

Interpreter

Generated by Doxygen 1.8.2

Tue Sep 25 2012 00:33:12

Contents

1	Main Page	1
2	Todo List	3
3	Module Index	5
3.1	Modules	5
4	Hierarchical Index	7
4.1	Class Hierarchy	7
5	Class Index	9
5.1	Class List	9
6	File Index	11
6.1	File List	11
7	Module Documentation	15
7.1	User-Level API	15
8	Class Documentation	17
8.1	BoxVolume Class Reference	17
8.1.1	Detailed Description	20
8.1.2	Constructor & Destructor Documentation	20
8.1.2.1	BoxVolume	20
8.1.2.2	~BoxVolume	20
8.1.3	Member Function Documentation	21
8.1.3.1	copy	21
8.1.3.2	Explode	21
8.1.3.3	get	21
8.1.3.4	get	21
8.1.3.5	getBoxVolumeID	22

8.1.3.6	getdao	22
8.1.3.7	gethasBoxVolume_MaximumPoint	22
8.1.3.8	gethasBoxVolume_MinimumPoint	22
8.1.3.9	gethasRobot_WorkVolume	22
8.1.3.10	gethasWorkstation_OtherObstacles	22
8.1.3.11	set	22
8.1.3.12	set	22
8.1.3.13	setdao	23
8.1.3.14	sethasBoxVolume_MaximumPoint	23
8.1.3.15	sethasBoxVolume_MinimumPoint	23
8.1.3.16	sethasRobot_WorkVolume	23
8.1.3.17	sethasWorkstation_OtherObstacles	23
8.1.4	Member Data Documentation	23
8.1.4.1	BoxVolumeID	23
8.1.4.2	dao	23
8.1.4.3	hasBoxVolume_MaximumPoint	24
8.1.4.4	hasBoxVolume_MinimumPoint	24
8.1.4.5	hasRobot_WorkVolume	24
8.1.4.6	hasWorkstation_OtherObstacles	24
8.2	BoxyObject Class Reference	24
8.2.1	Detailed Description	27
8.2.2	Constructor & Destructor Documentation	27
8.2.2.1	BoxyObject	27
8.2.2.2	~BoxyObject	27
8.2.3	Member Function Documentation	27
8.2.3.1	copy	27
8.2.3.2	Explode	28
8.2.3.3	get	28
8.2.3.4	get	28
8.2.3.5	getBoxyObjectID	29
8.2.3.6	getdao	29
8.2.3.7	gethasBox_Height	29
8.2.3.8	gethasBox_Length	29
8.2.3.9	gethasBox_Width	29
8.2.3.10	set	29
8.2.3.11	set	30
8.2.3.12	setdao	30

8.2.3.13	sethasBox_Height	30
8.2.3.14	sethasBox_Length	30
8.2.3.15	sethasBox_Width	30
8.2.4	Member Data Documentation	30
8.2.4.1	BoxyObjectID	30
8.2.4.2	dao	30
8.2.4.3	hasBox_Height	30
8.2.4.4	hasBox_Length	31
8.2.4.5	hasBox_Width	31
8.3	CanonicalHdr Struct Reference	31
8.3.1	Detailed Description	31
8.3.2	Member Data Documentation	31
8.3.2.1	msgID	31
8.3.2.2	msgType	32
8.3.2.3	time	32
8.4	CanonicalMsg Class Reference	32
8.4.1	Detailed Description	33
8.4.2	Constructor & Destructor Documentation	33
8.4.2.1	CanonicalMsg	33
8.4.3	Member Function Documentation	33
8.4.3.1	getMsgID	33
8.4.3.2	getTime	33
8.4.3.3	getType	34
8.4.3.4	setHeader	34
8.4.4	Member Data Documentation	35
8.4.4.1	hdr	35
8.4.4.2	nextID	35
8.5	CanonicalRobotCommand Class Reference	35
8.5.1	Detailed Description	38
8.5.2	Constructor & Destructor Documentation	38
8.5.2.1	CanonicalRobotCommand	38
8.5.2.2	~CanonicalRobotCommand	38
8.5.3	Member Function Documentation	38
8.5.3.1	actionInterpreter	38
8.5.3.2	attach_eff	39
8.5.3.3	canon_put_part	40
8.5.3.4	canon_take_part	41

8.5.3.5	create_kit	41
8.5.3.6	getKitTrayLocation	42
8.5.3.7	getPartLocation	44
8.5.3.8	getPartsTrayLocation	45
8.5.3.9	interpretPlan	47
8.5.3.10	print_closegripper	48
8.5.3.11	print_dwell	49
8.5.3.12	print_endcannon	49
8.5.3.13	print_initcannon	50
8.5.3.14	print_moveto	51
8.5.3.15	print.opengripper	52
8.5.3.16	put_kit	53
8.5.3.17	put_kit_tray	53
8.5.3.18	put_part	53
8.5.3.19	remove_eff	54
8.5.3.20	take_kit	55
8.5.3.21	take_kit_tray	55
8.5.3.22	take_part	56
8.5.4	Member Data Documentation	57
8.5.4.1	m_dwell	57
8.5.4.2	m_kit_tray	57
8.5.4.3	m_safe_z	57
8.6	CloseGripperMsg Class Reference	57
8.6.1	Detailed Description	59
8.6.2	Constructor & Destructor Documentation	59
8.6.2.1	CloseGripperMsg	60
8.6.2.2	~CloseGripperMsg	60
8.7	Connection Class Reference	60
8.7.1	Detailed Description	61
8.7.2	Constructor & Destructor Documentation	61
8.7.2.1	Connection	61
8.7.2.2	~Connection	61
8.7.3	Member Function Documentation	61
8.7.3.1	getCon	61
8.7.3.2	getDriver	62
8.7.3.3	setCon	63
8.7.3.4	setDriver	63

