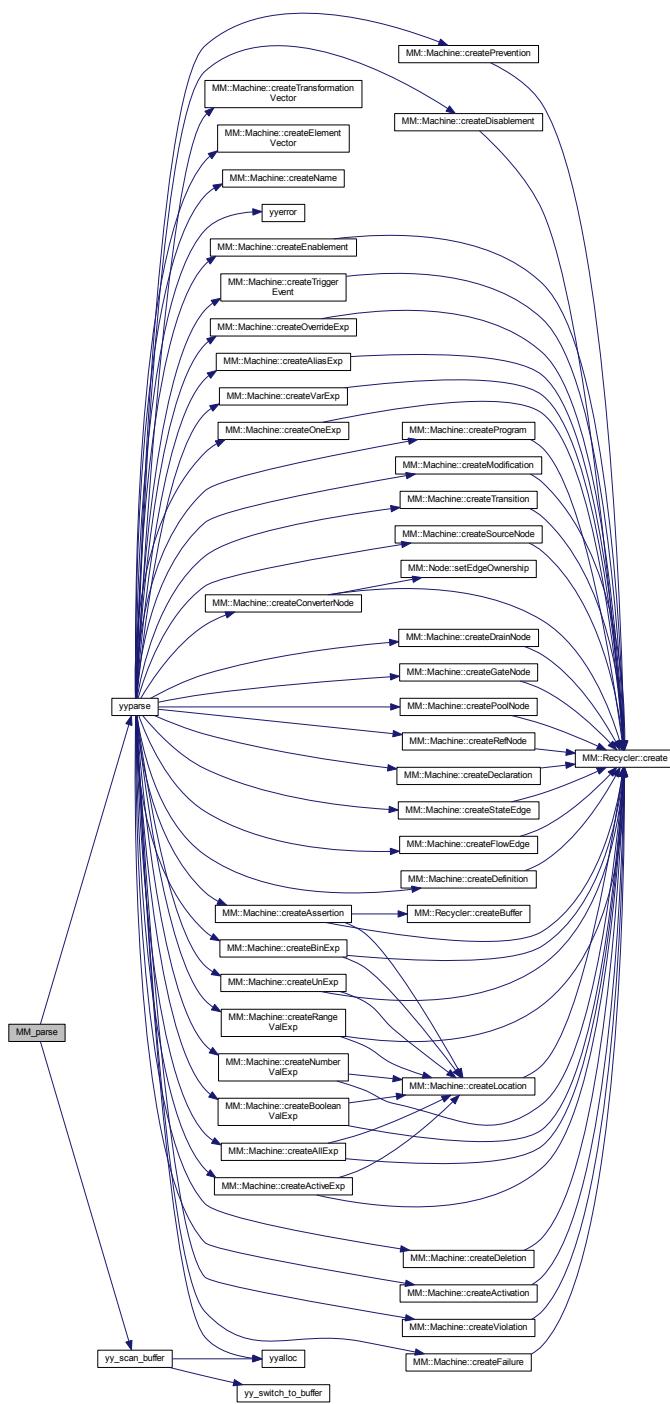
7.61.4.3 `int main(int argc, const char * argv[])`

Here is the call graph for this function:



7.61.4.4 MM::Program* MM_parse (MM::Machine * machine, const MM::CHAR * input)

Here is the call graph for this function:

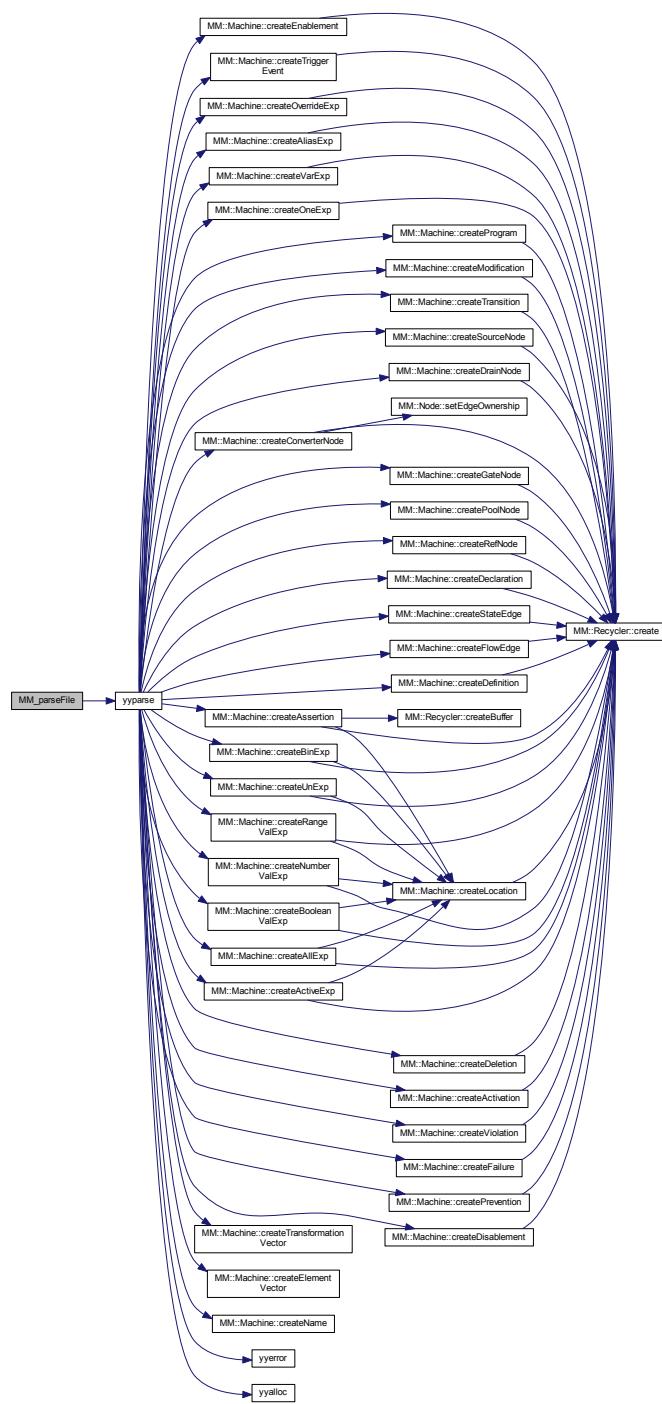


Here is the caller graph for this function:

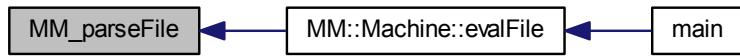


7.61.4.5 MM::Program* MM_parseFile (MM::Machine * machine, const MM::CHAR * file)

Here is the call graph for this function:



Here is the caller graph for this function:



7.61.4.6 `switch(yytype)`

7.61.4.7 `while(*++=++yys! = '\0')`

7.61.4.8 `YY_BUFFER_STATE yy_scan_buffer(char *base, yy_size_t size)`

Setup the input buffer state to scan directly from a user-specified character buffer.

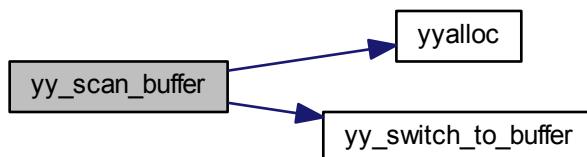
Parameters

<code>base</code>	the character buffer
<code>size</code>	the size in bytes of the character buffer

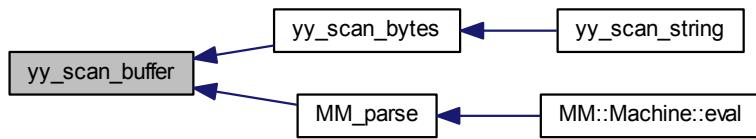
Returns

the newly allocated buffer state object.

Here is the call graph for this function:



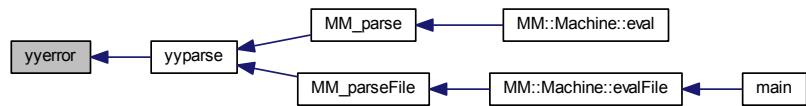
Here is the caller graph for this function:



7.61.4.9 YY_SYMBOL_PRINT (*yymsg* , *yytype* , *yyvaluep* , *yylocationp*)

7.61.4.10 void yyerror (*char * s*)

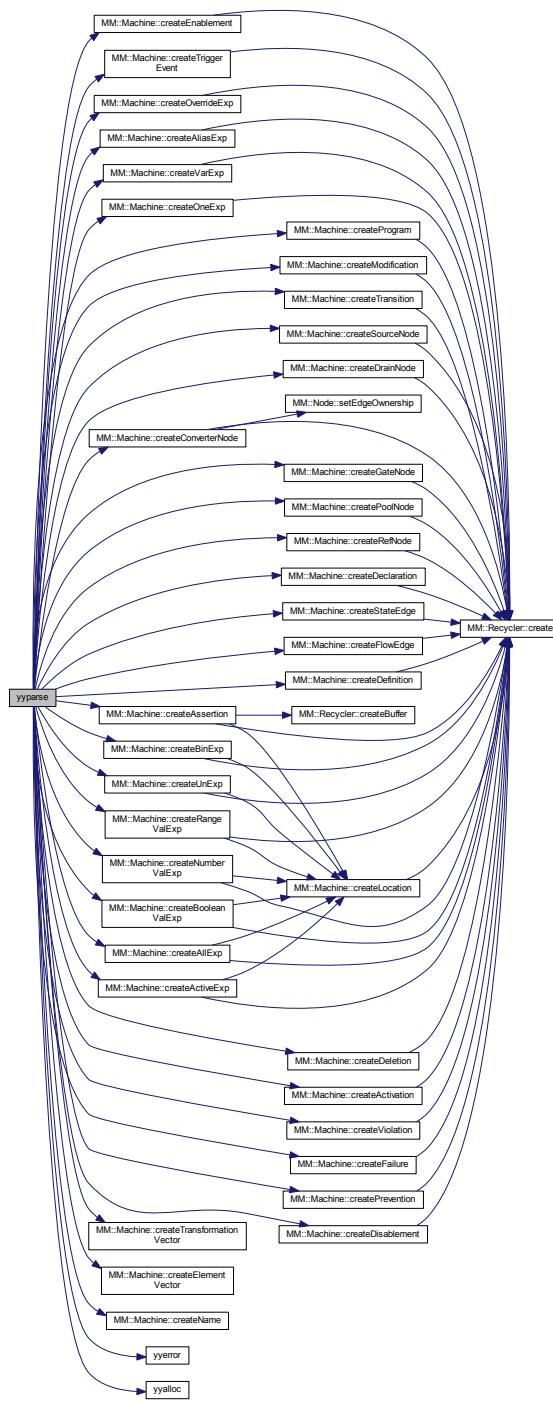
Here is the caller graph for this function:



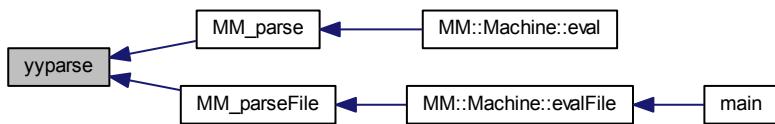
7.61.4.11 int yylex ()

7.61.4.12 int yyparse ()

Here is the call graph for this function:



Here is the caller graph for this function:



7.61.4.13 YYUSE(yylocationp)

7.61.5 Variable Documentation

7.61.5.1 **MM::Machine** * mm

7.61.5.2 **MM::Program** * program

7.61.5.3 int yychar

7.61.5.4 int yycolumn

7.61.5.5 return yyd

7.61.5.6 FILE* yyin

7.61.5.7 return yylen

7.61.5.8 int yylineno

7.61.5.9 YYLTYPE yylloc

7.61.5.10 YYLTYPE* yylocationp

7.61.5.11 YYSTYPE yylval

7.61.5.12 int yynerrs

7.61.5.13 const char* yys = yysrc

7.61.5.14 const char* yysrc

7.61.5.15 int yytype

7.61.5.16 YYSTYPE* yyvaluep

7.62 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm.tab.hpp File Reference

Classes

- union **YYSTYPE**
- struct **YYLTYPE**

Macros

- `#define YYSTYPE_IS_TRIVIAL 1`
- `#define YYSTYPE YYSTYPE /* obsolescent; will be withdrawn */`
- `#define YYSTYPE_IS_DECLARED 1`
- `#define YYLTYPE YYLTYPE /* obsolescent; will be withdrawn */`
- `#define YYLTYPE_IS_DECLARED 1`
- `#define YYLTYPE_IS_TRIVIAL 1`

Typedefs

- `typedef union YYSTYPE YYSTYPE`
- `typedef struct YYLTYPE YYLTYPE`

Enumerations

- enum `ytokentype {`
- `ID = 258, PRIVATE = 259, IN = 260, OUT = 261,`
- `INOUT = 262, PASSIVE = 263, AUTO = 264, USER = 265,`
- `START = 266, PUSH = 267, PULL = 268, ALL = 269,`
- `ANY = 270, FROM = 271, TO = 272, OF = 273,`
- `EQ = 274, NE = 275, LT = 276, GT = 277,`
- `LE = 278, GE = 279, AND = 280, OR = 281,`
- `NOT = 282, TRUE = 283, FALSE = 284, ACTIVE = 285,`
- `ASSERT = 286, DELETE = 287, ACTIVATE = 288, DISABLE = 289,`
- `ENABLE = 290, PREVENT = 291, TRIGGER = 292, FAIL = 293,`
- `VIOLATE = 294, MODIFY = 295, STEP = 296, AT = 297,`
- `MAX = 298, DICE = 299, ALIAS = 300, DOT = 301,`
- `PER = 302, REF = 303, DOT_GT = 304, SUB_GT = 305,`
- `LCURLY = 306, RCURLY = 307, COLON = 308, RANGE = 309,`
- `PERCENT = 310, SOURCE = 311, DRAIN = 312, POOL = 313,`
- `GATE = 314, CONVERTER = 315, SUB = 316, ADD = 317,`
- `MUL = 318, DIV = 319, FPVAL = 320, STRING = 321,`
- `ADDITION = 322, UNM = 323, RPAREN = 324, LPAREN = 325,`
- `ID = 258, PRIVATE = 259, IN = 260, OUT = 261,`
- `INOUT = 262, PASSIVE = 263, AUTO = 264, USER = 265,`
- `START = 266, PUSH = 267, PULL = 268, ALL = 269,`
- `ANY = 270, FROM = 271, TO = 272, OF = 273,`
- `EQ = 274, NE = 275, LT = 276, GT = 277,`
- `LE = 278, GE = 279, AND = 280, OR = 281,`
- `NOT = 282, TRUE = 283, FALSE = 284, ACTIVE = 285,`
- `ASSERT = 286, DELETE = 287, ACTIVATE = 288, DISABLE = 289,`
- `ENABLE = 290, PREVENT = 291, TRIGGER = 292, FAIL = 293,`
- `VIOLATE = 294, MODIFY = 295, STEP = 296, AT = 297,`
- `MAX = 298, DICE = 299, ALIAS = 300, DOT = 301,`
- `PER = 302, REF = 303, DOT_GT = 304, SUB_GT = 305,`
- `LCURLY = 306, RCURLY = 307, COLON = 308, RANGE = 309,`
- `PERCENT = 310, SOURCE = 311, DRAIN = 312, POOL = 313,`
- `GATE = 314, CONVERTER = 315, SUB = 316, ADD = 317,`
- `MUL = 318, DIV = 319, FPVAL = 320, STRING = 321,`
- `ADDITION = 322, UNM = 323, RPAREN = 324, LPAREN = 325 }`

Variables

- `YYSTYPE yylval`
- `YYLTYPE yylloc`

7.62.1 Macro Definition Documentation

7.62.1.1 `#define yytype YYLTYPE /* obsolescent; will be withdrawn */`

7.62.1.2 `#define YYLTYPE_IS_DECLARED 1`

7.62.1.3 `#define YYLTYPE_IS_TRIVIAL 1`

7.62.1.4 `#define YYSTYPE YYSTYPE /* obsolescent; will be withdrawn */`

7.62.1.5 `#define YYSTYPE_IS_DECLARED 1`

7.62.1.6 `#define YYSTYPE_IS_TRIVIAL 1`

7.62.2 Typedef Documentation

7.62.2.1 `typedef struct YYLTYPE YYLTYPE`

7.62.2.2 `typedef union YYSTYPE YYSTYPE`

7.62.3 Enumeration Type Documentation

7.62.3.1 `enum yytokentype`

Enumerator

ID

PRIVATE

IN

OUT

INOUT

PASSIVE

AUTO

USER

START

PUSH

PULL

ALL

ANY

FROM

TO

OF

EQ

NE

LT

GT

LE

GE

AND

OR

NOT

TRUE
FALSE
ACTIVE
ASSERT
DELETE
ACTIVATE
DISABLE
ENABLE
PREVENT
TRIGGER
FAIL
VIOLATE
MODIFY
STEP
AT
MAX
DICE
ALIAS
DOT
PER
REF
DOT_GT
SUB_GT
LCURLY
RCURLY
COLON
RANGE
PERCENT
SOURCE
DRAIN
POOL
GATE
CONVERTER
SUB
ADD
MUL
DIV
FPVAL
STRING
ADDITION
UNM
RPAREN
LPAREN
ID
PRIVATE

IN
OUT
INOUT
PASSIVE
AUTO
USER
START
PUSH
PULL
ALL
ANY
FROM
TO
OF
EQ
NE
LT
GT
LE
GE
AND
OR
NOT
TRUE
FALSE
ACTIVE
ASSERT
DELETE
ACTIVATE
DISABLE
ENABLE
PREVENT
TRIGGER
FAIL
VIOLATE
MODIFY
STEP
AT
MAX
DICE
ALIAS
DOT
PER
REF
DOT_GT

SUB_GT
LCURLY
RCURLY
COLON
RANGE
PERCENT
SOURCE
DRAIN
POOL
GATE
CONVERTER
SUB
ADD
MUL
DIV
FPVAL
STRING
ADDITION
UNM
RPAREN
LPAREN

7.62.4 Variable Documentation

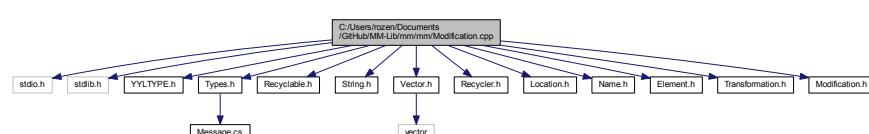
7.62.4.1 YYLTYPE yyloc

7.62.4.2 YYSTYPE yyval

7.63 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Modification.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "String.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Transformation.h"
#include "Modification.h"
```

Include dependency graph for Modification.cpp:



7.64 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Modification.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Modification](#)

Namespaces

- [MM](#)

7.64.1 Detailed Description

Author

Riemer van Rozen

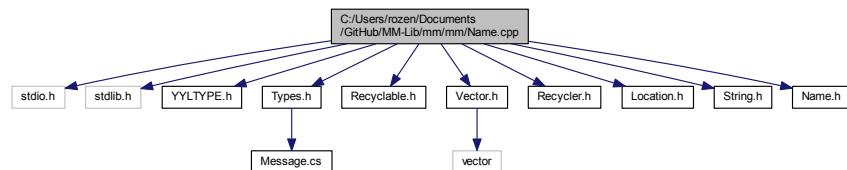
Date

October 16th 2013

7.65 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Name.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "String.h"
#include "Name.h"
```

Include dependency graph for Name.cpp:



Namespaces

- [MM](#)

7.65.1 Detailed Description

Author

Riemer van Rozen

Date

July 27th 2013

7.66 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Name.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Name](#)
- class [MM::Name::Compare](#)

Namespaces

- [MM](#)

7.66.1 Detailed Description

Author

Riemer van Rozen

Date

July 27th 2013

7.67 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Node.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Operator.h"
#include "Exp.h"
#include "Assertion.h"
#include "Deletion.h"
#include "Edge.h"
#include "StateEdge.h"
#include "FlowEdge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "Transformation.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "FlowEvent.h"
#include "TriggerEvent.h"
#include "Failure.h"
#include "Enablement.h"
#include "Disablement.h"
#include "Violation.h"
#include "Prevention.h"
#include "Activation.h"
#include "Program.h"
#include "PoolNodeBehavior.h"
#include "SourceNodeBehavior.h"
#include "DrainNodeBehavior.h"
#include "RefNodeBehavior.h"
#include "GateNodeBehavior.h"
#include "ConverterNodeBehavior.h"
#include "Declaration.h"
#include "InterfaceNode.h"
#include "Definition.h"
#include "Instance.h"
#include "ValExp.h"
#include "UnExp.h"
#include "BinExp.h"
#include "RangeValExp.h"
#include "BooleanValExp.h"
#include "NumberValExp.h"
#include "OverrideExp.h"
#include "ActiveExp.h"
#include "AllExp.h"
#include "DieExp.h"
#include "AliasExp.h"
#include "OneExp.h"
#include "VarExp.h"
#include "Reflector.h"
#include "Evaluator.h"
```

7.68 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Node.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Node](#)
- class [MM::Node::Compare](#)

Namespaces

- [MM](#)

7.68.1 Detailed Description

Author

Riemer van Rozen

Date

July 10th 2013

7.69 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/NodeBehavior.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Operator.h"
#include "Exp.h"
#include "Assertion.h"
#include "Deletion.h"
#include "Edge.h"
#include "StateEdge.h"
#include "FlowEdge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "Transformation.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "FlowEvent.h"
#include "TriggerEvent.h"
#include "Failure.h"
#include "Enablement.h"
#include "Disablement.h"
#include "Violation.h"
#include "Prevention.h"
#include "Activation.h"
#include "Program.h"
#include "PoolNodeBehavior.h"
#include "SourceNodeBehavior.h"
#include "DrainNodeBehavior.h"
#include "RefNodeBehavior.h"
#include "GateNodeBehavior.h"
#include "ConverterNodeBehavior.h"
#include "Declaration.h"
#include "InterfaceNode.h"
#include "Definition.h"
#include "Instance.h"
#include "ValExp.h"
#include "UnExp.h"
#include "BinExp.h"
#include "RangeValExp.h"
#include "BooleanValExp.h"
#include "NumberValExp.h"
#include "OverrideExp.h"
#include "ActiveExp.h"
#include "AllExp.h"
#include "DieExp.h"
#include "AliasExp.h"
#include "OneExp.h"
#include "VarExp.h"
#include "Reflector.h"
```

7.70 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/NodeBehavior.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::NodeBehavior](#)

Namespaces

- [MM](#)

7.70.1 Detailed Description

Author

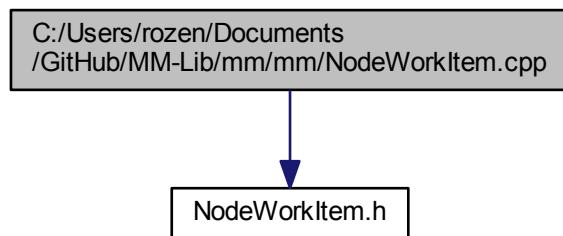
Riemer van Rozen

Date

October 9th 2013

7.71 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/NodeWorkItem.cpp File Reference

```
#include "NodeWorkItem.h"  
Include dependency graph for NodeWorkItem.cpp:
```



7.72 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/NodeWorkItem.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::NodeWorkItem](#)

Namespaces

- [MM](#)

7.72.1 Detailed Description

Author

Riemer van Rozen

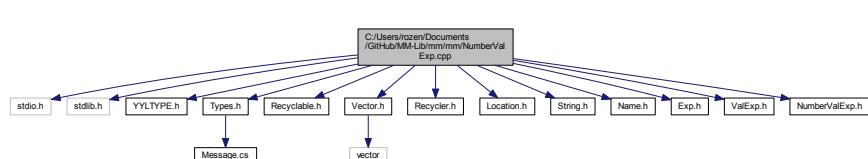
Date

January 27th 2014

7.73 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/NumberValExp.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "String.h"
#include "Name.h"
#include "Exp.h"
#include "ValExp.h"
#include "NumberValExp.h"
```

Include dependency graph for NumberValExp.cpp:



Namespaces

- MM

7.73.1 Detailed Description

Author

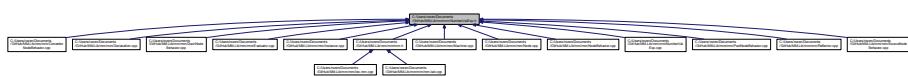
Riemer van Rozen

Date

July 19th 2013

7.74 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/NumberValExp.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class MM::NumberValExp

Namespaces

- MM

7.74.1 Detailed Description

Author

Riemer van Rozen

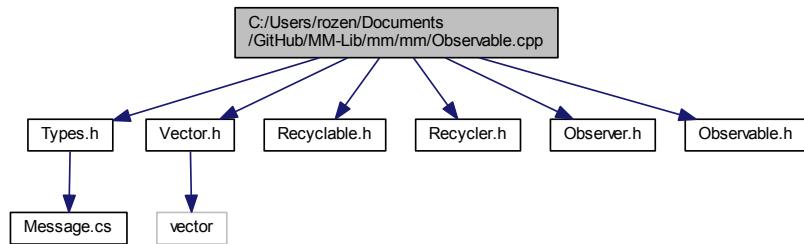
Date

July 19th 2013

7.75 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Observable.cpp File Reference

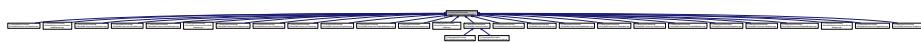
```
#include "Types.h"
#include "Vector.h"
#include "Recyclable.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
```

Include dependency graph for Observable.cpp:



7.76 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Observable.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Observable](#)

Namespaces

- [MM](#)

7.76.1 Detailed Description

Author

Riemer van Rozen

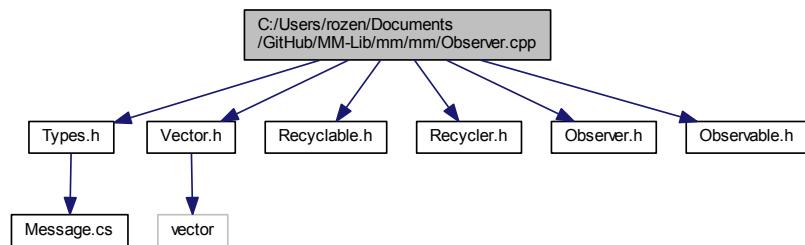
Date

September 26th 2013

7.77 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Observer.cpp File Reference

```
#include "Types.h"
#include "Vector.h"
#include "Recyclable.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
```

Include dependency graph for Observer.cpp:



7.78 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Observer.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Observer](#)

Namespaces

- [MM](#)

7.78.1 Detailed Description

Author

Riemer van Rozen

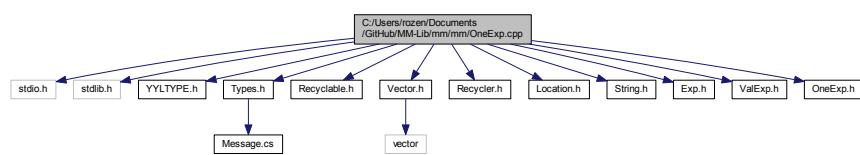
Date

September 26th 2013

7.79 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/OneExp.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "String.h"
#include "Exp.h"
#include "ValExp.h"
#include "OneExp.h"
```

Include dependency graph for OneExp.cpp:



Namespaces

- MM

7.80 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/OneExp.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class MM::OneExp

Namespaces

- MM

7.80.1 Detailed Description

Author

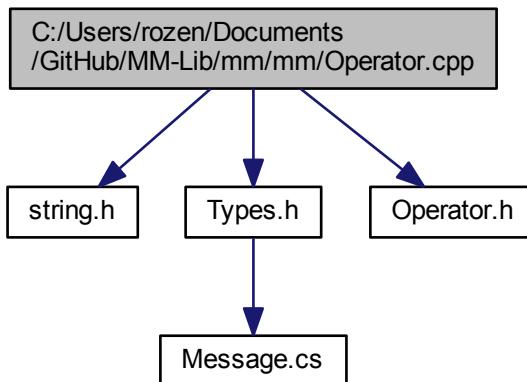
Riemer van Rozen

Date

July 20th 2013

7.81 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Operator.cpp File Reference

```
#include <string.h>
#include "Types.h"
#include "Operator.h"
Include dependency graph for Operator.cpp:
```

**Namespaces**

- MM

7.81.1 Detailed Description

Author

Riemer van Rozen

Date

July 27th 2013

7.82 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Operator.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Operator](#)

Namespaces

- [MM](#)

7.82.1 Detailed Description

Author

Riemer van Rozen

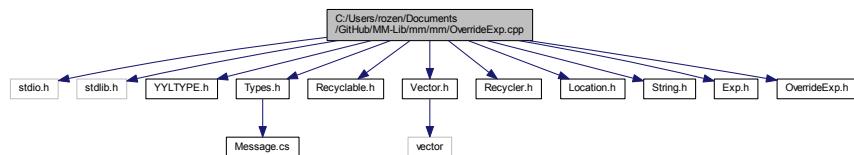
Date

July 27th 2013

7.83 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/OverrideExp.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "String.h"
#include "Exp.h"
#include "OverrideExp.h"
```

Include dependency graph for OverrideExp.cpp:



Namespaces

- [MM](#)

7.83.1 Detailed Description

Author

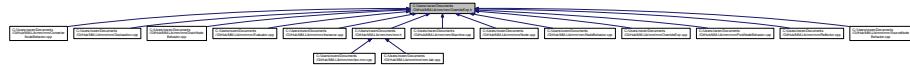
Riemer van Rozen

Date

July 20th 2013

7.84 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/OverrideExp.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::OverrideExp](#)

Namespaces

- [MM](#)

7.84.1 Detailed Description

Author

Riemer van Rozen

Date

July 20th 2013

7.85 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/PoolNodeBehavior.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Transformation.h"
#include "Program.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "FlowEvent.h"
#include "TriggerEvent.h"
#include "Failure.h"
#include "Activation.h"
#include "Enablement.h"
#include "Disablement.h"
#include "Violation.h"
#include "Prevention.h"
#include "Operator.h"
#include "Exp.h"
#include "Assertion.h"
#include "Deletion.h"
#include "Edge.h"
#include "StateEdge.h"
#include "FlowEdge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "PoolNodeBehavior.h"
#include "SourceNodeBehavior.h"
#include "DrainNodeBehavior.h"
#include "GateNodeBehavior.h"
#include "RefNodeBehavior.h"
#include "ConverterNodeBehavior.h"
#include "Declaration.h"
#include "InterfaceNode.h"
#include "Definition.h"
#include "Instance.h"
#include "ValExp.h"
#include "UnExp.h"
#include "BinExp.h"
#include "DieExp.h"
#include "RangeValExp.h"
#include "BooleanValExp.h"
#include "NumberValExp.h"
#include "OverrideExp.h"
#include "ActiveExp.h"
#include "AllExp.h"
#include "AliasExp.h"
#include "OneExp.h"
#include "VarExp.h"
#include "Reflector.h"
```

7.86 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/PoolNodeBehavior.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class MM::PoolNodeBehavior

Namespaces

- MM

7.86.1 Detailed Description

Author

Riemer van Rozen

Date

October 7th 2013

7.87 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Prevention.cpp File Reference

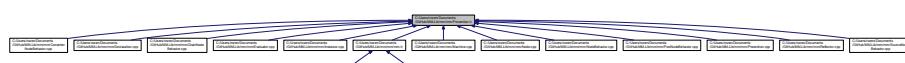
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Transformation.h"
#include "Program.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "Prevention.h"
#include "Operator.h"
#include "Exp.h"
#include "Edge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "Declaration.h"
#include "Definition.h"
#include "Instance.h"
```

Include dependency graph for Prevention.cpp:



7.88 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Prevention.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Prevention](#)

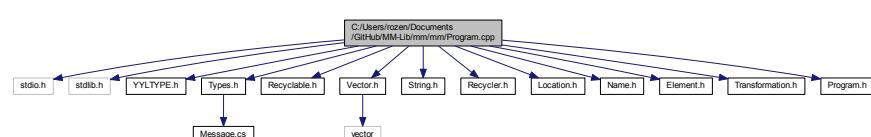
Namespaces

- [MM](#)

7.89 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Program.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "String.h"
#include "Recycler.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Transformation.h"
#include "Program.h"
```

Include dependency graph for Program.cpp:



7.90 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Program.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Program](#)

Namespaces

- [MM](#)

7.90.1 Detailed Description

Author

Riemer van Rozen

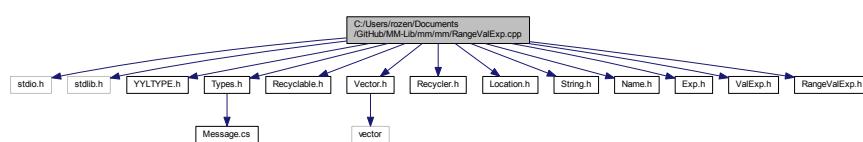
Date

October 16th 2013

7.91 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/RangeValExp.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "String.h"
#include "Name.h"
#include "Exp.h"
#include "ValExp.h"
#include "RangeValExp.h"
```

Include dependency graph for RangeValExp.cpp:



Namespaces

- MM

7.91.1 Detailed Description

Author

Riemer van Rozen

Date

July 19th 2013

7.92 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/RangeValExp.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class MM::RangeValExp

Namespaces

- MM

7.92.1 Detailed Description

Author

Riemer van Rozen

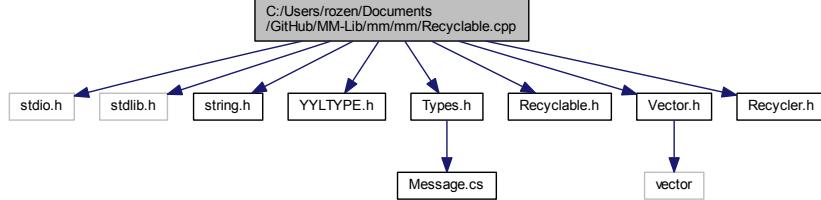
Date

July 19th 2013

7.93 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Recyclable.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
```

Include dependency graph for Recyclable.cpp:



7.94 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Recyclable.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class MM::Recyclable

Namespaces

- MM

7.94.1 Detailed Description

Author

Riemer van Rozen

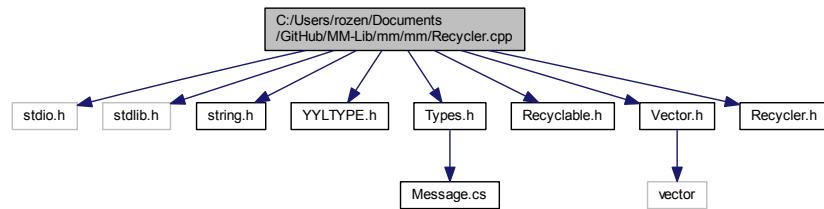
Date

August 1st 2013

7.95 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Recycler.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
```

Include dependency graph for Recycler.cpp:



7.96 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Recycler.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Recycler](#)

Namespaces

- [MM](#)

7.96.1 Detailed Description

Author

Riemer van Rozen

Date

August 1st 2013

7.97 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Reflector.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Operator.h"
#include "Exp.h"
#include "Assertion.h"
#include "Deletion.h"
#include "Edge.h"
#include "StateEdge.h"
#include "FlowEdge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "Transformation.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "FlowEvent.h"
#include "TriggerEvent.h"
#include "Failure.h"
#include "Enablement.h"
#include "Disablement.h"
#include "Violation.h"
#include "Prevention.h"
#include "Activation.h"
#include "Program.h"
#include "PoolNodeBehavior.h"
#include "SourceNodeBehavior.h"
#include "DrainNodeBehavior.h"
#include "RefNodeBehavior.h"
#include "GateNodeBehavior.h"
#include "ConverterNodeBehavior.h"
#include "Declaration.h"
#include "InterfaceNode.h"
#include "Definition.h"
#include "Instance.h"
#include "ValExp.h"
#include "UnExp.h"
#include "BinExp.h"
#include "RangeValExp.h"
#include "BooleanValExp.h"
#include "NumberValExp.h"
#include "OverrideExp.h"
#include "ActiveExp.h"
#include "AllExp.h"
#include "DieExp.h"
#include "AliasExp.h"
#include "OneExp.h"
#include "VarExp.h"
#include "Reflector.h"
#include "Evaluator.h"
```

7.98 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Reflector.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Reflector](#)

Namespaces

- [MM](#)

7.98.1 Detailed Description

Author

Riemer van Rozen

Date

September 29th 2013

7.99 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/RefNodeBehavior.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "String.h"
#include "Name.h"
#include "Element.h"
#include "Exp.h"
#include "Edge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "PoolNodeBehavior.h"
#include "RefNodeBehavior.h"
#include "Declaration.h"
#include "Definition.h"
#include "Instance.h"
#include "InterfaceNode.h"
```

Include dependency graph for RefNodeBehavior.cpp:



7.100 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/RefNodeBehavior.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::RefNodeBehavior](#)

Namespaces

- [MM](#)

7.100.1 Detailed Description

Author

Riemer van Rozen

Date

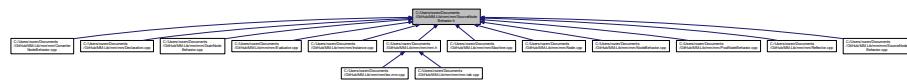
July 28th 2013

7.101 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/SourceNodeBehavior.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Transformation.h"
#include "Program.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "FlowEvent.h"
#include "TriggerEvent.h"
#include "Failure.h"
#include "Activation.h"
#include "Enablement.h"
#include "Disablement.h"
#include "Violation.h"
#include "Prevention.h"
#include "Operator.h"
#include "Exp.h"
#include "Assertion.h"
#include "Deletion.h"
#include "Edge.h"
#include "StateEdge.h"
#include "FlowEdge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "PoolNodeBehavior.h"
#include "SourceNodeBehavior.h"
#include "DrainNodeBehavior.h"
#include "GateNodeBehavior.h"
#include "RefNodeBehavior.h"
#include "ConverterNodeBehavior.h"
#include "Declaration.h"
#include "InterfaceNode.h"
#include "Definition.h"
#include "Instance.h"
#include "ValExp.h"
#include "UnExp.h"
#include "BinExp.h"
#include "DieExp.h"
#include "RangeValExp.h"
#include "BooleanValExp.h"
#include "NumberValExp.h"
#include "OverrideExp.h"
#include "ActiveExp.h"
#include "AllExp.h"
#include "AliasExp.h"
#include "OneExp.h"
#include "VarExp.h"
#include "Reflector.h"
```