8.7.4	Member Data Documentation	63
8.7.4.1	con	63
8.7.4.2	driver	63
8.8	Controller Class Reference	63
8.8.1	Detailed Description	66
8.8.2	Constructor & Destructor Documentation	66
8.8.2.1	Controller	66
8.8.2.2	~Controller	66
8.8.3	Member Function Documentation	66
8.8.3.1	dequeueMsg	66
8.8.3.2	processCloseGripper	67
8.8.3.3	processDwell	68
8.8.3.4	processEndCanon	68
8.8.3.5	processInitCanon	68
8.8.3.6	processMessage	69
8.8.3.7	processMoveSmoothlyTo	69
8.8.3.8	processMoveStraightTo	69
8.8.3.9	processMoveTo	70
8.8.3.10	processOpenGripper	70
8.8.3.11	processSetAbsoluteAcceleration	70
8.8.3.12	processSetAngleUnits	71
8.8.3.13	processSetEndAngleTolerance	71
8.8.3.14	processSetEndPointTolerance	71
8.8.3.15	processSetIntermediatePointTolerance	72
8.8.3.16	processSetLengthUnits	72
8.8.3.17	processSetRelativeAcceleration	72
8.8.3.18	processSetRelativeSpeed	73
8.8.3.19	queueCloseGripper	73
8.8.3.20	queueDwell	73
8.8.3.21	queueMsg	73
8.8.4	Member Data Documentation	74
8.8.4.1	cmdQueue	74
8.8.4.2	queue_length	74
8.9	DAO Class Reference	74
8.9.1	Detailed Description	76
8.9.2	Constructor & Destructor Documentation	76
8.9.2.1	DAO	76

8.9.2.2	~DAO	77
8.9.3	Member Function Documentation	77
8.9.3.1	Explode	77
8.9.3.2	fillGetSqlQueries	78
8.9.3.3	get	79
8.9.3.4	getclassName	80
8.9.3.5	getconnection	81
8.9.3.6	set	81
8.9.3.7	setclassName	82
8.9.3.8	setconnection	83
8.9.4	Member Data Documentation	83
8.9.4.1	className	83
8.9.4.2	connection	83
8.9.4.3	getSqlQueriesDataMulti	83
8.9.4.4	getSqlQueriesDataSingle	83
8.9.4.5	getSqlQueriesObjectMulti	83
8.9.4.6	getSqlQueriesObjectSingle	83
8.9.4.7	map	83
8.9.4.8	nameDone	83
8.9.4.9	path	83
8.9.4.10	pathmulti	83
8.9.4.11	setSqlQueries	84
8.9.4.12	updateSqlQueries	84
8.10	DataThing Class Reference	84
8.10.1	Detailed Description	86
8.10.2	Constructor & Destructor Documentation	86
8.10.2.1	DataThing	86
8.10.2.2	~DataThing	86
8.10.3	Member Function Documentation	86
8.10.3.1	copy	86
8.10.3.2	Explode	87
8.10.3.3	get	88
8.10.3.4	get	88
8.10.3.5	getdao	88
8.10.3.6	getDataThingID	88
8.10.3.7	getname	88
8.10.3.8	set	90

8.10.3.9	set	91
8.10.3.10	setdao	91
8.10.4	Member Data Documentation	91
8.10.4.1	dao	91
8.10.4.2	DataThingID	91
8.10.4.3	name	91
8.11	DwellMsg Class Reference	91
8.11.1	Detailed Description	94
8.11.2	Constructor & Destructor Documentation	94
8.11.2.1	DwellMsg	94
8.11.2.2	DwellMsg	94
8.11.2.3	~DwellMsg	94
8.11.3	Member Data Documentation	94
8.11.3.1	time	94
8.12	EndCanonMsg Class Reference	94
8.12.1	Detailed Description	97
8.12.2	Constructor & Destructor Documentation	97
8.12.2.1	EndCanonMsg	97
8.12.2.2	EndCanonMsg	97
8.12.2.3	~EndCanonMsg	97
8.12.3	Member Data Documentation	97
8.12.3.1	reason	97
8.13	EndEffector Class Reference	97
8.13.1	Detailed Description	100
8.13.2	Constructor & Destructor Documentation	100
8.13.2.1	EndEffector	100
8.13.2.2	~EndEffector	100
8.13.3	Member Function Documentation	101
8.13.3.1	copy	101
8.13.3.2	Explode	101
8.13.3.3	get	101
8.13.3.4	get	101
8.13.3.5	getdao	102
8.13.3.6	getEndEffectorID	102
8.13.3.7	gethadByEndEffector_Robot	102
8.13.3.8	gethasEffector_MaximumLoadWeight	102
8.13.3.9	gethasEndEffector_Description	102

8.13.3.10 gethasEndEffector_Id	102
8.13.3.11 gethasEndEffector_Weight	102
8.13.3.12 gethasEndEffectorHolder_EndEffector	102
8.13.3.13 set	103
8.13.3.14 set	103
8.13.3.15 setdao	103
8.13.3.16 sethadByEndEffector_Robot	104
8.13.3.17 sethasEffector_MaximumLoadWeight	104
8.13.3.18 sethasEndEffector_Description	104
8.13.3.19 sethasEndEffector_Id	104
8.13.3.20 sethasEndEffector_Weight	104
8.13.3.21 sethasEndEffectorHolder_EndEffector	104
8.13.4 Member Data Documentation	104
8.13.4.1 dao	104
8.13.4.2 EndEffectorID	104
8.13.4.3 hadByEndEffector_Robot	104
8.13.4.4 hasEffector_MaximumLoadWeight	104
8.13.4.5 hasEndEffector_Description	104
8.13.4.6 hasEndEffector_Id	105
8.13.4.7 hasEndEffector_Weight	105
8.13.4.8 hasEndEffectorHolder_EndEffector	105
8.14 EndEffectorChangingStation Class Reference	105
8.14.1 Detailed Description	108
8.14.2 Constructor & Destructor Documentation	108
8.14.2.1 EndEffectorChangingStation	108
8.14.2.2 ~EndEffectorChangingStation	108
8.14.3 Member Function Documentation	108
8.14.3.1 copy	108
8.14.3.2 Explode	109
8.14.3.3 get	109
8.14.3.4 get	109
8.14.3.5 getdao	110
8.14.3.6 getEndEffectorChangingStationID	110
8.14.3.7 gethadByEndEffectorHolder_ChangingStation	110
8.14.3.8 gethasWorkstation_ChangingStation	110
8.14.3.9 set	110
8.14.3.10 set	110