7.102 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/SourceNodeBehavior.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class `MM::SourceNodeBehavior`

Namespaces

- MM

7.102.1 Detailed Description

Author

Riemer van Rozen

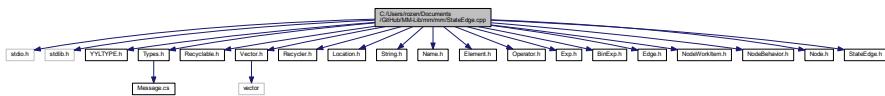
Date

July 11th 2013

7.103 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/StateEdge.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "String.h"
#include "Name.h"
#include "Element.h"
#include "Operator.h"
#include "Exp.h"
#include "BinExp.h"
#include "Edge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "StateEdge.h"
```

Include dependency graph for StateEdge.cpp:



7.104 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/StateEdge.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class MM::StateEdge

Namespaces

- MM

7.104.1 Detailed Description

Author

Riemer van Rozen

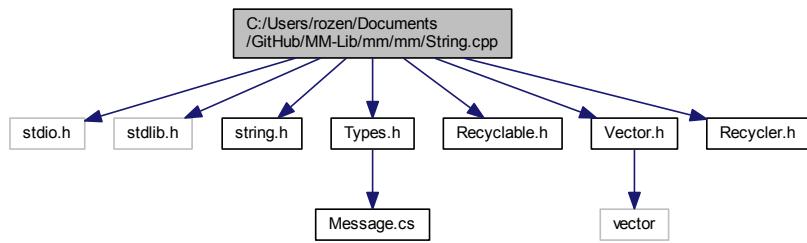
Date

July 11th 2013

7.105 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/String.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
```

Include dependency graph for String.cpp:



7.106 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/String.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::String](#)

Namespaces

- [MM](#)

7.106.1 Detailed Description

Author

Riemer van Rozen

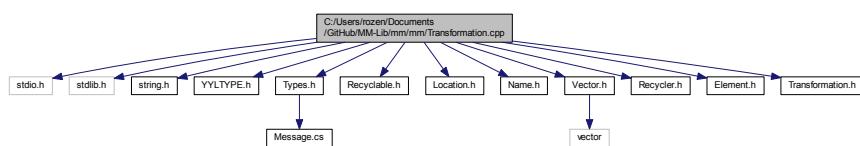
Date

October 7th 2013

7.107 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Transformation.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Location.h"
#include "Name.h"
#include "Vector.h"
#include "Recycler.h"
#include "Element.h"
#include "Transformation.h"
```

Include dependency graph for Transformation.cpp:



7.108 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Transformation.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Transformation](#)

Namespaces

- [MM](#)

7.108.1 Detailed Description

Author

Riemer van Rozen

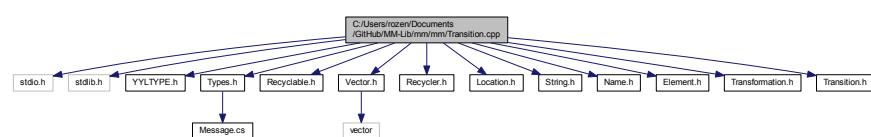
Date

October 16th 2013

7.109 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Transition.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "String.h"
#include "Name.h"
#include "Element.h"
#include "Transformation.h"
#include "Transition.h"
```

Include dependency graph for Transition.cpp:



7.110 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Transition.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Transition](#)

Namespaces

- [MM](#)

7.110.1 Detailed Description

Author

Riemer van Rozen

Date

October 7th 2013

7.111 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/TriggerEvent.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Transformation.h"
#include "Program.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "TriggerEvent.h"
#include "Operator.h"
#include "Exp.h"
#include "Edge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "Declaration.h"
#include "Definition.h"
#include "Instance.h"
```

Include dependency graph for TriggerEvent.cpp:



7.112 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/TriggerEvent.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::TriggerEvent](#)

Namespaces

- [MM](#)

7.112.1 Detailed Description

Author

Riemer van Rozen

Date

March 24th 2014

Author

Riemer van Rozen

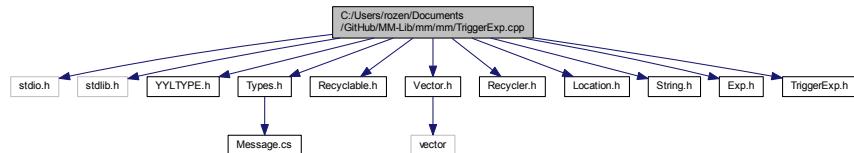
Date

March 26th 2014

7.113 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/TriggerExp.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "String.h"
#include "Exp.h"
#include "TriggerExp.h"
```

Include dependency graph for TriggerExp.cpp:



Namespaces

- MM

7.113.1 Detailed Description

Author

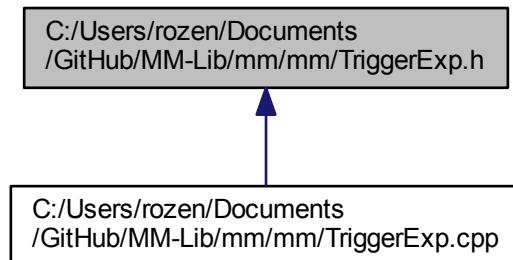
Riemer van Rozen

Date

July 20th 2013

7.114 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/TriggerExp.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::TriggerExp](#)

Namespaces

- [MM](#)

7.114.1 Detailed Description

Author

Riemer van Rozen

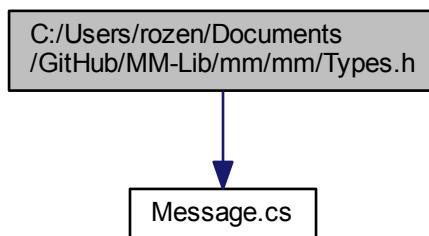
Date

July 20th 2013

7.115 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Types.h File Reference

```
#include "Message.cs"
```

Include dependency graph for Types.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- MM

Macros

- #define MM_Printf
- #define MM_MAX_RESOURCES LONG_MAX
- #define MM_TRUE true /*> true value */
- #define MM_FALSE false /*> false value */
- #define MM_NULL 0 /*> null value */

TypeDefs

- typedef unsigned char MM::UINT8
- typedef long MM::INT32
- typedef unsigned long MM::UINT32
- typedef char MM::CHAR
- typedef void MM::VOID
- typedef bool MM::BOOLEAN
- typedef MM::VOID(* MM::CALLBACK)(MM::UINT32 caller, MM::UINT32 message, MM::UINT32 instance, MM::UINT32 element, MM::UINT32 val)
- typedef enum MM::TID MM::TID

Enumerations

- enum MM::TID {
 MM::T_NULL, MM::T_Recycler, MM::T_Recyclable, MM::T_Observer,
 MM::T_Observable, MM::T_String, MM::T_Map, MM::T_Vector,
 MM::T_Reflector, MM::T_Machine, MM::T_Delegate, MM::T_InstanceObserver,
 MM::T_Program, MM::T_Evaluator, MM::T_Instance, MM::T_Transformation,
 MM::T_Transition, MM::T_Modification, MM::T_Event, MM::T_FlowEvent,
 MM::T_TriggerEvent, MM::T_Failure, MM::T_Activation, MM::T_Violation,
 MM::T_Prevention, MM::T_Enablement, MM::T_Disablement, MM::T_Location,
 MM::T_Name, MM::T_Element, MM::T_Definition, MM::T_Declaration,
 MM::T_Assertion, MM::T_Deletion, MM::T_Edge, MM::T_FlowEdge,
 MM::T_StateEdge, MM::T_Node, MM::T_InterfaceNode, MM::T_NodeBehavior,
 MM::T_PoolNodeBehavior, MM::T_DrainNodeBehavior, MM::T_GateNodeBehavior, MM::T_SourceNodeBehavior,
 MM::T_RefNodeBehavior, MM::T_ConverterNodeBehavior, MM::T_Exp, MM::T_TriggerExp,
 MM::T_AliasExp, MM::T_OneExp, MM::T_ActiveExp, MM::T_DieExp,
 MM::T_OverrideExp, MM::T_VarExp, MM::T_AllExp, MM::T_BinExp,
 MM::T_UnExp, MM::T_ValExp, MM::T_BooleanValExp, MM::T_NumberValExp,
 MM::T_RangeValExp }

7.115.1 Detailed Description

Author

Riemer van Rozen

Date

July 10th 2013

7.115.2 Macro Definition Documentation

7.115.2.1 #define MM_FALSE false /*> false value */

7.115.2.2 #define MM_MAX_RESOURCES LONG_MAX

boolean type

7.115.2.3 #define MM_NULL 0 /*> null value */

7.115.2.4 #define MM_printf

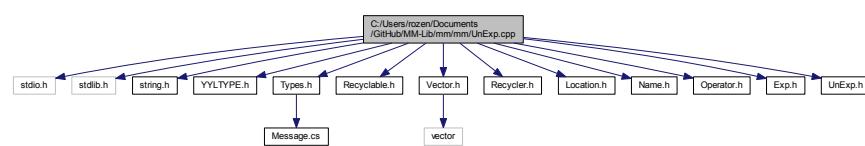
7.115.2.5 #define MM_TRUE true /*> true value */

7.116 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/UnExp.cpp File Reference

```
#include <stdio.h>
```

```
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "Name.h"
#include "Operator.h"
#include "Exp.h"
#include "UnExp.h"
```

Include dependency graph for UnExp.cpp:



Namespaces

- MM

7.116.1 Detailed Description

Author

Riemer van Rozen

Date

July 19th 2013

7.117 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/UnExp.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::UnExp](#)
Defines the `UnExp` class.

Namespaces

- MM

7.117.1 Detailed Description

Author

Riemer van Rozen

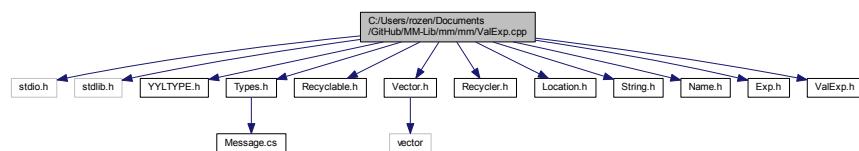
Date

July 19th 2013

7.118 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/ValExp.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "String.h"
#include "Name.h"
#include "Exp.h"
#include "ValExp.h"
```

Include dependency graph for ValExp.cpp:



Namespaces

- MM

7.118.1 Detailed Description

Author

Riemer van Rozen

Date

July 19th 2013

7.119 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/ValExp.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::ValExp](#)

Defines the [ValExp](#) class.

Namespaces

- [MM](#)

7.119.1 Detailed Description

Author

Riemer van Rozen

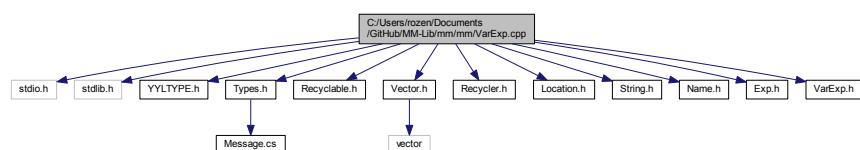
Date

July 19th 2013

7.120 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/VarExp.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Recycler.h"
#include "Location.h"
#include "String.h"
#include "Name.h"
#include "Exp.h"
#include "VarExp.h"
```

Include dependency graph for VarExp.cpp:



Namespaces

- [MM](#)

7.120.1 Detailed Description

Author

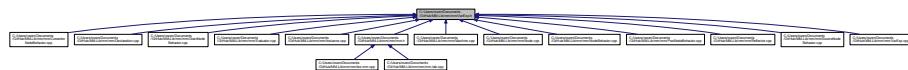
Riemer van Rozen

Date

July 19th 2013

7.121 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/VarExp.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::VarExp](#)
Defines the `VarExp` class.

Namespaces

- [MM](#)

7.121.1 Detailed Description

Author

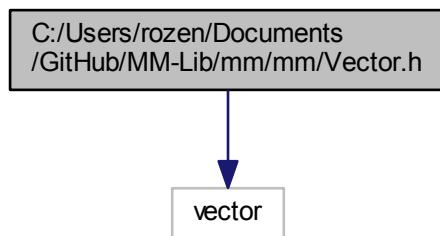
Riemer van Rozen

Date

July 19th 2013

7.122 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Vector.h File Reference

```
#include <vector>
Include dependency graph for Vector.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Vector< T >](#)
- class [MM::Vector< T >::Iterator](#)

Namespaces

- [MM](#)

7.123 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Violation.cpp File Reference

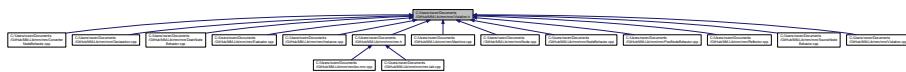
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "YYLTYPE.h"
#include "Types.h"
#include "Recyclable.h"
#include "Vector.h"
#include "Map.h"
#include "Recycler.h"
#include "Observer.h"
#include "Observable.h"
#include "Location.h"
#include "Name.h"
#include "Element.h"
#include "Transformation.h"
#include "Program.h"
#include "Modification.h"
#include "Transition.h"
#include "Event.h"
#include "Violation.h"
#include "Operator.h"
#include "Exp.h"
#include "Assertion.h"
#include "Edge.h"
#include "NodeWorkItem.h"
#include "NodeBehavior.h"
#include "Node.h"
#include "Declaration.h"
#include "Definition.h"
#include "Instance.h"
```

Include dependency graph for Violation.cpp:



7.124 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Violation.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class [MM::Violation](#)

Namespaces

- [MM](#)

7.124.1 Detailed Description

Author

Riemer van Rozen

Date

October 16th 2013

7.125 C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/YYLTYPE.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct [YYLTYPE](#)

Macros

- #define [YYLTYPE_IS_DECLARED](#) 1

TypeDefs

- typedef struct [YYLTYPE](#) [YYLTYPE](#)

7.125.1 Detailed Description

Author

Riemer van Rozen

Date

July 29th 2013

7.125.2 Macro Definition Documentation

7.125.2.1 #define YYLTYPE_IS_DECLARED 1

7.125.3 Typedef Documentation

7.125.3.1 typedef struct YYLTYPE YYLTYPE

Index

- ~Activation
 - MM::Activation, 27
- ~ActiveExp
 - MM::ActiveExp, 30
- ~AliasExp
 - MM::AliasExp, 34
- ~AllExp
 - MM::AllExp, 37
- ~Assertion
 - MM::Assertion, 41
- ~BinExp
 - MM::BinExp, 44
- ~BooleanValExp
 - MM::BooleanValExp, 48
- ~ConverterNodeBehavior
 - MM::ConverterNodeBehavior, 53
- ~Declaration
 - MM::Declaration, 58
- ~Definition
 - MM::Definition, 65
- ~Delegate
 - MM::Machine::Delegate, 75
- ~Deletion
 - MM::Deletion, 78
- ~DieExp
 - MM::DieExp, 81
- ~Disablement
 - MM::Disablement, 86
- ~DrainNodeBehavior
 - MM::DrainNodeBehavior, 89
- ~Edge
 - MM::Edge, 94
- ~Element
 - MM::Element, 100
- ~Enablement
 - MM::Enablement, 109
- ~Evaluator
 - MM::Evaluator, 112
- ~Event
 - MM::Event, 123
- ~Exp
 - MM::Exp, 128
- ~Failure
 - MM::Failure, 134
- ~FlowEdge
 - MM::FlowEdge, 137
- ~FlowEvent
 - MM::FlowEvent, 141
- ~GateNodeBehavior
 - MM::GateNodeBehavior, 145
- ~Instance
 - MM::Instance, 151
- ~InstanceObserver
 - MM::Machine::InstanceObserver, 173
- ~InterfaceNode
 - MM::InterfaceNode, 177
- ~Iterator
 - MM::Map::Iterator, 179
 - MM::Vector::Iterator, 181
- ~Location
 - MM::Location, 186
- ~Machine
 - MM::Machine, 192
- ~Map
 - MM::Map, 228
- ~Modification
 - MM::Modification, 231
- ~Name
 - MM::Name, 234
- ~Node
 - MM::Node, 245
- ~NodeBehavior
 - MM::NodeBehavior, 260
- ~NodeWorkItem
 - MM::NodeWorkItem, 271
- ~NumberValExp
 - MM::NumberValExp, 276
- ~Observable
 - MM::Observable, 279
- ~Observer
 - MM::Observer, 282
- ~OneExp
 - MM::OneExp, 285
- ~OverrideExp
 - MM::OverrideExp, 291
- ~PoolNodeBehavior
 - MM::PoolNodeBehavior, 295
- ~Prevention
 - MM::Prevention, 309
- ~Program
 - MM::Program, 311
- ~RangeValExp
 - MM::RangeValExp, 315
- ~Recyclable
 - MM::Recyclable, 319
- ~Recycler
 - MM::Recycler, 324
- ~RefNodeBehavior
 - MM::RefNodeBehavior, 325

MM::RefNodeBehavior, 335
~Reflector
 MM::Reflector, 329
~SourceNodeBehavior
 MM::SourceNodeBehavior, 341
~StateEdge
 MM::StateEdge, 346
~String
 MM::String, 350
~Transformation
 MM::Transformation, 359
~Transition
 MM::Transition, 365
~TriggerEvent
 MM::TriggerEvent, 369
~TriggerExp
 MM::TriggerExp, 372
~UnExp
 MM::UnExp, 376
~ValExp
 MM::ValExp, 380
~VarExp
 MM::VarExp, 386
~Vector
 MM::Vector, 388
~Violation
 MM::Violation, 395
_Act
 MM::NodeBehavior, 259
_How
 MM::NodeBehavior, 259
_IO
 MM::NodeBehavior, 259
_OP
 MM::Operator, 287
_When
 MM::NodeBehavior, 259

ACT_ERROR
 MM::NodeBehavior, 259
ACT_PULL
 MM::NodeBehavior, 259
ACT_PUSH
 MM::NodeBehavior, 259
ACTIVATE
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489
ACTIVE
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489
ADD
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490
ADDITION
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490
ALIAS
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489

ALL
 mm.tab.cpp, 475, 476
 mm.tab.hpp, 487, 489
AND
 mm.tab.cpp, 475, 476
 mm.tab.hpp, 487, 489
ANY
 mm.tab.cpp, 475, 476
 mm.tab.hpp, 487, 489
ASSERT
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489
AT
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489
AUTO
 mm.tab.cpp, 475, 476
 mm.tab.hpp, 487, 489
ACT_LEN
 MM::NodeBehavior, 269
ACT_STR
 MM::NodeBehavior, 269
Act
 MM::NodeBehavior, 259
act
 YYSTYPE, 399
activate
 MM::Machine, 192
activateTriggerTargets
 MM::ConverterNodeBehavior, 53
 MM::InterfaceNode, 177
 MM::Node, 245
 MM::NodeBehavior, 260
Activation, 28
 MM::Activation, 26, 27
ActiveExp, 32
 MM::ActiveExp, 30
add
 MM::ConverterNodeBehavior, 53
 MM::DrainNodeBehavior, 89
 MM::GateNodeBehavior, 145
 MM::Instance, 151
 MM::InterfaceNode, 177
 MM::Node, 245
 MM::NodeBehavior, 260
 MM::PoolNodeBehavior, 295
 MM::RefNodeBehavior, 335
 MM::SourceNodeBehavior, 341
 MM::Vector, 388
addAlias
 MM::Node, 246
addAll
 MM::Vector, 389
addCondition
 MM::Node, 246
addDefinitionObserver
 MM::Machine, 192
addElement

MM::Definition, 65
 MM::Reflector, 329
 MM::Transformation, 359
addInput
 MM::Node, 246
addInstanceObserver
 MM::Machine, 193
addInterface
 MM::Declaration, 58
 MM::PoolNodeBehavior, 296
addObserver
 MM::Observable, 279
addOutput
 MM::Node, 246
addPullAllNode
 MM::Definition, 65
addPullAnyNode
 MM::Definition, 66
addPushAllNode
 MM::Definition, 66
addPushAnyNode
 MM::Definition, 66
addTransformation
 MM::Program, 311
addTrigger
 MM::Node, 246
AliasExp, 32
 MM::AliasExp, 34
AllExp, 39
 MM::AllExp, 37
append
 MM::Name, 235
 MM::String, 350, 351
appendInt
 MM::String, 351
Assertion, 39
 MM::Assertion, 41
at
 MM::Vector, 389
BEGIN
 lex.mm.cpp, 442
BOOLEAN
 MM, 21
begin
 MM::ConverterNodeBehavior, 53
 MM::Declaration, 59
 MM::DrainNodeBehavior, 89
 MM::Element, 101
 MM::GateNodeBehavior, 145
 MM::Instance, 151
 MM::Node, 246
 MM::NodeBehavior, 260
 MM::PoolNodeBehavior, 297
 MM::RefNodeBehavior, 335
 MM::SourceNodeBehavior, 341
BinExp, 42
 MM::BinExp, 44
BooleanValExp, 47
COLON
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490
CONVERTER
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- Activation.cpp, 401
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- Activation.h, 402
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- ActiveExp.cpp, 402
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- ActiveExp.h, 402
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- AliasExp.cpp, 403
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- AliasExp.h, 404
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/All- Exp.cpp, 405
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/All- Exp.h, 405
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- Assertion.cpp, 406
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- Assertion.h, 406
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- BinExp.cpp, 407
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- BinExp.h, 407
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- BooleanValExp.cpp, 408
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- BooleanValExp.h, 409
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- ConverterNodeBehavior.cpp, 411
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- ConverterNodeBehavior.h, 412
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- Declaration.cpp, 414
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- Declaration.h, 415
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- Definition.cpp, 415
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- Definition.h, 416
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- Deletion.cpp, 416
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- Deletion.h, 417
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- DieExp.cpp, 417
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- DieExp.h, 418
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/- Disablement.cpp, 419

C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Disablement.h, 419
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
DrainNodeBehavior.cpp, 421
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
DrainNodeBehavior.h, 422
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Edge.cpp, 422
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Edge.h, 423
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Element.cpp, 423
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Element.h, 424
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Enablement.cpp, 425
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Enablement.h, 425
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Evaluator.cpp, 427
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Evaluator.h, 428
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Event.cpp, 428
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Event.h, 429
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Exp.cpp, 429
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Exp.h, 430
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Failure.cpp, 431
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Failure.h, 431
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
FlowEdge.cpp, 432
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
FlowEdge.h, 432
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
FlowEvent.cpp, 433
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
FlowEvent.h, 433
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
GateNodeBehavior.cpp, 434
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
GateNodeBehavior.h, 434
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Instance.cpp, 436
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Instance.h, 437
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
InterfaceNode.cpp, 438
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
InterfaceNode.h, 438
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Location.cpp, 456
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Location.h, 457
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Machine.cpp, 459
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Machine.h, 464
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Map.h, 464
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Message.cs, 465
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Modification.cpp, 490
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Modification.h, 491
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Name.cpp, 491
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Name.h, 492
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Node.cpp, 494
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Node.h, 495
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
NodeBehavior.cpp, 496
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
NodeBehavior.h, 497
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
NodeWorkItem.cpp, 497
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
NodeWorkItem.h, 498
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
NumberValExp.cpp, 498
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
NumberValExp.h, 499
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Observable.cpp, 499
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Observable.h, 500
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Observer.cpp, 500
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Observer.h, 501
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
OneExp.cpp, 502
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
OneExp.h, 502
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Operator.cpp, 503
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Operator.h, 503
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
OverrideExp.cpp, 504
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
OverrideExp.h, 505
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
PoolNodeBehavior.cpp, 506
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
PoolNodeBehavior.h, 507
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Prevention.cpp, 508

C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Prevention.h, 508
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Program.cpp, 509
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Program.h, 509
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
RangeValExp.cpp, 510
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
RangeValExp.h, 510
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Recyclable.cpp, 511
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Recyclable.h, 511
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Recycler.cpp, 512
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Recycler.h, 512
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
RefNodeBehavior.cpp, 516
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
RefNodeBehavior.h, 516
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Reflector.cpp, 514
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Reflector.h, 515
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
SourceNodeBehavior.cpp, 518
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
SourceNodeBehavior.h, 519
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
StateEdge.cpp, 519
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
StateEdge.h, 520
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
String.cpp, 520
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
String.h, 521
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Transformation.cpp, 522
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Transformation.h, 522
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Transition.cpp, 523
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Transition.h, 523
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
TriggerEvent.cpp, 524
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
TriggerEvent.h, 524
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
TriggerExp.cpp, 525
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
TriggerExp.h, 526
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Types.h, 527
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
UnExp.cpp, 528
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
UnExp.h, 529
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
ValExp.cpp, 530
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
ValExp.h, 530
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
VarExp.cpp, 531
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
VarExp.h, 532
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Vector.h, 532
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Violation.cpp, 533
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/-
Violation.h, 534
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/Y-
LTYPE.h, 534
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/lex.-
mm.cpp, 439
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm.-
h, 467
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/mm.-
tab.cpp, 468
C:/Users/rozen/Documents/GitHub/MM-Lib/mm/mm/mm.-
tab.hpp, 485
CALLBACK
MM, 21
CHAR
MM, 21
COMMA_CHAR
MM::Definition, 73
change
MM::ConverterNodeBehavior, 53
MM::Declaration, 60
MM::DrainNodeBehavior, 89
MM::Element, 101
MM::GateNodeBehavior, 145
MM::Node, 246
MM::NodeBehavior, 260
MM::PoolNodeBehavior, 298
MM::RefNodeBehavior, 336
MM::SourceNodeBehavior, 341
clear
MM::Map, 228
MM::String, 352
MM::Vector, 389
clearActive
MM::Instance, 152
clearDisabled
MM::Instance, 152
clearElements
MM::Definition, 66
MM::Transformation, 360
conformsTo
MM::NodeBehavior, 261
contains
MM::Map, 228

MM::Vector, 389
containsElement
 MM::Definition, 66
ConverterNodeBehavior, 56
 MM::ConverterNodeBehavior, 53
create
 MM::Recycler, 324
createActivation
 MM::Machine, 193
createActiveExp
 MM::Machine, 193, 194
createAliasExp
 MM::Machine, 194
createAllExp
 MM::Machine, 194
createAnonymousTriggerEdge
 MM::Machine, 195
createAssertion
 MM::Machine, 195
createBinExp
 MM::Machine, 196
createBooleanValExp
 MM::Machine, 196, 197
createBuffer
 MM::Recycler, 325
createConverterNode
 MM::Machine, 197
createDeclaration
 MM::Machine, 198
createDefinition
 MM::Machine, 198, 199
createDeletion
 MM::Machine, 199
createDisablement
 MM::Machine, 200
createDrainNode
 MM::Machine, 200
createEdgeVector
 MM::Machine, 201
createElementVector
 MM::Machine, 201
createEnablement
 MM::Machine, 201, 202
createFailure
 MM::Machine, 202
createFlowEdge
 MM::Machine, 203
createFlowEvent
 MM::Machine, 203
createGateNode
 MM::Machine, 204
createInstance
 MM::Machine, 204
createInstances
 MM::Instance, 152
createInterfaceNode
 MM::Machine, 205
createLocation
 MM::Machine, 205
createModification
 MM::Machine, 206, 207
createName
 MM::Machine, 207, 208
createName2ElementMap
 MM::Machine, 209
createName2NodeMap
 MM::Machine, 209
createNodeVector
 MM::Machine, 209
createNumberValExp
 MM::Machine, 209
createOneExp
 MM::Machine, 210
createOverrideExp
 MM::Machine, 210, 211
createPoolNode
 MM::Machine, 211
createPrevention
 MM::Machine, 212
createProgram
 MM::Machine, 212, 213
createRangeValExp
 MM::Machine, 213, 214
createRefNode
 MM::Machine, 214
createSourceNode
 MM::Machine, 214
createStateEdge
 MM::Machine, 215
createString
 MM::Machine, 215
createTransformationVector
 MM::Machine, 216
createTransition
 MM::Machine, 216, 217
createTriggerEvent
 MM::Machine, 217, 218
createUnExp
 MM::Machine, 218
createVarExp
 MM::Machine, 219
createViolation
 MM::Machine, 219, 220

DELETE
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489
DICE
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489
DISABLE
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489
DIV
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490
DOT

mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489
DOT_GT
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489
DRAIN
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490
Declaration, 63
 MM::Declaration, 58
Definition, 73
 MM::Definition, 65
deinit
 MM::Reflector, 329
Delegate
 MM::Machine::Delegate, 75
delegates
 MM::Machine, 227
deleteValue
 MM::Instance, 153
Deletion, 76
 MM::Deletion, 78
deprioritize
 MM::Definition, 66
destroyAllInstances
 MM::Instance, 153
destroyInstance
 MM::Instance, 154
destroyInstances
 MM::Instance, 154
DieExp, 79
 MM::DieExp, 81
Disablement, 83
 MM::Disablement, 85, 86
doTriggers
 MM::RefNodeBehavior, 336
DrainNodeBehavior, 87
 MM::DrainNodeBehavior, 89
ENABLE
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489
EQ
 mm.tab.cpp, 475, 476
 mm.tab.hpp, 487, 489
ECHO
 lex.mm.cpp, 442
eList
 YYSTYPE, 400
EOB_ACT_LAST_MATCH
 lex.mm.cpp, 442
ERROR
 lex.mm.cpp, 442
eatWhiteSpace
 MM::Machine, 220
Edge, 93
 MM::Edge, 94
Element, 107
 MM::Element, 100
element
 MM::Event, 126
 YYSTYPE, 399
elementAt
 MM::Vector, 389
Enablement, 110
 MM::Enablement, 109
end
 MM::ConverterNodeBehavior, 53
 MM::Declaration, 60
 MM::DrainNodeBehavior, 89
 MM::Element, 101
 MM::GateNodeBehavior, 146
 MM::Node, 246
 MM::NodeBehavior, 261
 MM::PoolNodeBehavior, 298
 MM::RefNodeBehavior, 336
 MM::SourceNodeBehavior, 341
equals
 MM::Name, 235, 236
eval
 MM::Evaluator, 112–120
 MM::Machine, 220
evalFile
 MM::Machine, 221
Evaluator, 121
 MM::Evaluator, 112
Event, 127
 MM::Event, 123
Exp, 132
 MM::Exp, 128
exp
 YYSTYPE, 400
FAIL
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489
FALSE
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489
FPVAL
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490
FROM
 mm.tab.cpp, 475, 476
 mm.tab.hpp, 487, 489
FLEX_BETA
 lex.mm.cpp, 442
FLEX_SCANNER
 lex.mm.cpp, 442
FLEXINT_H
 lex.mm.cpp, 442
FailEvent, 132
Failure
 MM::Failure, 134
finalize
 MM::Instance, 155
findDeclaredDefinition
 MM::Definition, 66

findNode
 MM::Definition, 67

findQueriedDefinition
 MM::Definition, 67

first_column
 YYLTYPE, 398

first_line
 YYLTYPE, 398

flex_int16_t
 lex.mm.cpp, 446

flex_int32_t
 lex.mm.cpp, 446

flex_int8_t
 lex.mm.cpp, 446

flex_uint16_t
 lex.mm.cpp, 446

flex_uint32_t
 lex.mm.cpp, 446

flex_uint8_t
 lex.mm.cpp, 446

FlowEdge, 138
 MM::FlowEdge, 137

FlowEvent, 138
 MM::FlowEvent, 141

for
 mm.tab.cpp, 478

GATE
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490

GE
 mm.tab.cpp, 475, 476
 mm.tab.hpp, 487, 489

GT
 mm.tab.cpp, 475, 476
 mm.tab.hpp, 487, 489

GateNodeBehavior, 143
 MM::GateNodeBehavior, 145

get
 MM::Map, 228

getAct
 MM::NodeBehavior, 261

getActEdge
 MM::FlowEvent, 141

getActInstance
 MM::FlowEvent, 141

getActNode
 MM::FlowEvent, 141

getAdd
 MM::PoolNodeBehavior, 299

getAlias
 MM::RefNodeBehavior, 336

getAliases
 MM::Node, 246

getAmount
 MM::FlowEvent, 141
 MM::Node, 247
 MM::NodeBehavior, 261
 MM::PoolNodeBehavior, 299

getAt
 MM::PoolNodeBehavior, 299

getBehavior
 MM::InterfaceNode, 177
 MM::Node, 247

getBuffer
 MM::Name, 236
 MM::String, 352

getCapacity
 MM::ConverterNodeBehavior, 53
 MM::DrainNodeBehavior, 89
 MM::GateNodeBehavior, 146
 MM::Instance, 155
 MM::InterfaceNode, 177
 MM::Node, 248
 MM::NodeBehavior, 261
 MM::PoolNodeBehavior, 299
 MM::RefNodeBehavior, 336
 MM::SourceNodeBehavior, 341

getColumn
 MM::Location, 186

getConditions
 MM::Node, 248

getCreateMessage
 MM::ConverterNodeBehavior, 53
 MM::DrainNodeBehavior, 89
 MM::GateNodeBehavior, 146
 MM::NodeBehavior, 261
 MM::PoolNodeBehavior, 300
 MM::RefNodeBehavior, 336
 MM::SourceNodeBehavior, 341

getDeclaration
 MM::Instance, 155
 MM::InterfaceNode, 177

getDefinition
 MM::Declaration, 60
 MM::Instance, 156
 MM::Machine, 223
 MM::PoolNodeBehavior, 300
 MM::Reflector, 329

getDeleteMessage
 MM::ConverterNodeBehavior, 53
 MM::DrainNodeBehavior, 90
 MM::GateNodeBehavior, 146
 MM::NodeBehavior, 262
 MM::PoolNodeBehavior, 300
 MM::RefNodeBehavior, 336
 MM::SourceNodeBehavior, 341

getDrainNode
 MM::ConverterNodeBehavior, 54

getEdge
 MM::NodeWorkItem, 271

getElement
 MM::Definition, 67
 MM::Event, 124

getElements
 MM::Definition, 68
 MM::Transformation, 360

getEvaluatedExp
 MM::Instance, 156

getEvaluator
 MM::Machine, 223

getExp
 MM::Assertion, 41
 MM::Edge, 95
 MM::OverrideExp, 291
 MM::UnExp, 377

getFrom
 MM::ConverterNodeBehavior, 54

getGateValue
 MM::Instance, 157

getHow
 MM::NodeBehavior, 262

getIO
 MM::NodeBehavior, 262

getIndex
 MM::Instance, 157

getInput
 MM::Node, 249

getInstance
 MM::Event, 124
 MM::Instance, 158
 MM::Machine, 223
 MM::NodeWorkItem, 272
 MM::Reflector, 329

getInstanceName
 MM::Machine, 224

getInstances
 MM::Instance, 159

getIntValue
 MM::NumberValExp, 276
 MM::RangeValExp, 315

getInterface
 MM::Declaration, 61
 MM::PoolNodeBehavior, 300

getIterator
 MM::Map, 228
 MM::Vector, 390

getLength
 MM::Location, 186
 MM::Name, 236

getLhsExp
 MM::BinExp, 44

getLine
 MM::Location, 186

getLocation
 MM::Activation, 27
 MM::Assertion, 41
 MM::BooleanValExp, 49
 MM::Deletion, 78
 MM::Disablement, 86
 MM::Enablement, 109
 MM::Failure, 134
 MM::Name, 237
 MM::NumberValExp, 276
 MM::Prevention, 309

getLog
 MM::Machine, 224

getMax
 MM::DieExp, 81
 MM::PoolNodeBehavior, 300
 MM::RangeValExp, 315

getMessage
 MM::Activation, 27
 MM::Assertion, 41
 MM::Disablement, 86
 MM::Enablement, 109
 MM::Event, 124
 MM::Failure, 134
 MM::FlowEvent, 141
 MM::Prevention, 309
 MM::TriggerEvent, 369
 MM::Violation, 395

getMessageLength
 MM::Assertion, 41

getMin
 MM::RangeValExp, 315

getName
 MM::ActiveExp, 30
 MM::Disablement, 86
 MM::Element, 101
 MM::Enablement, 109
 MM::Machine, 224
 MM::Name, 237
 MM::VarExp, 386

getNewIterator
 MM::Vector, 391

getNewValue
 MM::Instance, 159

getNext
 MM::Map::Iterator, 179, 180
 MM::Vector::Iterator, 181

getNode
 MM::InterfaceNode, 177
 MM::NodeWorkItem, 272

getOf
 MM::PoolNodeBehavior, 300

getOldValue
 MM::Instance, 159

getOperator
 MM::BinExp, 44
 MM::UnExp, 377

getOutput
 MM::Node, 249

getParent
 MM::Instance, 160

getPosition
 MM::Vector, 391

getPreName
 MM::Name, 237

getPullAllNodes
 MM::Definition, 68

getPullAnyNodes
 MM::Definition, 68
getPushAllNodes
 MM::Definition, 68
getPushAnyNodes
 MM::Definition, 68
getReference
 MM::RefNodeBehavior, 336
getReflector
 MM::Machine, 225
getResources
 MM::ConverterNodeBehavior, 54
 MM::DrainNodeBehavior, 90
 MM::GateNodeBehavior, 146
 MM::Instance, 160
 MM::InterfaceNode, 177
 MM::Node, 250
 MM::NodeBehavior, 262
 MM::PoolNodeBehavior, 300
 MM::RefNodeBehavior, 337
 MM::SourceNodeBehavior, 342
getRhsExp
 MM::BinExp, 44
getSize
 MM::String, 353
getSource
 MM::Edge, 95
getSourceInstance
 MM::FlowEvent, 141
getSourceName
 MM::Edge, 95
getSourceNode
 MM::ConverterNodeBehavior, 54
 MM::FlowEvent, 141
getTarget
 MM::Edge, 95
getTargetInstance
 MM::FlowEvent, 142
getTargetName
 MM::Edge, 96
getTargetNode
 MM::FlowEvent, 142
getTo
 MM::ConverterNodeBehavior, 54
getTransformations
 MM::Program, 311
getTriggerEdge
 MM::ConverterNodeBehavior, 54
getTriggers
 MM::Node, 250
getTypeId
 MM::Activation, 27
 MM::ActiveExp, 30
 MM::AliasExp, 34
 MM::AllExp, 37
 MM::Assertion, 41
 MM::BinExp, 45
 MM::BooleanValExp, 49
 MM::ConverterNodeBehavior, 54
 MM::Declaration, 61
 MM::Definition, 68
 MM::Deletion, 78
 MM::DieExp, 82
 MM::Disablement, 86
 MM::DrainNodeBehavior, 90
 MM::Edge, 96
 MM::Element, 102
 MM::Enablement, 109
 MM::Evaluator, 120
 MM::Event, 124
 MM::Exp, 129
 MM::Failure, 134
 MM::FlowEdge, 137
 MM::FlowEvent, 142
 MM::GateNodeBehavior, 146
 MM::Instance, 160
 MM::InterfaceNode, 177
 MM::Location, 186
 MM::Machine, 225
 MM::Machine::Delegate, 75
 MM::Machine::InstanceObserver, 174
 MM::Map, 229
 MM::Modification, 231
 MM::Name, 238
 MM::Node, 251
 MM::NodeBehavior, 263
 MM::NumberValExp, 276
 MM::Observable, 279
 MM::Observer, 282
 MM::OneExp, 285
 MM::OverrideExp, 291
 MM::PoolNodeBehavior, 301
 MM::Prevention, 309
 MM::Program, 311
 MM::RangeValExp, 316
 MM::Recyclable, 319
 MM::Recycler, 326
 MM::Reflector, 330
 MM::RefNodeBehavior, 337
 MM::SourceNodeBehavior, 342
 MM::StateEdge, 347
 MM::String, 354
 MM::Transformation, 361
 MM::Transition, 365
 MM::TriggerEvent, 369
 MM::TriggerExp, 372
 MM::UnExp, 377
 MM::ValExp, 380
 MM::VarExp, 386
 MM::Vector, 391
 MM::Violation, 395
 MM::TypeName
 MM::Declaration, 61
 MM::PoolNodeBehavior, 301
 getUpdateMessage
 MM::ConverterNodeBehavior, 54