8.14.3.11 setdao	111
8.14.3.12 sethadByEndEffectorHolder_ChangingStation	111
8.14.3.13 sethasWorkstation_ChangingStation	111
8.14.4 Member Data Documentation	111
8.14.4.1 dao	111
8.14.4.2 EndEffectorChangingStationID	111
8.14.4.3 hadByEndEffectorHolder_ChangingStation	111
8.14.4.4 hasWorkstation_ChangingStation	111
8.15 EndEffectorHolder Class Reference	111
8.15.1 Detailed Description	114
8.15.2 Constructor & Destructor Documentation	114
8.15.2.1 EndEffectorHolder	114
8.15.2.2 ~EndEffectorHolder	114
8.15.3 Member Function Documentation	114
8.15.3.1 copy	114
8.15.3.2 Explode	115
8.15.3.3 get	115
8.15.3.4 get	115
8.15.3.5 getdao	115
8.15.3.6 getEndEffectorHolderID	115
8.15.3.7 gethadByEndEffectorHolder_ChangingStation	115
8.15.3.8 gethasEndEffectorHolder_EndEffector	116
8.15.3.9 set	116
8.15.3.10 set	116
8.15.3.11 setdao	116
8.15.3.12 sethadByEndEffectorHolder_ChangingStation	116
8.15.3.13 sethasEndEffectorHolder_EndEffector	116
8.15.4 Member Data Documentation	116
8.15.4.1 dao	116
8.15.4.2 EndEffectorHolderID	116
8.15.4.3 hadByEndEffectorHolder_ChangingStation	117
8.15.4.4 hasEndEffectorHolder_EndEffector	117
8.16 FileOperator Class Reference	117
8.16.1 Detailed Description	118
8.16.2 Constructor & Destructor Documentation	119
8.16.2.1 FileOperator	119
8.16.2.2 ~FileOperator	119

8.16.3 Member Function Documentation	119
8.16.3.1 copyVector	119
8.16.3.2 createOutputFile	119
8.16.3.3 getCanonFile	120
8.16.3.4 openFile	120
8.16.3.5 readMap	120
8.16.3.6 readMapOfVector	120
8.16.3.7 readVector	121
8.16.3.8 readVectorOfVector	121
8.16.3.9 removeDuplicates	121
8.16.3.10 removeParentheses	121
8.16.3.11 splitString	122
8.16.3.12 stripSpace	122
8.16.3.13 writeData	122
8.17 GripperEffector Class Reference	123
8.17.1 Detailed Description	126
8.17.2 Constructor & Destructor Documentation	126
8.17.2.1 GripperEffector	126
8.17.2.2 ~GripperEffector	126
8.17.3 Member Function Documentation	126
8.17.3.1 copy	126
8.17.3.2 Explode	127
8.17.3.3 get	127
8.17.3.4 get	127
8.17.3.5 getdao	127
8.17.3.6 getGripperEffectorID	127
8.17.3.7 set	127
8.17.3.8 set	127
8.17.3.9 setdao	128
8.17.4 Member Data Documentation	128
8.17.4.1 dao	128
8.17.4.2 GripperEffectorID	128
8.18 IniFile Class Reference	128
8.18.1 Detailed Description	129
8.18.2 Constructor & Destructor Documentation	130
8.18.2.1 IniFile	130
8.18.2.2 IniFile	130

8.18.2.3 ~IniFile	130
8.18.2.4 IniFile	130
8.18.3 Member Function Documentation	130
8.18.3.1 close	130
8.18.3.2 find	130
8.18.3.3 findSection	130
8.18.3.4 iniFind	131
8.18.3.5 iniSection	131
8.18.3.6 isSection	131
8.18.3.7 open	131
8.18.3.8 operator=	131
8.18.3.9 section	131
8.18.3.10 valid	131
8.19 INIFILE_ENTRY Struct Reference	132
8.19.1 Detailed Description	132
8.19.2 Member Data Documentation	132
8.19.2.1 rest	132
8.19.2.2 tag	132
8.20 InitCanonMsg Class Reference	133
8.20.1 Detailed Description	134
8.20.2 Constructor & Destructor Documentation	134
8.20.2.1 InitCanonMsg	135
8.20.2.2 ~InitCanonMsg	135
8.21 Kit Class Reference	135
8.21.1 Detailed Description	138
8.21.2 Constructor & Destructor Documentation	138
8.21.2.1 Kit	138
8.21.2.2 ~Kit	138
8.21.3 Member Function Documentation	139
8.21.3.1 copy	139
8.21.3.2 Explode	139
8.21.3.3 get	140
8.21.3.4 get	140
8.21.3.5 getdao	140
8.21.3.6 gethadByKit_LargeBoxWithKits	140
8.21.3.7 gethadByPart_Kit	140
8.21.3.8 gethasKit_DesignRef	140