MM::DrainNodeBehavior, 90
 MM::GateNodeBehavior, 146
 MM::NodeBehavior, 263
 MM::PoolNodeBehavior, 301
 MM::RefNodeBehavior, 337
 MM::SourceNodeBehavior, 342
getUsed
 MM::String, 354
getValue
 MM::BooleanValExp, 49
 MM::Instance, 161
 MM::NumberValExp, 276
getWhen
 MM::NodeBehavior, 263
getWork
 MM::NodeBehavior, 263
greaterEquals
 MM::BooleanValExp, 49
 MM::NumberValExp, 276
 MM::OneExp, 285
 MM::RangeValExp, 316
 MM::ValExp, 381
HOW_ALL
 MM::NodeBehavior, 259
HOW_ANY
 MM::NodeBehavior, 259
HOW_ERROR
 MM::NodeBehavior, 259
HOW_LEN
 MM::NodeBehavior, 269
HOW_STR
 MM::NodeBehavior, 270
hasCapacity
 MM::ConverterNodeBehavior, 54
 MM::DrainNodeBehavior, 90
 MM::GateNodeBehavior, 146
 MM::Instance, 161
 MM::InterfaceNode, 177
 MM::Node, 251
 MM::NodeBehavior, 264
 MM::PoolNodeBehavior, 301
 MM::RefNodeBehavior, 337
 MM::SourceNodeBehavior, 342
hasEdgeOwnership
 MM::Node, 251
hasNext
 MM::Map::Iterator, 180
 MM::Vector::Iterator, 182
hasResources
 MM::ConverterNodeBehavior, 54
 MM::DrainNodeBehavior, 90
 MM::GateNodeBehavior, 146
 MM::Instance, 162
 MM::InterfaceNode, 177
 MM::Node, 251
 MM::NodeBehavior, 264
 MM::PoolNodeBehavior, 301
 MM::RefNodeBehavior, 337
How
 MM::NodeBehavior, 259
how
 YYSTYPE, 400
ID
 mm.tab.cpp, 474, 476
 mm.tab.hpp, 487, 488
IN
 mm.tab.cpp, 475, 476
 mm.tab.hpp, 487, 488
INOUT
 mm.tab.cpp, 475, 476
 mm.tab.hpp, 487, 489
IO_ERROR
 MM::NodeBehavior, 259
IO_IN
 MM::NodeBehavior, 259
IO_INOUT
 MM::NodeBehavior, 259
IO_OUT
 MM::NodeBehavior, 259
IO_PRIVATE
 MM::NodeBehavior, 259
IDPART
 lex.mm.cpp, 442
IDPART2
 lex.mm.cpp, 442
IN_M_COMMENT
 lex.mm.cpp, 442
IN_S_COMMENT
 lex.mm.cpp, 442
INITIAL
 lex.mm.cpp, 442
INT16_MAX
 lex.mm.cpp, 442
INT16_MIN
 lex.mm.cpp, 442
INT32
 MM, 21
INT32_MAX
 lex.mm.cpp, 442
INT32_MIN
 lex.mm.cpp, 442
INT8_MAX
 lex.mm.cpp, 442
INT8_MIN
 lex.mm.cpp, 443
IO
 MM::NodeBehavior, 259
IO_LEN
 MM::NodeBehavior, 270
IO_STR
 MM::NodeBehavior, 270
if
 lex.mm.cpp, 447
 mm.tab.cpp, 478
inID

lex.mm.cpp, 455
init
 MM::Reflector, 330, 331
Instance, 148
 MM::Instance, 150
instance
 MM::Event, 126
InstanceObserver
 MM::Machine::InstanceObserver, 173
instanceof
 MM::Activation, 27
 MM::ActiveExp, 30
 MM::AliasExp, 34
 MM::AllExp, 37
 MM::Assertion, 41
 MM::BinExp, 45
 MM::BooleanValExp, 49
 MM::ConverterNodeBehavior, 54
 MM::Declaration, 61
 MM::Definition, 68
 MM::Deletion, 78
 MM::DieExp, 82
 MM::Disablement, 86
 MM::DrainNodeBehavior, 90
 MM::Edge, 96
 MM::Element, 103
 MM::Enablement, 109
 MM::Evaluator, 120
 MM::Event, 124
 MM::Exp, 129
 MM::Failure, 134
 MM::FlowEdge, 137
 MM::FlowEvent, 142
 MM::GateNodeBehavior, 146
 MM::Instance, 162
 MM::InterfaceNode, 177
 MM::Location, 186
 MM::Machine, 225
 MM::Machine::Delegate, 75
 MM::Machine::InstanceObserver, 174
 MM::Map, 229
 MM::Modification, 231
 MM::Name, 238
 MM::Node, 251
 MM::NodeBehavior, 264
 MM::NumberValExp, 277
 MM::Observable, 279
 MM::Observer, 282
 MM::OneExp, 286
 MM::OverrideExp, 291
 MM::PoolNodeBehavior, 302
 MM::Prevention, 309
 MM::Program, 311
 MM::RangeValExp, 316
 MM::Recyclable, 319
 MM::Recycler, 326
 MM::Reflector, 332
 MM::RefNodeBehavior, 337
 MM::SourceNodeBehavior, 342
 MM::StateEdge, 347
 MM::String, 355
 MM::Transformation, 361
 MM::Transition, 365
 MM::TriggerEvent, 369
 MM::TriggerExp, 372
 MM::UnExp, 377
 MM::ValExp, 381
 MM::VarExp, 386
 MM::Vector, 392
 MM::Violation, 395
InterfaceNode, 174
 MM::InterfaceNode, 176
io
 YYSTYPE, 400
isActive
 MM::Instance, 163
isAlaaliasExp
 MM::BinExp, 45
isAlias
 MM::StateEdge, 347
isCondition
 MM::StateEdge, 347
isDisabled
 MM::Instance, 163
 MM::Node, 252
isEmpty
 MM::Map, 229
 MM::Vector, 392
isEvaluatedExp
 MM::Instance, 163
isMarked
 MM::Instance, 164
isSatisfied
 MM::Node, 252
isTrigger
 MM::StateEdge, 347
isTriggerExp
 MM::BinExp, 45
isVisible
 MM::Element, 104
isatty
 lex.mm.cpp, 447
Iterator
 MM::Map::Iterator, 179
 MM::Vector::Iterator, 181
LCURLY
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490
LE
 mm.tab.cpp, 475, 476
 mm.tab.hpp, 487, 489
LPAREN
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490
LT
 mm.tab.cpp, 475, 476

mm.tab.hpp, 487, 489
 LBRACE_CHAR
 MM::Definition, 73
 LEX_ID_SIZE
 lex.mm.cpp, 443
 LOG_SIZE
 MM::Machine, 227
 last_column
 YYLTYPE, 399
 last_line
 YYLTYPE, 399
 lex.mm.cpp
 BEGIN, 442
 ECHO, 442
 ERROR, 442
 FLEX_BETA, 442
 FLEX_SCANNER, 442
 FLEXINT_H, 442
 flex_int16_t, 446
 flex_int32_t, 446
 flex_int8_t, 446
 flex_uint16_t, 446
 flex_uint32_t, 446
 flex_uint8_t, 446
 IDPART, 442
 IDPART2, 442
 IN_M_COMMENT, 442
 IN_S_COMMENT, 442
 INITIAL, 442
 INT16_MAX, 442
 INT16_MIN, 442
 INT32_MAX, 442
 INT32_MIN, 442
 INT8_MAX, 442
 INT8_MIN, 443
 if, 447
 inID, 455
 isatty, 447
 LEX_ID_SIZE, 443
 REJECT, 443
 UINT16_MAX, 443
 UINT32_MAX, 443
 UINT8_MAX, 443
 unput, 443
 while, 447
 YY_AT_BOL, 443
 YY_BREAK, 443
 YY_BUF_SIZE, 443
 YY_BUFFER_NEW, 443
 YY_BUFFER_NORMAL, 443
 YY_BUFFER_STATE, 446
 YY_CHAR, 447
 YY_CURRENT_BUFFER, 443
 YY_DECL, 443, 456
 YY_DECL_IS_OURS, 443
 YY_END_OF_BUFFER, 443
 YY_EXIT_FAILURE, 443
 YY_EXTRA_TYPE, 444
 YY_FATAL_ERROR, 444
 YY_FLUSH_BUFFER, 444
 YY_INPUT, 444
 YY_INT_ALIGNED, 444
 YY_LESS_FILENO, 444
 YY_MORE_ADJ, 444
 YY_NEW_FILE, 444
 YY_NULL, 444
 YY_NUM_RULES, 444
 YY_READ_BUF_SIZE, 444
 YY_RULE_SETUP, 444
 YY_SC_TO_UI, 445
 YY_SKIP_YYWRAP, 445
 YY_START, 445
 YY_STATE_BUF_SIZE, 445
 YY_STATE_EOF, 445
 YY_USER_ACTION, 445
 YYSTATE, 446
 YYTABLES_NAME, 446
 yy_act, 455
 yy_bp, 455
 yy_cp, 455
 yy_create_buffer, 447
 yy_delete_buffer, 448
 yy flex_debug, 456
 yy_flush_buffer, 448
 yy_new_buffer, 444
 yy_scan_buffer, 449
 yy_scan_bytes, 449
 yy_scan_string, 450
 yy_set_bol, 445
 yy_set_interactive, 445
 yy_size_t, 447
 yy_state_type, 447
 yy_switch_to_buffer, 451
 yyalloc, 451
 yycolumn, 456
 yyconst, 445
 yyfree, 451
 yyget_debug, 452
 yyget_extra, 452
 yyget_in, 452
 yyget_leng, 452
 yyget_lineno, 452
 yyget_out, 452
 yyget_text, 452
 yyid, 456
 yyidpos, 456
 yyin, 456
 yyleng, 456
 yyless, 446
 yylex, 452
 yylex_destroy, 452
 yylineno, 456
 ymore, 446
 yyout, 456
 yypop_buffer_state, 453
 yypush_buffer_state, 453

yyrealloc, 453
yyrestart, 453
yyset_debug, 455
yyset_extra, 455
yyset_in, 455
yyset_lineno, 455
yyset_out, 455
yystr, 456
yyterminate, 446
yytext, 456
yytext_ptr, 446
yywrap, 446

LibMM
 MSG_ACTIVATE, 18
 MSG_ADD_VALUE, 18
 MSG_DEL_CONDITION, 18
 MSG_DEL_CONVERTER, 18
 MSG_DEL_DECL, 18
 MSG_DEL_DRAIN, 18
 MSG_DEL_FLOW, 18
 MSG_DEL_GATE, 18
 MSG_DEL_INST, 18
 MSG_DEL_POOL, 18
 MSG_DEL_REF, 18
 MSG_DEL_SOURCE, 18
 MSG_DEL_TRIGGER, 18
 MSG_DEL_TYPE, 18
 MSG_DISABLE, 18
 MSG_ENABLE, 18
 MSG_ERROR, 18
 MSG_FAIL, 18
 MSG_HAS_VALUE, 18
 MSG_NEW_CONDITION, 18
 MSG_NEW_CONVERTER, 18
 MSG_NEW_DECL, 18
 MSG_NEW_DRAIN, 18
 MSG_NEW_FLOW, 18
 MSG_NEW_GATE, 18
 MSG_NEW_INST, 18
 MSG_NEW_POOL, 18
 MSG_NEW_REF, 18
 MSG_NEW_SOURCE, 18
 MSG_NEW_TRIGGER, 18
 MSG_NEW_TYPE, 18
 MSG_PREVENT, 18
 MSG_SUB_VALUE, 18
 MSG_TRIGGER, 18
 MSG_UPD_CONDITION, 18
 MSG_UPD_CONVERTER, 18
 MSG_UPD_DECL, 18
 MSG_UPD_DRAIN, 18
 MSG_UPD_FLOW, 18
 MSG_UPD_GATE, 18
 MSG_UPD_POOL, 18
 MSG_UPD_REF, 18
 MSG_UPD_SOURCE, 18
 MSG_UPD_TRIGGER, 18
 MSG_UPD_TYPE, 18

 MSG_VIOLATE, 18
 LibMM, 17
 MESSAGE, 17
 linebreak
 MM::String, 355
 Location, 184
 MM::Location, 185

MAX
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489

MM
 T_Activation, 22
 T_ActiveExp, 23
 T_AliasExp, 23
 T_AllExp, 23
 T_Assertion, 23
 T_BinExp, 23
 T_BooleanValExp, 23
 T_ConverterNodeBehavior, 23
 T_Declaration, 23
 T_Definition, 23
 T_Delegate, 22
 T_Deletion, 23
 T_DieExp, 23
 T_Disablement, 23
 T_DrainNodeBehavior, 23
 T_Edge, 23
 T_Element, 23
 T_Enablement, 22
 T_Evaluator, 22
 T_Event, 22
 T_Exp, 23
 T_Failure, 22
 T_FlowEdge, 23
 T_FlowEvent, 22
 T_GateNodeBehavior, 23
 T_Instance, 22
 T_InstanceObserver, 22
 T_InterfaceNode, 23
 T_Location, 23
 T_Machine, 22
 T_Map, 22
 T_Modification, 22
 T_NULL, 22
 T_Name, 23
 T_Node, 23
 T_NodeBehavior, 23
 T_NumberValExp, 23
 T_Observable, 22
 T_Observer, 22
 T_OneExp, 23
 T_OverrideExp, 23
 T_PoolNodeBehavior, 23
 T_Prevention, 22
 T_Program, 22
 T_RangeValExp, 23
 T_Recyclable, 22
 T_Recycler, 22

T_RefNodeBehavior, 23
 T_Reflector, 22
 T_SourceNodeBehavior, 23
 T_StateEdge, 23
 T_String, 22
 T_Transformation, 22
 T_Transition, 22
 T_TriggerEvent, 22
 T_TriggerExp, 23
 T_UnExp, 23
 T_ValExp, 23
 T_VarExp, 23
 T_Vector, 22
 T_Violation, 22
MM::NodeBehavior
 ACT_ERROR, 259
 ACT_PULL, 259
 ACT_PUSH, 259
 HOW_ALL, 259
 HOW_ANY, 259
 HOW_ERROR, 259
 IO_ERROR, 259
 IO_IN, 259
 IO_INOUT, 259
 IO_OUT, 259
 IO_PRIVATE, 259
 WHEN_AUTO, 260
 WHEN_ERROR, 260
 WHEN_PASSIVE, 260
 WHEN_START, 260
 WHEN_USER, 260
MM::Operator
 OP_ADD, 287
 OP_AND, 287
 OP_DIV, 287
 OP_EQ, 288
 OP_ERROR, 287
 OP_GE, 288
 OP_GT, 288
 OP_LE, 288
 OP_LT, 288
 OP_MUL, 287
 OP_NEQ, 288
 OP_NOT, 288
 OP_OR, 288
 OP_PER, 288
 OP_PERCENT, 288
 OP_SUB, 287
 OP_UNM, 288
MODIFY
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489
MSG_ACTIVATE
 LibMM, 18
MSG_ADD_VALUE
 LibMM, 18
MSG_DEL_CONDITION
 LibMM, 18
 MSG_DEL_CONVERTER
 LibMM, 18
 MSG_DEL_DECL
 LibMM, 18
 MSG_DEL_DRAIN
 LibMM, 18
 MSG_DEL_FLOW
 LibMM, 18
 MSG_DEL_GATE
 LibMM, 18
 MSG_DEL_INST
 LibMM, 18
 MSG_DEL_POOL
 LibMM, 18
 MSG_DEL_REF
 LibMM, 18
 MSG_DEL_SOURCE
 LibMM, 18
 MSG_DEL_TRIGGER
 LibMM, 18
 MSG_DEL_TYPE
 LibMM, 18
 MSG_DISABLE
 LibMM, 18
 MSG_ENABLE
 LibMM, 18
 MSG_ERROR
 LibMM, 18
 MSG_FAIL
 LibMM, 18
 MSG_HAS_VALUE
 LibMM, 18
 MSG_NEW_CONDITION
 LibMM, 18
 MSG_NEW_CONVERTER
 LibMM, 18
 MSG_NEW_DECL
 LibMM, 18
 MSG_NEW_DRAIN
 LibMM, 18
 MSG_NEW_FLOW
 LibMM, 18
 MSG_NEW_GATE
 LibMM, 18
 MSG_NEW_INST
 LibMM, 18
 MSG_NEW_POOL
 LibMM, 18
 MSG_NEW_REF
 LibMM, 18
 MSG_NEW_SOURCE
 LibMM, 18
 MSG_NEW_TRIGGER
 LibMM, 18
 MSG_NEW_TYPE
 LibMM, 18
 MSG_PREVENT
 LibMM, 18

MSG_SUB_VALUE
 LibMM, 18
MSG_TRIGGER
 LibMM, 18
MSG_UPD_CONDITION
 LibMM, 18
MSG_UPD_CONVERTER
 LibMM, 18
MSG_UPD_DECL
 LibMM, 18
MSG_UPD_DRAIN
 LibMM, 18
MSG_UPD_FLOW
 LibMM, 18
MSG_UPD_GATE
 LibMM, 18
MSG_UPD_POOL
 LibMM, 18
MSG_UPD_REF
 LibMM, 18
MSG_UPD_SOURCE
 LibMM, 18
MSG_UPD_TRIGGER
 LibMM, 18
MSG_UPD_TYPE
 LibMM, 18
MSG_VIOLATE
 LibMM, 18
MUL
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490
MESSAGE
 LibMM, 17
MM, 19
 BOOLEAN, 21
 CALLBACK, 21
 CHAR, 21
 INT32, 21
 TID, 22
 UINT32, 22
 UINT8, 22
 VOID, 22
MM::Activation, 25
 ~Activation, 27
 Activation, 26, 27
 getLocation, 27
 getMessage, 27
 getTypeld, 27
 instanceof, 27
 recycle, 27
 toString, 27
MM::ActiveExp, 28
 ~ActiveExp, 30
 ActiveExp, 30
 getName, 30
 getTypeld, 30
 instanceof, 30
 recycle, 31
 toString, 31
MM::AliasExp, 32
 ~AliasExp, 34
 AliasExp, 34
 getTypeld, 34
 instanceof, 34
 recycle, 35
 toString, 35
MM::AllExp, 36
 ~AllExp, 37
 AllExp, 37
 getTypeld, 37
 instanceof, 37
 recycle, 38
 toString, 38
MM::Assertion, 39
 ~Assertion, 41
 Assertion, 41
 getExp, 41
 getLocation, 41
 getMessage, 41
 getMessageLength, 41
 getTypeld, 41
 instanceof, 41
 recycle, 41
 toString, 42
MM::BinExp, 42
 ~BinExp, 44
 BinExp, 44
 getLhsExp, 44
 getOperator, 44
 getRhsExp, 44
 getTypeld, 45
 instanceof, 45
 isAlaiaExp, 45
 isTriggerExp, 45
 recycle, 46
 toString, 46
MM::BooleanValExp, 47
 ~BooleanValExp, 48
 BooleanValExp, 48
 getLocation, 49
 getTypeld, 49
 getValue, 49
 greaterEquals, 49
 instanceof, 49
 recycle, 49
 toString, 50
MM::ConverterNodeBehavior, 51
 ~ConverterNodeBehavior, 53
 activateTriggerTargets, 53
 add, 53
 begin, 53
 change, 53
 ConverterNodeBehavior, 53
 end, 53
 getCapacity, 53
 getCreateMessage, 53

getDeleteMessage, 53
 getDrainNode, 54
 getFrom, 54
 getResources, 54
 getSourceNode, 54
 getTo, 54
 getTriggerEdge, 54
 getTypeId, 54
 getUpdateMessage, 54
 hasCapacity, 54
 hasResources, 54
 instanceof, 54
 recycle, 54
 setDrainNode, 55
 setFrom, 55
 setSourceNode, 55
 setTo, 55
 setTriggerEdge, 55
 step, 55
 stepPullAll, 55
 stepPullAny, 55
 stepPushAll, 55
 stepPushAny, 55
 sub, 55
 toString, 55, 56
MM::Declaration, 57
 ~Declaration, 58
 addInterface, 58
 begin, 59
 change, 60
 Declaration, 58
 end, 60
 getDefinition, 60
 getInterface, 61
 getTypeId, 61
 getTypeName, 61
 instanceof, 61
 recycle, 61
 removeInterface, 61
 setDefinition, 62
 toString, 62
 update, 62
MM::Definition, 63
 ~Definition, 65
 addElement, 65
 addPullAllNode, 65
 addPullAnyNode, 66
 addPushAllNode, 66
 addPushAnyNode, 66
 COMMA_CHAR, 73
 clearElements, 66
 containsElement, 66
 Definition, 65
 deprioritize, 66
 findDeclaredDefinition, 66
 findNode, 67
 findQueriedDefinition, 67
 getElement, 67
 getElements, 68
 getPullAllNodes, 68
 getPullAnyNodes, 68
 getPushAllNodes, 68
 getPushAnyNodes, 68
 getTypeId, 68
 instanceof, 68
 LBRACE_CHAR, 73
 prioritize, 69
 putElement, 69
 RBRACE_CHAR, 73
 recycle, 69
 removeElement, 69, 70
 removePullAllNode, 70
 removePullAnyNode, 70
 removePushAllNode, 70
 removePushAnyNode, 70
 setNameToElementMap, 70
 setParent, 70
 setPullAllNodes, 70
 setPullAnyNodes, 71
 setPushAllNodes, 71
 setPushAnyNodes, 71
 toString, 72
MM::Deletion, 76
 ~Deletion, 78
 Deletion, 78
 getLocation, 78
 getTypeId, 78
 instanceof, 78
 recycle, 78
 toString, 78
MM::DieExp, 79
 ~DieExp, 81
 DieExp, 81
 getMax, 81
 getTypeId, 82
 instanceof, 82
 recycle, 82
 toString, 83
MM::Disablement, 84
 ~Disablement, 86
 Disablement, 85, 86
 getLocation, 86
 getMessage, 86
 getName, 86
 getTypeId, 86
 instanceof, 86
 recycle, 86
 toString, 86
MM::DrainNodeBehavior, 87
 ~DrainNodeBehavior, 89
 add, 89
 begin, 89
 change, 89
 DrainNodeBehavior, 89
 end, 89
 getCapacity, 89

getCreateMessage, 89
getDeleteMessage, 90
getResources, 90
getTypeld, 90
getUpdateMessage, 90
hasCapacity, 90
hasResources, 90
instanceof, 90
recycle, 90
stepPullAll, 91
stepPushAll, 91
stepPushAny, 92
sub, 92
toString, 92
MM::Edge, 93
~Edge, 94
Edge, 94
getExp, 95
getSource, 95
getSourceName, 95
getTarget, 95
getTargetName, 96
getTypeld, 96
instanceof, 96
recycle, 97
setExp, 97
setSource, 97
setTarget, 98
toString, 98
MM::Element, 99
~Element, 100
begin, 101
change, 101
Element, 100
end, 101
getName, 101
getTypeld, 102
instanceof, 103
isVisible, 104
name, 107
recycle, 105
setName, 106
setVisible, 106
toString, 106
visible, 107
MM::Enablement, 107
~Enablement, 109
Enablement, 109
getLocation, 109
getMessage, 109
getName, 109
getTypeld, 109
instanceof, 109
recycle, 109
toString, 110
MM::Evaluator, 111
~Evaluator, 112
eval, 112–120
Evaluator, 112
getTypeld, 120
instanceof, 120
recycle, 120
step, 120
toString, 121
MM::Event, 122
~Event, 123
element, 126
Event, 123
getElement, 124
getInstance, 124
getMessage, 124
getTypeld, 124
instance, 126
instanceof, 124
recycle, 125
setElement, 126
setInstance, 126
toString, 126
MM::Exp, 127
~Exp, 128
Exp, 128
getTypeld, 129
instanceof, 129
recycle, 130
toString, 131
MM::Failure, 132
~Failure, 134
Failure, 134
getLocation, 134
getMessage, 134
getTypeld, 134
instanceof, 134
recycle, 134
toString, 135
MM::FlowEdge, 135
~FlowEdge, 137
FlowEdge, 137
getTypeld, 137
instanceof, 137
recycle, 137
toString, 137, 138
MM::FlowEvent, 139
~FlowEvent, 141
FlowEvent, 141
getActEdge, 141
getActInstance, 141
getActNode, 141
getAmount, 141
getMessage, 141
getSourceInstance, 141
getSourceNode, 141
getTargetInstance, 142
getTargetNode, 142
getTypeld, 142
instanceof, 142
recycle, 142

setSourceInstance, 142
 setTargetInstance, 142
 toString, 142
MM::GateNodeBehavior, 143
 ~GateNodeBehavior, 145
 add, 145
 begin, 145
 change, 145
 end, 146
 GateNodeBehavior, 145
 getCapacity, 146
 getCreateMessage, 146
 getDeleteMessage, 146
 getResources, 146
 getTypeId, 146
 getUpdateMessage, 146
 hasCapacity, 146
 hasResources, 146
 instanceof, 146
 recycle, 147
 stepPullAll, 147
 stepPushAll, 147
 sub, 147
 toString, 147, 148
MM::Instance, 149
 ~Instance, 151
 add, 151
 begin, 151
 clearActive, 152
 clearDisabled, 152
 createInstances, 152
 deleteValue, 153
 destroyAllInstances, 153
 destroyInstance, 154
 destroyInstances, 154
 finalize, 155
 getCapacity, 155
 getDeclaration, 155
 getDefinition, 156
 getEvaluatedExp, 156
 getGateValue, 157
 getIndex, 157
 getInstance, 158
 getInstances, 159
 getNewValue, 159
 getOldValue, 159
 getParent, 160
 getResources, 160
 getTypeId, 160
 getValue, 161
 hasCapacity, 161
 hasResources, 162
 Instance, 150
 instanceof, 162
 isActive, 163
 isDisabled, 163
 isEvaluatedExp, 163
 isMarked, 164
 mark, 164
 nameToString, 165
 recycle, 165
 setActive, 166
 setDisabled, 166
 setEvaluatedExp, 166
 setGateValue, 167
 setNewValue, 167
 setNextActive, 168
 setOldValue, 168
 setValue, 168
 sub, 169
 sweep, 169
 toString, 170, 171
 update, 171
MM::InterfaceNode, 175
 ~InterfaceNode, 177
 activateTriggerTargets, 177
 add, 177
 getBehavior, 177
 getCapacity, 177
 getDeclaration, 177
 getNode, 177
 getResources, 177
 getTypeId, 177
 hasCapacity, 177
 hasResources, 177
 instanceof, 177
 InterfaceNode, 176
 recycle, 178
 sub, 178
 toString, 178
MM::Location, 184
 ~Location, 186
 getColumn, 186
 getLength, 186
 getLine, 186
 getTypeId, 186
 instanceof, 186
 Location, 185
 recycle, 187
 toString, 187
MM::Machine, 188
 ~Machine, 192
 activate, 192
 addDefinitionObserver, 192
 addInstanceObserver, 193
 createActivation, 193
 createActiveExp, 193, 194
 createAliasExp, 194
 createAllExp, 194
 createAnonymousTriggerEdge, 195
 createAssertion, 195
 createBinExp, 196
 createBooleanValExp, 196, 197
 createConverterNode, 197
 createDeclaration, 198
 createDefinition, 198, 199

createDeletion, 199
createDisablement, 200
createDrainNode, 200
createEdgeVector, 201
createElementVector, 201
createEnablement, 201, 202
createFailure, 202
createFlowEdge, 203
createFlowEvent, 203
createGateNode, 204
createInstance, 204
createInterfaceNode, 205
createLocation, 205
createModification, 206, 207
createName, 207, 208
createName2ElementMap, 209
createName2NodeMap, 209
createNodeVector, 209
createNumberValExp, 209
createOneExp, 210
createOverrideExp, 210, 211
createPoolNode, 211
createPrevention, 212
createProgram, 212, 213
createRangeValExp, 213, 214
createRefNode, 214
createSourceNode, 214
createStateEdge, 215
createString, 215
createTransformationVector, 216
createTransition, 216, 217
createTriggerEvent, 217, 218
createUnExp, 218
createVarExp, 219
createViolation, 219, 220
delegates, 227
eatWhiteSpace, 220
eval, 220
evalFile, 221
getDefinition, 223
getEvaluator, 223
getInstance, 223
getInstanceName, 224
getLog, 224
getName, 224
getReflector, 225
getTypeld, 225
instanceof, 225
LOG_SIZE, 227
Machine, 192
reflect, 226
removeObserver, 226
reset, 226
setDefinition, 226
setInstance, 226
step, 226, 227
MM::Machine::Delegate, 74
~Delegate, 75

Delegate, 75
getTypeld, 75
instanceof, 75
update, 76

MM::Machine::InstanceObserver, 172
~InstanceObserver, 173
getTypeld, 174
InstanceObserver, 173
instanceof, 174
update, 174

MM::Map
~Map, 228
clear, 228
contains, 228
get, 228
getIterator, 228
getTypeld, 229
instanceof, 229
isEmpty, 229
Map, 228
put, 229
putAll, 229
remove, 229

MM::Map::Iterator
~Iterator, 179
getNext, 179, 180
hasNext, 180
Iterator, 179
reset, 180

MM::Modification, 230
~Modification, 231
getTypeld, 231
instanceof, 231
Modification, 231
recycle, 232
toString, 232

MM::Name, 233
~Name, 234
append, 235
equals, 235, 236
getBuffer, 236
getLength, 236
getLocation, 237
getName, 237
getPreName, 237
getTypeld, 238
instanceof, 238
Name, 234
operator<, 243
print, 238
recycle, 238
setLocation, 240
setName, 240
setPreName, 241
toString, 241

MM::Name::Compare, 50
operator(), 51

MM::Node, 243

~Node, 245
 activateTriggerTargets, 245
 add, 245
 addAlias, 246
 addCondition, 246
 addInput, 246
 addOutput, 246
 addTrigger, 246
 begin, 246
 change, 246
 end, 246
 getAliases, 246
 getAmount, 247
 getBehavior, 247
 getCapacity, 248
 getConditions, 248
 getInput, 249
 getOutput, 249
 getResources, 250
 getTriggers, 250
 getTypeld, 251
 hasCapacity, 251
 hasEdgeOwnership, 251
 hasResources, 251
 instanceof, 251
 isDisabled, 252
 isSatisfied, 252
 Node, 245
 recycle, 253
 removeAlias, 254
 removeCondition, 254
 removeInput, 254
 removeOutput, 254
 removeTrigger, 254
 setAliases, 254
 setBehavior, 254
 setConditions, 254
 setEdgeOwnership, 254
 setInput, 254
 setOutput, 255
 setTriggers, 255
 step, 255
 sub, 255
 toString, 256
 MM::Node::Compare, 51
 operator(), 51
 MM::NodeBehavior, 257
 ~NodeBehavior, 260
 __Act, 259
 __How, 259
 __IO, 259
 __When, 259
 ACT_LEN, 269
 ACT_STR, 269
 Act, 259
 activateTriggerTargets, 260
 add, 260
 begin, 260
 change, 260
 conformsTo, 261
 end, 261
 getAct, 261
 getAmount, 261
 getCapacity, 261
 getCreateMessage, 261
 getDeleteMessage, 262
 getHow, 262
 getIO, 262
 getResources, 262
 getTypeld, 263
 getUpdateMessage, 263
 getWhen, 263
 getWork, 263
 HOW_LEN, 269
 HOW_STR, 270
 hasCapacity, 264
 hasResources, 264
 How, 259
 IO, 259
 IO_LEN, 270
 IO_STR, 270
 instanceof, 264
 NodeBehavior, 260
 recycle, 265
 setAct, 266
 setHow, 266
 setIO, 266
 setWhen, 266
 step, 266
 stepPullAll, 267
 stepPullAny, 267
 stepPushAll, 267
 stepPushAny, 268
 sub, 268
 toString, 268
 WHEN_LEN, 270
 WHEN_STR, 270
 When, 259
 MM::NodeWorkItem, 271
 ~NodeWorkItem, 271
 getEdge, 271
 getInstance, 272
 getNode, 272
 NodeWorkItem, 271
 MM::NumberValExp, 274
 ~NumberValExp, 276
 getIntValue, 276
 getLocation, 276
 getTypeld, 276
 getValue, 276
 greaterEquals, 276
 instanceof, 277
 NumberValExp, 275, 276
 recycle, 277
 toString, 277
 MM::Observable, 278

~Observable, 279
addObserver, 279
getTypeld, 279
instanceof, 279
notifyObservers, 279
Observable, 278
recylce, 280
removeObserver, 280
MM::Observer, 281
~Observer, 282
getTypeld, 282
instanceof, 282
Observer, 282
update, 283
MM::OneExp, 283
~OneExp, 285
getTypeld, 285
greaterEquals, 285
instanceof, 286
OneExp, 285
recycle, 286
toString, 286
MM::Operator, 287
__OP, 287
OP, 287
OP_LEN, 288
OP_STR, 288
MM::OverrideExp, 289
~OverrideExp, 291
getExp, 291
getTypeld, 291
instanceof, 291
OverrideExp, 290, 291
recycle, 292
toString, 292
MM::PoolNodeBehavior, 293
~PoolNodeBehavior, 295
add, 295
addInterface, 296
begin, 297
change, 298
end, 298
getAdd, 299
getAmount, 299
getAt, 299
getCapacity, 299
getCreateMessage, 300
getDefinition, 300
getDeleteMessage, 300
getInterface, 300
getMax, 300
getOf, 300
getResources, 300
getTypeld, 301
getTypeName, 301
getUpdateMessage, 301
hasCapacity, 301
hasResources, 301
instanceof, 302
PoolNodeBehavior, 295
recycle, 302
removeInterface, 302
setAdd, 303
setAt, 303
setDefinition, 303
setMax, 303
stepPullAll, 303
stepPushAll, 304
sub, 305
toString, 306
update, 307
MM::Prevention, 307
~Prevention, 309
getLocation, 309
getMessage, 309
getTypeld, 309
instanceof, 309
Prevention, 308, 309
recycle, 309
toString, 309
MM::Program, 310
~Program, 311
addTransformation, 311
getTransformations, 311
getTypeld, 311
instanceof, 311
Program, 311
recycle, 311
toString, 312
MM::RangeValExp, 313
~RangeValExp, 315
getIntValue, 315
getMax, 315
getMin, 315
getTypeld, 316
greaterEquals, 316
instanceof, 316
RangeValExp, 315
recycle, 316
toString, 317
MM::Recyclable, 317
~Recyclable, 319
getTypeld, 319
instanceof, 319
Recyclable, 319
recycle, 320
toString, 322
MM::Recycler, 323
~Recycler, 324
create, 324
createBuffer, 325
getTypeld, 326
instanceof, 326
recycle, 326
Recycler, 324
TYPE_STR, 327

uncreate, 326, 327
MM::RefNodeBehavior, 334
 ~RefNodeBehavior, 335
 add, 335
 begin, 335
 change, 336
 doTriggers, 336
 end, 336
 getAlias, 336
 getCapacity, 336
 getCreateMessage, 336
 getDeleteMessage, 336
 getReference, 336
 getResources, 337
 getTypeld, 337
 getUpdateMessage, 337
 hasCapacity, 337
 hasResources, 337
 instanceof, 337
 recycle, 337
 RefNodeBehavior, 335
 setAlias, 338
 step, 338
 stepPullAll, 338
 stepPullAny, 338
 stepPushAll, 338
 stepPushAny, 338
 sub, 338
 toString, 338, 339
MM::Reflector, 328
 ~Reflector, 329
 addElement, 329
 deinit, 329
 getDefinition, 329
 getInstance, 329
 getTypeld, 330
 init, 330, 331
 instanceof, 332
 merge, 332
 Reflector, 329
 removeElement, 333
MM::SourceNodeBehavior, 339
 ~SourceNodeBehavior, 341
 add, 341
 begin, 341
 change, 341
 end, 341
 getCapacity, 341
 getCreateMessage, 341
 getDeleteMessage, 341
 getResources, 342
 getTypeld, 342
 getUpdateMessage, 342
 hasCapacity, 342
 hasResources, 342
 instanceof, 342
 recycle, 342
 SourceNodeBehavior, 341
 stepPullAll, 342
 stepPullAny, 343
 stepPushAll, 343
 sub, 343
 toString, 344
MM::StateEdge, 345
 ~StateEdge, 346
 getTypeld, 347
 instanceof, 347
 isAlias, 347
 isCondition, 347
 isTrigger, 347
 recycle, 347
 StateEdge, 346
 toString, 348
MM::String, 349
 ~String, 350
 append, 350, 351
 appendInt, 351
 clear, 352
 getBuffer, 352
 getSize, 353
 getTypeld, 354
 getUsed, 354
 instanceof, 355
 linebreak, 355
 print, 355
 recycle, 356
 space, 356, 357
 String, 350
 toString, 357
MM::Transformation, 358
 ~Transformation, 359
 addElement, 359
 clearElements, 360
 getElements, 360
 getTypeld, 361
 instanceof, 361
 recycle, 362
 toString, 362
 Transformation, 359
MM::Transition, 363
 ~Transition, 365
 getTypeld, 365
 instanceof, 365
 recycle, 365
 toString, 366
 Transition, 365
MM::TriggerEvent, 367
 ~TriggerEvent, 369
 getLocation, 369
 getMessage, 369
 getTypeld, 369
 instanceof, 369
 recycle, 369
 toString, 370
 TriggerEvent, 369
MM::TriggerExp, 371