8.21.3.9 <code>gethasKit_Tray</code>	141
8.21.3.10 <code>getisKit_Finished</code>	141
8.21.3.11 <code>getKitID</code>	141
8.21.3.12 <code>set</code>	141
8.21.3.13 <code>set</code>	141
8.21.3.14 <code>setdao</code>	141
8.21.3.15 <code>sethadByKit_LargeBoxWithKits</code>	142
8.21.3.16 <code>sethadByPart_Kit</code>	142
8.21.3.17 <code>sethasKit_DesignRef</code>	142
8.21.3.18 <code>sethasKit_Tray</code>	142
8.21.3.19 <code>setisKit_Finished</code>	142
8.21.4 Member Data Documentation	142
8.21.4.1 <code>dao</code>	142
8.21.4.2 <code>hadByKit_LargeBoxWithKits</code>	142
8.21.4.3 <code>hadByPart_Kit</code>	142
8.21.4.4 <code>hasKit_DesignRef</code>	142
8.21.4.5 <code>hasKit_Tray</code>	142
8.21.4.6 <code>isKit_Finished</code>	142
8.21.4.7 <code>KitID</code>	143
8.22 KitDesign Class Reference	143
8.22.1 Detailed Description	146
8.22.2 Constructor & Destructor Documentation	146
8.22.2.1 <code>KitDesign</code>	146
8.22.2.2 <code>~KitDesign</code>	146
8.22.3 Member Function Documentation	147
8.22.3.1 <code>copy</code>	147
8.22.3.2 <code>Explode</code>	147
8.22.3.3 <code>get</code>	148
8.22.3.4 <code>get</code>	148
8.22.3.5 <code>getdao</code>	148
8.22.3.6 <code>gethadByKitDesign_Workstation</code>	148
8.22.3.7 <code>gethadByPartRefAndPose_KitDesign</code>	148
8.22.3.8 <code>gethasKitDesign_Id</code>	149
8.22.3.9 <code>gethasKitDesign_KitTraySkuRef</code>	149
8.22.3.10 <code>getKitDesignID</code>	149
8.22.3.11 <code>set</code>	149
8.22.3.12 <code>set</code>	149

8.22.3.13 setdao	149
8.22.3.14 sethadByKitDesign_Workstation	149
8.22.3.15 sethadByPartRefAndPose_KitDesign	149
8.22.3.16 sethasKitDesign_Id	150
8.22.3.17 sethasKitDesign_KitTraySkuRef	150
8.22.4 Member Data Documentation	150
8.22.4.1 dao	150
8.22.4.2 hadByKitDesign_Workstation	150
8.22.4.3 hadByPartRefAndPose_KitDesign	150
8.22.4.4 hasKitDesign_Id	150
8.22.4.5 hasKitDesign_KitTraySkuRef	150
8.22.4.6 KitDesignID	150
8.23 KittingPDDLProblem Class Reference	150
8.23.1 Detailed Description	151
8.23.2 Constructor & Destructor Documentation	152
8.23.2.1 KittingPDDLProblem	152
8.23.2.2 ~KittingPDDLProblem	152
8.23.3 Member Function Documentation	152
8.23.3.1 findParam	152
8.23.3.2 findParamType	153
8.23.3.3 parsePDDLProblem	154
8.24 KittingPlan Class Reference	155
8.24.1 Detailed Description	155
8.25 KittingWorkstation Class Reference	156
8.25.1 Detailed Description	159
8.25.2 Constructor & Destructor Documentation	160
8.25.2.1 KittingWorkstation	160
8.25.2.2 ~KittingWorkstation	160
8.25.3 Member Function Documentation	160
8.25.3.1 copy	160
8.25.3.2 Explode	160
8.25.3.3 get	161
8.25.3.4 get	161
8.25.3.5 getdao	161
8.25.3.6 gethadByKitDesign_Workstation	161
8.25.3.7 gethadBySku_Workstation	161
8.25.3.8 gethasWorkstation_AngleUnit	161

8.25.3.9 gethasWorkstation_ChangingStation	162
8.25.3.10 gethasWorkstation_LengthUnit	162
8.25.3.11 gethasWorkstation_OtherObstacles	162
8.25.3.12 gethasWorkstation_Robot	162
8.25.3.13 gethasWorkstation_WeightUnit	162
8.25.3.14 gethasWorkstation_WorkTable	162
8.25.3.15 getKittingWorkstationID	162
8.25.3.16 set	162
8.25.3.17 set	162
8.25.3.18 setdao	163
8.25.3.19 sethadByKitDesign_Workstation	163
8.25.3.20 sethadBySku_Workstation	163
8.25.3.21 sethasWorkstation_AngleUnit	163
8.25.3.22 sethasWorkstation_ChangingStation	163
8.25.3.23 sethasWorkstation_LengthUnit	163
8.25.3.24 sethasWorkstation_OtherObstacles	163
8.25.3.25 sethasWorkstation_Robot	163
8.25.3.26 sethasWorkstation_WeightUnit	163
8.25.3.27 sethasWorkstation_WorkTable	163
8.25.4 Member Data Documentation	164
8.25.4.1 dao	164
8.25.4.2 hadByKitDesign_Workstation	164
8.25.4.3 hadBySku_Workstation	164
8.25.4.4 hasWorkstation_AngleUnit	164
8.25.4.5 hasWorkstation_ChangingStation	164
8.25.4.6 hasWorkstation_LengthUnit	164
8.25.4.7 hasWorkstation_OtherObstacles	164
8.25.4.8 hasWorkstation_Robot	164
8.25.4.9 hasWorkstation_WeightUnit	164
8.25.4.10 hasWorkstation_WorkTable	164
8.25.4.11 KittingWorkstationID	164
8.26 KitTray Class Reference	165
8.26.1 Detailed Description	168
8.26.2 Constructor & Destructor Documentation	168
8.26.2.1 KitTray	168
8.26.2.2 ~KitTray	168
8.26.3 Member Function Documentation	169

8.26.3.1	copy	169
8.26.3.2	Explode	169
8.26.3.3	get	169
8.26.3.4	get	169
8.26.3.5	getdao	170
8.26.3.6	gethadByKitTray_LargeBoxWithEmptyKitTrays	170
8.26.3.7	gethasKit_Tray	170
8.26.3.8	gethasKitTray_SerialNumber	170
8.26.3.9	gethasKitTray_SkuRef	170
8.26.3.10	getKitTrayID	170
8.26.3.11	set	170
8.26.3.12	set	170
8.26.3.13	setdao	171
8.26.3.14	sethadByKitTray_LargeBoxWithEmptyKitTrays	171
8.26.3.15	sethasKit_Tray	171
8.26.3.16	sethasKitTray_SerialNumber	171
8.26.3.17	sethasKitTray_SkuRef	171
8.26.4	Member Data Documentation	171
8.26.4.1	dao	171
8.26.4.2	hadByKitTray_LargeBoxWithEmptyKitTrays	172
8.26.4.3	hasKit_Tray	172
8.26.4.4	hasKitTray_SerialNumber	172
8.26.4.5	hasKitTray_SkuRef	172
8.26.4.6	KitTrayID	172
8.27	KitTrayLocStruct Struct Reference	172
8.27.1	Detailed Description	174
8.27.2	Member Data Documentation	174
8.27.2.1	point	174
8.27.2.2	x_axis	174
8.27.2.3	z_axis	174
8.28	LargeBoxWithEmptyKitTrays Class Reference	174
8.28.1	Detailed Description	177
8.28.2	Constructor & Destructor Documentation	177
8.28.2.1	LargeBoxWithEmptyKitTrays	177
8.28.2.2	~LargeBoxWithEmptyKitTrays	177
8.28.3	Member Function Documentation	177
8.28.3.1	copy	177