~TriggerExp, 372
getTypeld, 372
instanceof, 372
recycle, 374
TRIGGER_CHAR, 375
toString, 374
TriggerExp, 372
MM::UnExp, 375
~UnExp, 376
getExp, 377
getOperator, 377
getTypeld, 377
instanceof, 377
recycle, 378
toString, 378
UnExp, 376
MM::ValExp, 379
~ValExp, 380
getTypeld, 380
greaterEquals, 381
instanceof, 381
recycle, 382
toString, 383
ValExp, 380
MM::VarExp, 384
~VarExp, 386
getName, 386
getTypeld, 386
instanceof, 386
recycle, 387
toString, 387
VarExp, 385
MM::Vector
~Vector, 388
add, 388
addAll, 389
at, 389
clear, 389
contains, 389
elementAt, 389
getIterator, 390
getNewIterator, 391
getPosition, 391
getTypeld, 391
instanceof, 392
isEmpty, 392
pop, 392
remove, 392
size, 392
Vector, 388
MM::Vector< T >, 387
MM::Vector< T >::Iterator, 181
MM::Vector::Iterator
~Iterator, 181
getNext, 181
hasNext, 182
Iterator, 181
reset, 183
MM::Violation, 393
~Violation, 395
getLocation, 395
getMessage, 395
getTypeld, 395
instanceof, 395
recycle, 395
toString, 395
VIOLATE_LEN, 396
VIOLATE_STR, 396
Violation, 394, 395
MM_FALSE
Types.h, 528
MM_MAX_RESOURCES
Types.h, 528
MM_NULL
Types.h, 528
MM_TRUE
Types.h, 528
MM_parse
Machine.cpp, 460
mm.tab.cpp, 478
MM_parseFile
Machine.cpp, 462
mm.tab.cpp, 480
MM_printf
Types.h, 528
Machine, 188
MM::Machine, 192
Machine.cpp
MM_parse, 460
MM_parseFile, 462
main
mm.tab.cpp, 478
Map, 230
MM::Map, 228
mark
MM::Instance, 164
merge
MM::Reflector, 332
mm
mm.tab.cpp, 485
mm.tab.cpp
ACTIVATE, 475, 477
ACTIVE, 475, 477
ADD, 476, 477
ADDITION, 476, 477
ALIAS, 475, 477
ALL, 475, 476
AND, 475, 476
ANY, 475, 476
ASSERT, 475, 477
AT, 475, 477
AUTO, 475, 476
COLON, 476, 477
CONVERTER, 476, 477
DELETE, 475, 477
DICE, 475, 477

DISABLE, 475, 477
 DIV, 476, 477
 DOT, 475, 477
 DOT_GT, 475, 477
 DRAIN, 476, 477
 ENABLE, 475, 477
 EQ, 475, 476
 FAIL, 475, 477
 FALSE, 475, 477
 FPVAL, 476, 477
 FROM, 475, 476
 GATE, 476, 477
 GE, 475, 476
 GT, 475, 476
 ID, 474, 476
 IN, 475, 476
 INOUT, 475, 476
 LCURLY, 476, 477
 LE, 475, 476
 LPAREN, 476, 477
 LT, 475, 476
 MAX, 475, 477
 MODIFY, 475, 477
 MUL, 476, 477
 NE, 475, 476
 NOT, 475, 477
 OF, 475, 476
 OR, 475, 476
 OUT, 475, 476
 PASSIVE, 475, 476
 PER, 475, 477
 PERCENT, 476, 477
 POOL, 476, 477
 PREVENT, 475, 477
 PRIVATE, 474, 476
 PULL, 475, 476
 PUSH, 475, 476
 RANGE, 476, 477
 RCURLY, 476, 477
 REF, 475, 477
 RPAREN, 476, 477
 SOURCE, 476, 477
 START, 475, 476
 STEP, 475, 477
 STRING, 476, 477
 SUB, 476, 477
 SUB_GT, 476, 477
 TO, 475, 476
 TRIGGER, 475, 477
 TRUE, 475, 477
 UNM, 476, 477
 USER, 475, 476
 VIOLATE, 475, 477
 mm.tab.hpp
 ACTIVATE, 488, 489
 ACTIVE, 488, 489
 ADD, 488, 490
 ADDITION, 488, 490
 ALIAS, 488, 489
 ALL, 487, 489
 AND, 487, 489
 ANY, 487, 489
 ASSERT, 488, 489
 AT, 488, 489
 AUTO, 487, 489
 COLON, 488, 490
 CONVERTER, 488, 490
 DELETE, 488, 489
 DICE, 488, 489
 DISABLE, 488, 489
 DIV, 488, 490
 DOT, 488, 489
 DOT_GT, 488, 489
 DRAIN, 488, 490
 ENABLE, 488, 489
 EQ, 487, 489
 FAIL, 488, 489
 FALSE, 488, 489
 FPVAL, 488, 490
 FROM, 487, 489
 GATE, 488, 490
 GE, 487, 489
 GT, 487, 489
 ID, 487, 488
 IN, 487, 488
 INOUT, 487, 489
 LCURLY, 488, 490
 LE, 487, 489
 LPAREN, 488, 490
 LT, 487, 489
 MAX, 488, 489
 MODIFY, 488, 489
 MUL, 488, 490
 NE, 487, 489
 NOT, 487, 489
 OF, 487, 489
 OR, 487, 489
 OUT, 487, 489
 PASSIVE, 487, 489
 PER, 488, 489
 PERCENT, 488, 490
 POOL, 488, 490
 PREVENT, 488, 489
 PRIVATE, 487, 488
 PULL, 487, 489
 PUSH, 487, 489
 RANGE, 488, 490
 RCURLY, 488, 490
 REF, 488, 489
 RPAREN, 488, 490
 SOURCE, 488, 490
 START, 487, 489
 STEP, 488, 489
 STRING, 488, 490
 SUB, 488, 490
 SUB_GT, 488, 489

TO, 487, 489
TRIGGER, 488, 489
TRUE, 487, 489
UNM, 488, 490
USER, 487, 489
VIOLATE, 488, 489

mm.tab.cpp
 for, 478
 if, 478
 MM_parse, 478
 MM_parseFile, 480
 main, 478
 mm, 485
 program, 485
 switch, 482
 while, 482
 YY_, 471
 YY_BUFFER_STATE, 474
 YY_LOCATION_PRINT, 471
 YY_REDUCE_PRINT, 471
 YY_STACK_PRINT, 471
 YY_SYMBOL_PRINT, 471, 482
 YYABORT, 471
 YYACCEPT, 471
 YYBACKUP, 471
 YYBISON, 471
 YYBISON_VERSION, 472
 YYCOPY, 472
 YYDEBUG, 472
 YYDPRINTF, 472
 YYEMPTY, 472
 YYEOF, 472
 YYERRCODE, 472
 YYERROR, 472
 YYERROR_VERBOSE, 472
 YYFAIL, 472
 YYFINAL, 472
 YYFREE, 472
 YYID, 472
 YYINITDEPTH, 472
 YYLAST, 472
 YYLEX, 472
 YYLLOC_DEFAULT, 472
 YYLSP_NEEDED, 473
 YYMALLOC, 473
 YYMAXDEPTH, 473
 YYMAXUTOK, 473
 YYNNTS, 473
 YYNRULES, 473
 YYNSTATES, 473
 YYNTOKENS, 473
 YYPACT_NINF, 473
 YYPOPSTACK, 473
 YYPULL, 473
 YYPURE, 473
 YYPUSH, 473
 YYRECOVERING, 473
 YYRHSLOC, 473

 YYSIZE_MAXIMUM, 473
 YYSIZE_T, 473
 YYSKELTON_NAME, 473
 YYSTACK_ALLOC, 473
 YYSTACK_BYTES, 473
 YYSTACK_FREE, 473
 YYSTACK_RELOCATE, 474
 YYSTYPE, 474
 YYSTYPE_IS_TRIVIAL, 474
 YYTABLE_NINF, 474
 YYTERROR, 474
 YYTOKEN_TABLE, 474
 YYTOKENTYPE, 474
 YYTRANSLATE, 474
 YYUNDEFTOK, 474
 YYUSE, 474, 485
 yy_scan_buffer, 482
 yychar, 485
 yyclearin, 472
 yycolumn, 485
 yyd, 485
 yyerrok, 472
 yyerror, 483
 yyin, 485
 yylen, 485
 yylex, 483
 yylineno, 485
 yyloc, 485
 yylocationp, 485
 yylval, 485
 yynerrs, 485
 yparse, 483
 yys, 485
 yysrc, 485
 ystype, 474
 yytokentype, 474
 yytype, 485
 yytype_int16, 474
 yytype_int8, 474
 yytype_uint16, 474
 yytype_uint8, 474
 yyvaluep, 485

mm.tab.hpp
 YYLTYPE, 487
 YYLTYPE_IS_TRIVIAL, 487
 YYSTYPE, 487
 YYSTYPE_IS_TRIVIAL, 487
 yyloc, 490
 yytype, 487
 ylval, 490
 ystype, 487
 yytokentype, 487
Modification, 233
 MM::Modification, 231

NE
 mm.tab.cpp, 475, 476
 mm.tab.hpp, 487, 489

NOT

mm.tab.cpp, 475, 477
 mm.tab.hpp, 487, 489
Name, 243
 MM::Name, 234
name
 MM::Element, 107
 YYSTYPE, 400
nameToString
 MM::Instance, 165
Node, 256
 MM::Node, 245
NodeBehavior, 271
 MM::NodeBehavior, 260
NodeWorkItem, 273
 MM::NodeWorkItem, 271
notifyObservers
 MM::Observable, 279
NumberValExp, 273
 MM::NumberValExp, 275, 276
OF
 mm.tab.cpp, 475, 476
 mm.tab.hpp, 487, 489
OP_ADD
 MM::Operator, 287
OP_AND
 MM::Operator, 287
OP_DIV
 MM::Operator, 287
OP_EQ
 MM::Operator, 288
OP_ERROR
 MM::Operator, 287
OP_GE
 MM::Operator, 288
OP_GT
 MM::Operator, 288
OP_LE
 MM::Operator, 288
OP_LT
 MM::Operator, 288
OP_MUL
 MM::Operator, 287
OP_NEQ
 MM::Operator, 288
OP_NOT
 MM::Operator, 288
OP_OR
 MM::Operator, 288
OP_PER
 MM::Operator, 288
OP_PERCENT
 MM::Operator, 288
OP_SUB
 MM::Operator, 287
OP_UNM
 MM::Operator, 288
OR
 mm.tab.cpp, 475, 476
OUT
 mm.tab.hpp, 487, 489
OP
 MM::Operator, 287
OP_LEN
 MM::Operator, 288
OP_STR
 MM::Operator, 288
Observable, 281
 MM::Observable, 278
Observer, 281
 MM::Observer, 282
OneExp, 287
 MM::OneExp, 285
Operator, 289
operator<
 MM::Name, 243
operator()
 MM::Name::Compare, 51
 MM::Node::Compare, 51
OverrideExp
 MM::OverrideExp, 290, 291
OvertideExp, 293
PASSIVE
 mm.tab.cpp, 475, 476
 mm.tab.hpp, 487, 489
PER
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489
PERCENT
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490
POOL
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490
PREVENT
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489
PRIVATE
 mm.tab.cpp, 474, 476
 mm.tab.hpp, 487, 488
PULL
 mm.tab.cpp, 475, 476
 mm.tab.hpp, 487, 489
PUSH
 mm.tab.cpp, 475, 476
 mm.tab.hpp, 487, 489
PoolNodeBehavior, 307
 MM::PoolNodeBehavior, 295
pop
 MM::Vector, 392
Prevention
 MM::Prevention, 308, 309
print
 MM::Name, 238
 MM::String, 355

prioritize
 MM::Definition, 69
Program, 313
 MM::Program, 311
program
 mm.tab.cpp, 485
 YYSTYPE, 400
put
 MM::Map, 229
putAll
 MM::Map, 229
putElement
 MM::Definition, 69

RANGE
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490

RCURLY
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490

REF
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489

RPAREN
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490

RBRACE_CHAR
 MM::Definition, 73

REJECT
 lex.mm.cpp, 443

RangeValExp, 317
 MM::RangeValExp, 315

Recyclable, 323
 MM::Recyclable, 319

recycle
 MM::Activation, 27
 MM::ActiveExp, 31
 MM::AliasExp, 35
 MM::AllExp, 38
 MM::Assertion, 41
 MM::BinExp, 46
 MM::BooleanValExp, 49
 MM::ConverterNodeBehavior, 54
 MM::Declaration, 61
 MM::Definition, 69
 MM::Deletion, 78
 MM::DieExp, 82
 MM::Disablement, 86
 MM::DrainNodeBehavior, 90
 MM::Edge, 97
 MM::Element, 105
 MM::Enablement, 109
 MM::Evaluator, 120
 MM::Event, 125
 MM::Exp, 130
 MM::Failure, 134
 MM::FlowEdge, 137
 MM::FlowEvent, 142
 MM::GateNodeBehavior, 147

 MM::Instance, 165
 MM::InterfaceNode, 178
 MM::Location, 187
 MM::Modification, 232
 MM::Name, 238
 MM::Node, 253
 MM::NodeBehavior, 265
 MM::NumberValExp, 277
 MM::OneExp, 286
 MM::OverrideExp, 292
 MM::PoolNodeBehavior, 302
 MM::Prevention, 309
 MM::Program, 311
 MM::RangeValExp, 316
 MM::Recyclable, 320
 MM::Recycler, 326
 MM::RefNodeBehavior, 337
 MM::SourceNodeBehavior, 342
 MM::StateEdge, 347
 MM::String, 356
 MM::Transformation, 362
 MM::Transition, 365
 MM::TriggerEvent, 369
 MM::TriggerExp, 374
 MM::UnExp, 378
 MM::ValExp, 382
 MM::VarExp, 387
 MM::Violation, 395

 Recycler, 328
 MM::Recycler, 324
 recyclce
 MM::Observable, 280
 RefNodeBehavior, 333
 MM::RefNodeBehavior, 335
 reflect
 MM::Machine, 226
 Reflector, 328
 MM::Reflector, 329
 remove
 MM::Map, 229
 MM::Vector, 392
 removeAlias
 MM::Node, 254
 removeCondition
 MM::Node, 254
 removeElement
 MM::Definition, 69, 70
 MM::Reflector, 333
 removeInput
 MM::Node, 254
 removeInterface
 MM::Declaration, 61
 MM::PoolNodeBehavior, 302
 removeObserver
 MM::Machine, 226
 MM::Observable, 280
 removeOutput
 MM::Node, 254

removePullAllNode
 MM::Definition, 70
 removePullAnyNode
 MM::Definition, 70
 removePushAllNode
 MM::Definition, 70
 removePushAnyNode
 MM::Definition, 70
 removeTrigger
 MM::Node, 254
 reset
 MM::Machine, 226
 MM::Map::Iterator, 180
 MM::Vector::Iterator, 183
SOURCE
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490
START
 mm.tab.cpp, 475, 476
 mm.tab.hpp, 487, 489
STEP
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489
STRING
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490
SUB
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490
SUB_GT
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 489
setAct
 MM::NodeBehavior, 266
setActive
 MM::Instance, 166
setAdd
 MM::PoolNodeBehavior, 303
setAlias
 MM::RefNodeBehavior, 338
setAliases
 MM::Node, 254
setAt
 MM::PoolNodeBehavior, 303
setBehavior
 MM::Node, 254
setConditions
 MM::Node, 254
setDefinition
 MM::Declaration, 62
 MM::Machine, 226
 MM::PoolNodeBehavior, 303
setDisabled
 MM::Instance, 166
setDrainNode
 MM::ConverterNodeBehavior, 55
setEdgeOwnership
 MM::Node, 254
 setElement
 MM::Event, 126
 setEvaluatedExp
 MM::Instance, 166
 setExp
 MM::Edge, 97
 setFrom
 MM::ConverterNodeBehavior, 55
 setGateValue
 MM::Instance, 167
 setHow
 MM::NodeBehavior, 266
 setIO
 MM::NodeBehavior, 266
 setInput
 MM::Node, 254
 setInstance
 MM::Event, 126
 MM::Machine, 226
 setLocation
 MM::Name, 240
 setMax
 MM::PoolNodeBehavior, 303
 setName
 MM::Element, 106
 MM::Name, 240
 setNameToElementMap
 MM::Definition, 70
 setNewValue
 MM::Instance, 167
 setNextActive
 MM::Instance, 168
 setOldValue
 MM::Instance, 168
 setOutput
 MM::Node, 255
 setParent
 MM::Definition, 70
 setPreName
 MM::Name, 241
 setPullAllNodes
 MM::Definition, 70
 setPullAnyNodes
 MM::Definition, 71
 setPushAllNodes
 MM::Definition, 71
 setPushAnyNodes
 MM::Definition, 71
 setSource
 MM::Edge, 97
 setSourceInstance
 MM::FlowEvent, 142
 setSourceNode
 MM::ConverterNodeBehavior, 55
 setTarget
 MM::Edge, 98
 setTargetInstance
 MM::FlowEvent, 142

setTo
 MM::ConverterNodeBehavior, 55
setTriggerEdge
 MM::ConverterNodeBehavior, 55
setTriggers
 MM::Node, 255
setValue
 MM::Instance, 168
setVisible
 MM::Element, 106
setWhen
 MM::NodeBehavior, 266
size
 MM::Vector, 392
SourceNodeBehavior, 339
 MM::SourceNodeBehavior, 341
space
 MM::String, 356, 357
StateEdge, 344
 MM::StateEdge, 346
step
 MM::ConverterNodeBehavior, 55
 MM::Evaluator, 120
 MM::Machine, 226, 227
 MM::Node, 255
 MM::NodeBehavior, 266
 MM::RefNodeBehavior, 338
stepPullAll
 MM::ConverterNodeBehavior, 55
 MM::DrainNodeBehavior, 91
 MM::GateNodeBehavior, 147
 MM::NodeBehavior, 267
 MM::PoolNodeBehavior, 303
 MM::RefNodeBehavior, 338
 MM::SourceNodeBehavior, 342
stepPullAny
 MM::ConverterNodeBehavior, 55
 MM::NodeBehavior, 267
 MM::RefNodeBehavior, 338
 MM::SourceNodeBehavior, 343
stepPushAll
 MM::ConverterNodeBehavior, 55
 MM::DrainNodeBehavior, 91
 MM::GateNodeBehavior, 147
 MM::NodeBehavior, 267
 MM::PoolNodeBehavior, 304
 MM::RefNodeBehavior, 338
 MM::SourceNodeBehavior, 343
stepPushAny
 MM::ConverterNodeBehavior, 55
 MM::DrainNodeBehavior, 92
 MM::NodeBehavior, 268
 MM::RefNodeBehavior, 338
str
 YYSTYPE, 400
String, 348
 MM::String, 350
sub