8.28.3.2	Explode	178
8.28.3.3	get	178
8.28.3.4	get	178
8.28.3.5	getdao	179
8.28.3.6	gethadByKitTray_LargeBoxWithEmptyKitTrays	179
8.28.3.7	gethasLargeBoxWithEmptyKitTrays_LargeContainer	179
8.28.3.8	getLargeBoxWithEmptyKitTraysID	179
8.28.3.9	set	179
8.28.3.10	set	179
8.28.3.11	setdao	180
8.28.3.12	sethadByKitTray_LargeBoxWithEmptyKitTrays	180
8.28.3.13	sethasLargeBoxWithEmptyKitTrays_LargeContainer	180
8.28.4	Member Data Documentation	180
8.28.4.1	dao	180
8.28.4.2	hadByKitTray_LargeBoxWithEmptyKitTrays	180
8.28.4.3	hasLargeBoxWithEmptyKitTrays_LargeContainer	180
8.28.4.4	LargeBoxWithEmptyKitTraysID	180
8.29	LargeBoxWithKits Class Reference	180
8.29.1	Detailed Description	183
8.29.2	Constructor & Destructor Documentation	183
8.29.2.1	LargeBoxWithKits	183
8.29.2.2	~LargeBoxWithKits	183
8.29.3	Member Function Documentation	184
8.29.3.1	copy	184
8.29.3.2	Explode	184
8.29.3.3	get	184
8.29.3.4	get	184
8.29.3.5	getdao	185
8.29.3.6	gethadByKit_LargeBoxWithKits	185
8.29.3.7	gethasLargeBoxWithKits_Capacity	185
8.29.3.8	gethasLargeBoxWithKits_KitDesignRef	185
8.29.3.9	gethasLargeBoxWithKits_LargeContainer	185
8.29.3.10	getLargeBoxWithKitsID	185
8.29.3.11	set	185
8.29.3.12	set	185
8.29.3.13	setdao	186
8.29.3.14	sethadByKit_LargeBoxWithKits	186

8.29.3.15 sethasLargeBoxWithKits_Capacity	186
8.29.3.16 sethasLargeBoxWithKits_KitDesignRef	186
8.29.3.17 sethasLargeBoxWithKits_LargeContainer	186
8.29.4 Member Data Documentation	186
8.29.4.1 dao	186
8.29.4.2 hadByKit_LargeBoxWithKits	186
8.29.4.3 hasLargeBoxWithKits_Capacity	187
8.29.4.4 hasLargeBoxWithKits_KitDesignRef	187
8.29.4.5 hasLargeBoxWithKits_LargeContainer	187
8.29.4.6 LargeBoxWithKitsID	187
8.30 LargeContainer Class Reference	187
8.30.1 Detailed Description	190
8.30.2 Constructor & Destructor Documentation	190
8.30.2.1 LargeContainer	190
8.30.2.2 ~LargeContainer	190
8.30.3 Member Function Documentation	191
8.30.3.1 copy	191
8.30.3.2 Explode	191
8.30.3.3 get	191
8.30.3.4 get	191
8.30.3.5 getdao	191
8.30.3.6 gethasLargeBoxWithEmptyKitTrays_LargeContainer	192
8.30.3.7 gethasLargeBoxWithKits_LargeContainer	192
8.30.3.8 gethasLargeContainer_SerialNumber	192
8.30.3.9 gethasLargeContainer_SkuRef	192
8.30.3.10 getLargeContainerID	192
8.30.3.11 set	192
8.30.3.12 set	192
8.30.3.13 setdao	192
8.30.3.14 sethasLargeBoxWithEmptyKitTrays_LargeContainer	193
8.30.3.15 sethasLargeBoxWithKits_LargeContainer	193
8.30.3.16 sethasLargeContainer_SerialNumber	193
8.30.3.17 sethasLargeContainer_SkuRef	193
8.30.4 Member Data Documentation	193
8.30.4.1 dao	193
8.30.4.2 hasLargeBoxWithEmptyKitTrays_LargeContainer	193
8.30.4.3 hasLargeBoxWithKits_LargeContainer	193

8.30.4.4	hasLargeContainer_SerialNumber	193
8.30.4.5	hasLargeContainer_SkuRef	193
8.30.4.6	LargeContainerID	193
8.31	MessageMsg Class Reference	194
8.31.1	Detailed Description	196
8.31.2	Constructor & Destructor Documentation	196
8.31.2.1	MessageMsg	196
8.31.2.2	MessageMsg	196
8.31.2.3	~MessageMsg	196
8.31.3	Member Data Documentation	196
8.31.3.1	message	196
8.32	MoveSmoothlyToMsg Class Reference	196
8.32.1	Detailed Description	199
8.32.2	Constructor & Destructor Documentation	199
8.32.2.1	MoveSmoothlyToMsg	199
8.32.2.2	~MoveSmoothlyToMsg	199
8.32.3	Member Data Documentation	199
8.32.3.1	num	199
8.33	MoveStraightToMsg Class Reference	199
8.33.1	Detailed Description	201
8.33.2	Constructor & Destructor Documentation	201
8.33.2.1	MoveStraightToMsg	202
8.33.2.2	~MoveStraightToMsg	202
8.34	MoveToMsg Class Reference	202
8.34.1	Detailed Description	203
8.34.2	Constructor & Destructor Documentation	203
8.34.2.1	MoveToMsg	204
8.34.2.2	~MoveToMsg	204
8.35	MyController Class Reference	204
8.35.1	Detailed Description	207
8.35.2	Constructor & Destructor Documentation	207
8.35.2.1	MyController	207
8.35.2.2	~MyController	207
8.35.3	Member Function Documentation	207
8.35.3.1	processCloseGripper	207
8.35.3.2	processDwell	207
8.35.3.3	processEndCanon	207

8.35.3.4 processInitCanon	208
8.35.3.5 processMessage	208
8.35.3.6 processMoveSmoothlyTo	208
8.35.3.7 processMoveStraightTo	208
8.35.3.8 processMoveTo	208
8.35.3.9 processOpenGripper	208
8.35.3.10 processSetAbsoluteAcceleration	208
8.35.3.11 processSetAngleUnits	208
8.35.3.12 processSetEndAngleTolerance	209
8.35.3.13 processSetEndPointTolerance	209
8.35.3.14 processSetIntermediatePointTolerance	209
8.35.3.15 processSetLengthUnits	209
8.35.3.16 processSetRelativeAcceleration	209
8.35.3.17 processSetRelativeSpeed	209
8.36 OpenGripperMsg Class Reference	209
8.36.1 Detailed Description	211
8.36.2 Constructor & Destructor Documentation	211
8.36.2.1 OpenGripperMsg	212
8.36.2.2 ~OpenGripperMsg	212
8.37 Part Class Reference	212
8.37.1 Detailed Description	215
8.37.2 Constructor & Destructor Documentation	215
8.37.2.1 Part	215
8.37.2.2 ~Part	215
8.37.3 Member Function Documentation	215
8.37.3.1 copy	215
8.37.3.2 Explode	216
8.37.3.3 get	216
8.37.3.4 get	216
8.37.3.5 getdao	217
8.37.3.6 gethadByPart_Kit	217
8.37.3.7 gethadByPart_PartsTrayWithParts	217
8.37.3.8 gethasPart_SerialNumber	217
8.37.3.9 gethasPart_SkuRef	217
8.37.3.10 getPartID	217
8.37.3.11 set	217
8.37.3.12 set	217

8.37.3.13 setdao	218
8.37.3.14 sethadByPart_Kit	218
8.37.3.15 sethadByPart_PartsTrayWithParts	218
8.37.3.16 sethasPart_SerialNumber	218
8.37.3.17 sethasPart_SkuRef	218
8.37.4 Member Data Documentation	218
8.37.4.1 dao	218
8.37.4.2 hadByPart_Kit	218
8.37.4.3 hadByPart_PartsTrayWithParts	218
8.37.4.4 hasPart_SerialNumber	218
8.37.4.5 hasPart_SkuRef	218
8.37.4.6 PartID	219
8.38 PartLocStruct Struct Reference	219
8.38.1 Detailed Description	221
8.38.2 Member Data Documentation	221
8.38.2.1 point	221
8.38.2.2 x_axis	221
8.38.2.3 z_axis	221
8.39 PartRefAndPose Class Reference	221
8.39.1 Detailed Description	224
8.39.2 Constructor & Destructor Documentation	224
8.39.2.1 PartRefAndPose	224
8.39.2.2 ~PartRefAndPose	224
8.39.3 Member Function Documentation	224
8.39.3.1 copy	224
8.39.3.2 Explode	225
8.39.3.3 get	225
8.39.3.4 get	225
8.39.3.5 getdao	226
8.39.3.6 gethadByPartRefAndPose_KitDesign	226
8.39.3.7 gethasPartRefAndPose_Point	226
8.39.3.8 gethasPartRefAndPose_Ref	226
8.39.3.9 gethasPartRefAndPose_XAxis	226
8.39.3.10 gethasPartRefAndPose_ZAxis	226
8.39.3.11 getPartRefAndPoselD	227
8.39.3.12 set	227
8.39.3.13 set	227

8.39.3.14 setdao	227
8.39.3.15 sethadByPartRefAndPose_KitDesign	227
8.39.3.16 sethasPartRefAndPose_Point	227
8.39.3.17 sethasPartRefAndPose_Ref	227
8.39.3.18 sethasPartRefAndPose_XAxis	227
8.39.3.19 sethasPartRefAndPose_ZAxis	228
8.39.4 Member Data Documentation	228
8.39.4.1 dao	228
8.39.4.2 hadByPartRefAndPose_KitDesign	228
8.39.4.3 hasPartRefAndPose_Point	228
8.39.4.4 hasPartRefAndPose_Ref	228
8.39.4.5 hasPartRefAndPose_XAxis	228
8.39.4.6 hasPartRefAndPose_ZAxis	228
8.39.4.7 PartRefAndPoseID	228
8.40 PartsBin Class Reference	228
8.40.1 Detailed Description	231
8.40.2 Constructor & Destructor Documentation	231
8.40.2.1 PartsBin	231
8.40.2.2 ~PartsBin	231
8.40.3 Member Function Documentation	231
8.40.3.1 copy	231
8.40.3.2 Explode	232
8.40.3.3 get	232
8.40.3.4 get	232
8.40.3.5 getdao	232
8.40.3.6 gethasBin_PartQuantity	232
8.40.3.7 gethasBin_PartSkuRef	233
8.40.3.8 getPartsBinID	233
8.40.3.9 set	233
8.40.3.10 set	233
8.40.3.11 setdao	233
8.40.3.12 sethasBin_PartQuantity	233
8.40.3.13 sethasBin_PartSkuRef	233
8.40.4 Member Data Documentation	233
8.40.4.1 dao	233
8.40.4.2 hasBin_PartQuantity	234
8.40.4.3 hasBin_PartSkuRef	234