 MM::ConverterNodeBehavior, 55
 MM::DrainNodeBehavior, 92
 MM::GateNodeBehavior, 147
 MM::Instance, 169
 MM::InterfaceNode, 178
 MM::Node, 255
 MM::NodeBehavior, 268
 MM::PoolNodeBehavior, 305
 MM::RefNodeBehavior, 338
 MM::SourceNodeBehavior, 343
sweep
 MM::Instance, 169
switch
 mm.tab.cpp, 482
t
 YYSTYPE, 400
T_Activation
 MM, 22
T_ActiveExp
 MM, 23
T_AliasExp
 MM, 23
T_AllExp
 MM, 23
T_Assertion
 MM, 23
T_BinExp
 MM, 23
T_BooleanValExp
 MM, 23
T_ConverterNodeBehavior
 MM, 23
T_Declaration
 MM, 23
T_Definition
 MM, 23
T_Delegate
 MM, 22
T_Deletion
 MM, 23
T_DieExp
 MM, 23
T_Disablement
 MM, 23
T_DrainNodeBehavior
 MM, 23
T_Edge
 MM, 23
T_Element
 MM, 23
T_Enablement
 MM, 22
T_Evaluator
 MM, 22
T_Event
 MM, 22
T_Exp
 MM, 23

T_Failure
 MM, 22
T_FlowEdge
 MM, 23
T_FlowEvent
 MM, 22
T_GateNodeBehavior
 MM, 23
T_Instance
 MM, 22
T_InstanceObserver
 MM, 22
T_InterfaceNode
 MM, 23
T_Location
 MM, 23
T_Machine
 MM, 22
T_Map
 MM, 22
T_Modification
 MM, 22
T_NULL
 MM, 22
T_Name
 MM, 23
T_Node
 MM, 23
T_NodeBehavior
 MM, 23
T_NumberValExp
 MM, 23
T_Observable
 MM, 22
T_Observer
 MM, 22
T_OneExp
 MM, 23
T_OverrideExp
 MM, 23
T_PoolNodeBehavior
 MM, 23
T_Prevention
 MM, 22
T_Program
 MM, 22
T_RangeValExp
 MM, 23
T_Recyclable
 MM, 22
T_Recycler
 MM, 22
T_RefNodeBehavior
 MM, 23
T_Reflector
 MM, 22
T_SourceNodeBehavior
 MM, 23

T_StateEdge
 MM, 23
T_String
 MM, 22
T_Transformation
 MM, 22
T_Transition
 MM, 22
T_TriggerEvent
 MM, 22
T_TriggerExp
 MM, 23
T_UnExp
 MM, 23
T_ValExp
 MM, 23
T_VarExp
 MM, 23
T_Vector
 MM, 22
T_Violation
 MM, 22
TO
 mm.tab.cpp, 475, 476
 mm.tab.hpp, 487, 489

TRIGGER
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489

TRUE
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 487, 489

TID
 MM, 22

tList
 YYSTYPE, 400

TRIGGER_CHAR
 MM::TriggerExp, 375

TYPE_STR
 MM::Recycler, 327

toString
 MM::Activation, 27
 MM::ActiveExp, 31
 MM::AliasExp, 35
 MM::AllExp, 38
 MM::Assertion, 42
 MM::BinExp, 46
 MM::BooleanValExp, 50
 MM::ConverterNodeBehavior, 55, 56
 MM::Declaration, 62
 MM::Definition, 72
 MM::Deletion, 78
 MM::DieExp, 83
 MM::Disablement, 86
 MM::DrainNodeBehavior, 92
 MM::Edge, 98
 MM::Element, 106
 MM::Enablement, 110
 MM::Evaluator, 121

MM::Event, 126
MM::Exp, 131
MM::Failure, 135
MM::FlowEdge, 137, 138
MM::FlowEvent, 142
MM::GateNodeBehavior, 147, 148
MM::Instance, 170, 171
MM::InterfaceNode, 178
MM::Location, 187
MM::Modification, 232
MM::Name, 241
MM::Node, 256
MM::NodeBehavior, 268
MM::NumberValExp, 277
MM::OneExp, 286
MM::OverrideExp, 292
MM::PoolNodeBehavior, 306
MM::Prevention, 309
MM::Program, 312
MM::RangeValExp, 317
MM::Recyclable, 322
MM::RefNodeBehavior, 338, 339
MM::SourceNodeBehavior, 344
MM::StateEdge, 348
MM::String, 357
MM::Transformation, 362
MM::Transition, 366
MM::TriggerEvent, 370
MM::TriggerExp, 374
MM::UnExp, 378
MM::ValExp, 383
MM::VarExp, 387
MM::Violation, 395
Transformation, 358
 MM::Transformation, 359
Transition, 367
 MM::Transition, 365
TriggerEvent, 367
 MM::TriggerEvent, 369
TriggerExp, 370
 MM::TriggerExp, 372
Types.h
 MM_FALSE, 528
 MM_MAX_RESOURCES, 528
 MM_NULL, 528
 MM_TRUE, 528
 MM_printf, 528
UNM
 mm.tab.cpp, 476, 477
 mm.tab.hpp, 488, 490
USER
 mm.tab.cpp, 475, 476
 mm.tab.hpp, 487, 489
UINT16_MAX
 lex.mm.cpp, 443
UINT32
 MM, 22
UINT32_MAX
 lex.mm.cpp, 443
 MM, 22
UINT8_MAX
 lex.mm.cpp, 443
UnExp, 375
 MM::UnExp, 376
uncreate
 MM::Recycler, 326, 327
unput
 lex.mm.cpp, 443
update
 MM::Declaration, 62
 MM::Instance, 171
 MM::Machine::Delegate, 76
 MM::Machine::InstanceObserver, 174
 MM::Observer, 283
 MM::PoolNodeBehavior, 307
VIOLATE
 mm.tab.cpp, 475, 477
 mm.tab.hpp, 488, 489
VIOLATE_LEN
 MM::Violation, 396
VIOLATE_STR
 MM::Violation, 396
VOID
 MM, 22
val
 YYSTYPE, 400
ValExp, 383
 MM::ValExp, 380
VarExp, 384
 MM::VarExp, 385
Vector
 MM::Vector, 388
Violation, 396
 MM::Violation, 394, 395
visible
 MM::Element, 107
WHEN_AUTO
 MM::NodeBehavior, 260
WHEN_ERROR
 MM::NodeBehavior, 260
WHEN_PASSIVE
 MM::NodeBehavior, 260
WHEN_START
 MM::NodeBehavior, 260
WHEN_USER
 MM::NodeBehavior, 260
WHEN_LEN
 MM::NodeBehavior, 270
WHEN_STR
 MM::NodeBehavior, 270
When
 MM::NodeBehavior, 259
when
 YYSTYPE, 400

while
 lex.mm.cpp, 447
 mm.tab.cpp, 482

YY_
 mm.tab.cpp, 471

YY_AT_BOL
 lex.mm.cpp, 443

YY_BREAK
 lex.mm.cpp, 443

YY_BUF_SIZE
 lex.mm.cpp, 443

YY_BUFFER_NEW
 lex.mm.cpp, 443

YY_BUFFER_NORMAL
 lex.mm.cpp, 443

YY_BUFFER_STATE
 lex.mm.cpp, 446
 mm.tab.cpp, 474

YY_CHAR
 lex.mm.cpp, 447

YY_CURRENT_BUFFER
 lex.mm.cpp, 443

YY_DECL
 lex.mm.cpp, 443, 456

YY_DECL_IS_OURS
 lex.mm.cpp, 443

YY_END_OF_BUFFER
 lex.mm.cpp, 443

YY_EXIT_FAILURE
 lex.mm.cpp, 443

YY_EXTRA_TYPE
 lex.mm.cpp, 444

YY_FATAL_ERROR
 lex.mm.cpp, 444

YY_FLUSH_BUFFER
 lex.mm.cpp, 444

YY_INPUT
 lex.mm.cpp, 444

YY_INT_ALIGNED
 lex.mm.cpp, 444

YY_LESS_LINENO
 lex.mm.cpp, 444

YY_LOCATION_PRINT
 mm.tab.cpp, 471

YY_MORE_ADJ
 lex.mm.cpp, 444

YY_NEW_FILE
 lex.mm.cpp, 444

YY_NULL
 lex.mm.cpp, 444

YY_NUM_RULES
 lex.mm.cpp, 444

YY_READ_BUF_SIZE
 lex.mm.cpp, 444

YY_REDUCE_PRINT
 mm.tab.cpp, 471

YY_RULE_SETUP
 lex.mm.cpp, 444

YY_SC_TO_UI
 lex.mm.cpp, 445

YY_SKIP_YYWRAP
 lex.mm.cpp, 445

YY_STACK_PRINT
 mm.tab.cpp, 471

YY_START
 lex.mm.cpp, 445

YY_STATE_BUF_SIZE
 lex.mm.cpp, 445

YY_STATE_EOF
 lex.mm.cpp, 445

YY_SYMBOL_PRINT
 mm.tab.cpp, 471, 482

YY_USER_ACTION
 lex.mm.cpp, 445

YYABORT
 mm.tab.cpp, 471

YYACCEPT
 mm.tab.cpp, 471

YYBACKUP
 mm.tab.cpp, 471

YYBISON
 mm.tab.cpp, 471

YYBISON_VERSION
 mm.tab.cpp, 472

YYCOPY
 mm.tab.cpp, 472

YYDEBUG
 mm.tab.cpp, 472

YYDPRINTF
 mm.tab.cpp, 472

YYEMPTY
 mm.tab.cpp, 472

YYEOF
 mm.tab.cpp, 472

YYERRCODE
 mm.tab.cpp, 472

YYERROR
 mm.tab.cpp, 472

YYERROR_VERBOSE
 mm.tab.cpp, 472

YYFAIL
 mm.tab.cpp, 472

YYFINAL
 mm.tab.cpp, 472

YYFREE
 mm.tab.cpp, 472

YYID
 mm.tab.cpp, 472

YYINITDEPTH
 mm.tab.cpp, 472

YYLAST
 mm.tab.cpp, 472

YYLEX
 mm.tab.cpp, 472

YYLLOC_DEFAULT
 mm.tab.cpp, 472

YYLSP_NEEDED
 mm.tab.cpp, 473
YYLTYPE, 398
 first_column, 398
 first_line, 398
 last_column, 399
 last_line, 399
 mm.tab.hpp, 487
 YYLTYPE.h, 535
YYLTYPE.h
 YYLTYPE, 535
YYLTYPE_IS_DECLARED
 mm.tab.hpp, 487
YYLTYPE_IS_TRIVIAL
 mm.tab.hpp, 487
YYMALLOC
 mm.tab.cpp, 473
YYMAXDEPTH
 mm.tab.cpp, 473
YYMAXUTOK
 mm.tab.cpp, 473
YYNNTS
 mm.tab.cpp, 473
YYNRULES
 mm.tab.cpp, 473
YYNSTATES
 mm.tab.cpp, 473
YYNTOKENS
 mm.tab.cpp, 473
YYPACT_NINF
 mm.tab.cpp, 473
YYPOPSTACK
 mm.tab.cpp, 473
YYPULL
 mm.tab.cpp, 473
YYPURE
 mm.tab.cpp, 473
YYPUSH
 mm.tab.cpp, 473
YYRECOVERING
 mm.tab.cpp, 473
YYRHSLOC
 mm.tab.cpp, 473
YYSIZE_MAXIMUM
 mm.tab.cpp, 473
YYSIZE_T
 mm.tab.cpp, 473
YYSKELETON_NAME
 mm.tab.cpp, 473
YYSTACK_ALLOC
 mm.tab.cpp, 473
YYSTACK_BYTES
 mm.tab.cpp, 473
YYSTACK_FREE
 mm.tab.cpp, 473
YYSTACK_GAP_MAXIMUM
 mm.tab.cpp, 474
YYSTACK_RELOCATE
 mm.tab.cpp, 474
 mm.tab.cpp, 474
YYSTATE
 lex.mm.cpp, 446
YYSTYPE, 399
 act, 399
 eList, 400
 element, 399
 exp, 400
 how, 400
 io, 400
 mm.tab.cpp, 474
 mm.tab.hpp, 487
 name, 400
 program, 400
 str, 400
 t, 400
 tList, 400
 val, 400
 when, 400
YYSTYPE_IS_DECLARED
 mm.tab.cpp, 474
 mm.tab.hpp, 487
YYSTYPE_IS_TRIVIAL
 mm.tab.cpp, 474
 mm.tab.hpp, 487
YYTABLE_NINF
 mm.tab.cpp, 474
YYTABLES_NAME
 lex.mm.cpp, 446
YYTERROR
 mm.tab.cpp, 474
YYTOKEN_TABLE
 mm.tab.cpp, 474
YYTOKENTYPE
 mm.tab.cpp, 474
YYTRANSLATE
 mm.tab.cpp, 474
YYUNDEFTOK
 mm.tab.cpp, 474
YYUSE
 mm.tab.cpp, 474, 485
yy_act
 lex.mm.cpp, 455
yy_at_bol
 yy_buffer_state, 397
yy_bp
 lex.mm.cpp, 455
yy_bs_column
 yy_buffer_state, 397
yy_bs_lineno
 yy_buffer_state, 397
yy_buf_pos
 yy_buffer_state, 397
yy_buf_size
 yy_buffer_state, 397
yy_buffer_state
 yy_at_bol, 397
 yy_bs_column, 397

yy_bs_lineno, 397
 yy_buf_pos, 397
 yy_buf_size, 397
 yy_buffer_status, 397
 yy_ch_buf, 397
 yy_fill_buffer, 397
 yy_input_file, 397
 yy_is_interactive, 397
 yy_is_our_buffer, 397
 yy_n_chars, 397
 yy_buffer_status
 yy_buffer_state, 397
 yy_ch_buf
 yy_buffer_state, 397
 yy_cp
 lex.mm.cpp, 455
 yy_create_buffer
 lex.mm.cpp, 447
 yy_delete_buffer
 lex.mm.cpp, 448
 yy_fill_buffer
 yy_buffer_state, 397
 yy flex_debug
 lex.mm.cpp, 456
 yy_flush_buffer
 lex.mm.cpp, 448
 yy_input_file
 yy_buffer_state, 397
 yy_is_interactive
 yy_buffer_state, 397
 yy_is_our_buffer
 yy_buffer_state, 397
 yy_n_chars
 yy_buffer_state, 397
 yy_new_buffer
 lex.mm.cpp, 444
 yy_nxt
 yy_trans_info, 397
 yy_scan_buffer
 lex.mm.cpp, 449
 mm.tab.cpp, 482
 yy_scan_bytes
 lex.mm.cpp, 449
 yy_scan_string
 lex.mm.cpp, 450
 yy_set_bol
 lex.mm.cpp, 445
 yy_set_interactive
 lex.mm.cpp, 445
 yy_size_t
 lex.mm.cpp, 447
 yy_state_type
 lex.mm.cpp, 447
 yy_switch_to_buffer
 lex.mm.cpp, 451
 yy_trans_info, 397
 yy_nxt, 397
 yy_verify, 397
 yy_verify
 yy_trans_info, 397
 yyalloc, 398
 lex.mm.cpp, 451
 yyls_alloc, 398
 yyss_alloc, 398
 yyvs_alloc, 398
 yychar
 mm.tab.cpp, 485
 yyclearin
 mm.tab.cpp, 472
 yycolumn
 lex.mm.cpp, 456
 mm.tab.cpp, 485
 yyconst
 lex.mm.cpp, 445
 yyd
 mm.tab.cpp, 485
 yyerrok
 mm.tab.cpp, 472
 yyerror
 mm.tab.cpp, 483
 yyfree
 lex.mm.cpp, 451
 yyget_debug
 lex.mm.cpp, 452
 yyget_extra
 lex.mm.cpp, 452
 yyget_in
 lex.mm.cpp, 452
 yyget_leng
 lex.mm.cpp, 452
 yyget_lineno
 lex.mm.cpp, 452
 yyget_out
 lex.mm.cpp, 452
 yyget_text
 lex.mm.cpp, 452
 yyid
 lex.mm.cpp, 456
 yyidpos
 lex.mm.cpp, 456
 yyin
 lex.mm.cpp, 456
 mm.tab.cpp, 485
 yylen
 mm.tab.cpp, 485
 yy leng
 lex.mm.cpp, 456
 yyless
 lex.mm.cpp, 446
 yylex
 lex.mm.cpp, 452
 mm.tab.cpp, 483
 yylex_destroy
 lex.mm.cpp, 452
 yylineno
 lex.mm.cpp, 456

mm.tab.cpp, 485
yyloc
 mm.tab.cpp, 485
 mm.tab.hpp, 490
yylocationp
 mm.tab.cpp, 485
yyls_alloc
 yyalloc, 398
yyltpe
 mm.tab.hpp, 487
yylval
 mm.tab.cpp, 485
 mm.tab.hpp, 490
yymore
 lex.mm.cpp, 446
yynerrs
 mm.tab.cpp, 485
yyout
 lex.mm.cpp, 456
yyparse
 mm.tab.cpp, 483
yypop_buffer_state
 lex.mm.cpp, 453
yypush_buffer_state
 lex.mm.cpp, 453
yyrealloc
 lex.mm.cpp, 453
yyrestart
 lex.mm.cpp, 453
yys
 mm.tab.cpp, 485
yyset_debug
 lex.mm.cpp, 455
yyset_extra
 lex.mm.cpp, 455
yyset_in
 lex.mm.cpp, 455
yyset_lineno
 lex.mm.cpp, 455
yyset_out
 lex.mm.cpp, 455
yysrc
 mm.tab.cpp, 485
yyss_alloc
 yyalloc, 398
yystr
 lex.mm.cpp, 456
yystype
 mm.tab.cpp, 474
 mm.tab.hpp, 487
yyterminate
 lex.mm.cpp, 446
yytext
 lex.mm.cpp, 456
yytext_ptr
 lex.mm.cpp, 446
yytokentype
 mm.tab.cpp, 474