8.40.4.4	PartsBinID	234
8.41	PartsTray Class Reference	234
8.41.1	Detailed Description	237
8.41.2	Constructor & Destructor Documentation	237
8.41.2.1	PartsTray	237
8.41.2.2	\sim PartsTray	237
8.41.3	Member Function Documentation	237
8.41.3.1	copy	237
8.41.3.2	Explode	238
8.41.3.3	get	238
8.41.3.4	get	238
8.41.3.5	getdao	239
8.41.3.6	gethasPartsTray_SerialNumber	239
8.41.3.7	gethasPartsTray_SkuRef	239
8.41.3.8	gethasPartsTrayWithParts_Tray	239
8.41.3.9	getPartsTrayID	239
8.41.3.10	set	239
8.41.3.11	set	239
8.41.3.12	setdao	239
8.41.3.13	sethasPartsTray_SerialNumber	240
8.41.3.14	sethasPartsTray_SkuRef	240
8.41.3.15	sethasPartsTrayWithParts_Tray	240
8.41.4	Member Data Documentation	240
8.41.4.1	dao	240
8.41.4.2	hasPartsTray_SerialNumber	240
8.41.4.3	hasPartsTray_SkuRef	240
8.41.4.4	hasPartsTrayWithParts_Tray	240
8.41.4.5	PartsTrayID	240
8.42	PartsTrayLocStruct Struct Reference	240
8.42.1	Detailed Description	242
8.42.2	Member Data Documentation	242
8.42.2.1	point	242
8.42.2.2	x_axis	242
8.42.2.3	z_axis	242
8.43	PartsTrayWithParts Class Reference	242
8.43.1	Detailed Description	245
8.43.2	Constructor & Destructor Documentation	245

8.43.2.1	PartsTrayWithParts	245
8.43.2.2	~PartsTrayWithParts	245
8.43.3	Member Function Documentation	245
8.43.3.1	copy	245
8.43.3.2	Explode	246
8.43.3.3	get	246
8.43.3.4	get	246
8.43.3.5	getdao	247
8.43.3.6	gethadByPart_PartsTrayWithParts	247
8.43.3.7	gethasPartsTrayWithParts_Tray	247
8.43.3.8	getPartsTrayWithPartsID	247
8.43.3.9	set	247
8.43.3.10	set	247
8.43.3.11	setdao	248
8.43.3.12	sethadByPart_PartsTrayWithParts	248
8.43.3.13	sethasPartsTrayWithParts_Tray	248
8.43.4	Member Data Documentation	248
8.43.4.1	dao	248
8.43.4.2	hadByPart_PartsTrayWithParts	248
8.43.4.3	hasPartsTrayWithParts_Tray	248
8.43.4.4	PartsTrayWithPartsID	248
8.44	PhysicalLocation Class Reference	249
8.44.1	Detailed Description	251
8.44.2	Constructor & Destructor Documentation	251
8.44.2.1	PhysicalLocation	251
8.44.2.2	~PhysicalLocation	251
8.44.3	Member Function Documentation	251
8.44.3.1	copy	251
8.44.3.2	Explode	252
8.44.3.3	get	253
8.44.3.4	get	253
8.44.3.5	getdao	253
8.44.3.6	gethasPhysicalLocation_RefObject	254
8.44.3.7	gethasSolidObject_PrimaryLocation	254
8.44.3.8	gethasSolidObject_SecondaryLocation	254
8.44.3.9	getPhysicalLocationID	254
8.44.3.10	set	254

8.44.3.11 set	255
8.44.3.12 setdao	255
8.44.3.13 sethasPhysicalLocation_RefObject	255
8.44.3.14 sethasSolidObject_PrimaryLocation	255
8.44.3.15 sethasSolidObject_SecondaryLocation	255
8.44.4 Member Data Documentation	255
8.44.4.1 dao	255
8.44.4.2 hasPhysicalLocation_RefObject	255
8.44.4.3 hasSolidObject_PrimaryLocation	256
8.44.4.4 hasSolidObject_SecondaryLocation	256
8.44.4.5 PhysicalLocationID	256
8.45 Point Class Reference	256
8.45.1 Detailed Description	259
8.45.2 Constructor & Destructor Documentation	259
8.45.2.1 Point	259
8.45.2.2 ~Point	260
8.45.3 Member Function Documentation	260
8.45.3.1 copy	260
8.45.3.2 Explode	260
8.45.3.3 get	261
8.45.3.4 get	261
8.45.3.5 getdao	261
8.45.3.6 gethasBoxVolume_MaximumPoint	261
8.45.3.7 gethasBoxVolume_MinimumPoint	261
8.45.3.8 gethasPartRefAndPose_Point	261
8.45.3.9 gethasPoint_X	262
8.45.3.10 gethasPoint_Y	262
8.45.3.11 gethasPoint_Z	262
8.45.3.12 gethasPoseLocation_Point	262
8.45.3.13 getPointID	262
8.45.3.14 set	263
8.45.3.15 set	263
8.45.3.16 setdao	263
8.45.3.17 sethasBoxVolume_MaximumPoint	263
8.45.3.18 sethasBoxVolume_MinimumPoint	263
8.45.3.19 sethasPartRefAndPose_Point	263
8.45.3.20 sethasPoint_X	263

8.45.3.21 <code>sethasPoint_Y</code>	263
8.45.3.22 <code>sethasPoint_Z</code>	264
8.45.3.23 <code>sethasPoseLocation_Point</code>	264
8.45.4 Member Data Documentation	264
8.45.4.1 <code>dao</code>	264
8.45.4.2 <code>hasBoxVolume_MaximumPoint</code>	264
8.45.4.3 <code>hasBoxVolume_MinimumPoint</code>	264
8.45.4.4 <code>hasPartRefAndPose_Point</code>	264
8.45.4.5 <code>hasPoint_X</code>	264
8.45.4.6 <code>hasPoint_Y</code>	264
8.45.4.7 <code>hasPoint_Z</code>	264
8.45.4.8 <code>hasPoseLocation_Point</code>	264
8.45.4.9 <code>PointID</code>	265
8.46 PoseLocation Class Reference	265
8.46.1 Detailed Description	268
8.46.2 Constructor & Destructor Documentation	268
8.46.2.1 <code>PoseLocation</code>	268
8.46.2.2 <code>~PoseLocation</code>	268
8.46.3 Member Function Documentation	268
8.46.3.1 <code>copy</code>	268
8.46.3.2 <code>Explode</code>	269
8.46.3.3 <code>get</code>	269
8.46.3.4 <code>get</code>	269
8.46.3.5 <code>getdao</code>	270
8.46.3.6 <code>gethasPoseLocation_Point</code>	270
8.46.3.7 <code>gethasPoseLocation_XAxis</code>	270
8.46.3.8 <code>gethasPoseLocation_ZAxis</code>	271
8.46.3.9 <code>getPoseLocationID</code>	271
8.46.3.10 <code>set</code>	271
8.46.3.11 <code>set</code>	271
8.46.3.12 <code>setdao</code>	272
8.46.3.13 <code>sethasPoseLocation_Point</code>	272
8.46.3.14 <code>sethasPoseLocation_XAxis</code>	272
8.46.3.15 <code>sethasPoseLocation_ZAxis</code>	272
8.46.4 Member Data Documentation	272
8.46.4.1 <code>dao</code>	272
8.46.4.2 <code>hasPoseLocation_Point</code>	272

8.46.4.3 hasPoseLocation_XAxis	272
8.46.4.4 hasPoseLocation_ZAxis	273
8.46.4.5 PoseLocationID	273
8.47 PoseLocationIn Class Reference	273
8.47.1 Detailed Description	276
8.47.2 Constructor & Destructor Documentation	276
8.47.2.1 PoseLocationIn	276
8.47.2.2 ~PoseLocationIn	276
8.47.3 Member Function Documentation	276
8.47.3.1 copy	276
8.47.3.2 Explode	277
8.47.3.3 get	277
8.47.3.4 get	277
8.47.3.5 getdao	277
8.47.3.6 getPoseLocationInID	277
8.47.3.7 set	277
8.47.3.8 set	277
8.47.3.9 setdao	278
8.47.4 Member Data Documentation	278
8.47.4.1 dao	278
8.47.4.2 PoseLocationInID	278
8.48 PoseLocationOn Class Reference	278
8.48.1 Detailed Description	281
8.48.2 Constructor & Destructor Documentation	281
8.48.2.1 PoseLocationOn	281
8.48.2.2 ~PoseLocationOn	281
8.48.3 Member Function Documentation	281
8.48.3.1 copy	281
8.48.3.2 Explode	282
8.48.3.3 get	282
8.48.3.4 get	282
8.48.3.5 getdao	282
8.48.3.6 getPoseLocationOnID	282
8.48.3.7 set	282
8.48.3.8 set	282
8.48.3.9 setdao	283
8.48.4 Member Data Documentation	283

8.48.4.1	dao	283
8.48.4.2	PoseLocationOnID	283
8.49	PoseOnlyLocation Class Reference	283
8.49.1	Detailed Description	286
8.49.2	Constructor & Destructor Documentation	286
8.49.2.1	PoseOnlyLocation	286
8.49.2.2	~PoseOnlyLocation	286
8.49.3	Member Function Documentation	286
8.49.3.1	copy	286
8.49.3.2	Explode	287
8.49.3.3	get	287
8.49.3.4	get	287
8.49.3.5	getdao	287
8.49.3.6	getPoseOnlyLocationID	287
8.49.3.7	set	287
8.49.3.8	set	287
8.49.3.9	setdao	288
8.49.4	Member Data Documentation	288
8.49.4.1	dao	288
8.49.4.2	PoseOnlyLocationID	288
8.50	RelativeLocation Class Reference	288
8.50.1	Detailed Description	291
8.50.2	Constructor & Destructor Documentation	291
8.50.2.1	RelativeLocation	291
8.50.2.2	~RelativeLocation	291
8.50.3	Member Function Documentation	291
8.50.3.1	copy	291
8.50.3.2	Explode	292
8.50.3.3	get	292
8.50.3.4	get	292
8.50.3.5	getdao	292
8.50.3.6	gethasRelativeLocation_Description	293
8.50.3.7	getRelativeLocationID	293
8.50.3.8	set	293
8.50.3.9	set	293
8.50.3.10	setdao	293
8.50.3.11	sethasRelativeLocation_Description	294

8.50.4 Member Data Documentation	294
8.50.4.1 dao	294
8.50.4.2 hasRelativeLocation_Description	294
8.50.4.3 RelativeLocationID	294
8.51 RelativeLocationIn Class Reference	294
8.51.1 Detailed Description	297
8.51.2 Constructor & Destructor Documentation	297
8.51.2.1 RelativeLocationIn	297
8.51.2.2 ~RelativeLocationIn	297
8.51.3 Member Function Documentation	297
8.51.3.1 copy	297
8.51.3.2 Explode	298
8.51.3.3 get	298
8.51.3.4 get	298
8.51.3.5 getdao	298
8.51.3.6 getRelativeLocationInID	298
8.51.3.7 set	298
8.51.3.8 set	298
8.51.3.9 setdao	299
8.51.4 Member Data Documentation	299
8.51.4.1 dao	299
8.51.4.2 RelativeLocationInID	299
8.52 RelativeLocationOn Class Reference	299
8.52.1 Detailed Description	302
8.52.2 Constructor & Destructor Documentation	302
8.52.2.1 RelativeLocationOn	302
8.52.2.2 ~RelativeLocationOn	302
8.52.3 Member Function Documentation	302
8.52.3.1 copy	302
8.52.3.2 Explode	303
8.52.3.3 get	303
8.52.3.4 get	303
8.52.3.5 getdao	303
8.52.3.6 getRelativeLocationOnID	303
8.52.3.7 set	303
8.52.3.8 set	303
8.52.3.9 setdao	304

8.52.4 Member Data Documentation	304
8.52.4.1 dao	304
8.52.4.2 RelativeLocationOnID	304
8.53 Robot Class Reference	304
8.53.1 Detailed Description	307
8.53.2 Constructor & Destructor Documentation	307
8.53.2.1 Robot	307
8.53.2.2 ~Robot	307
8.53.3 Member Function Documentation	308
8.53.3.1 copy	308
8.53.3.2 Explode	308
8.53.3.3 get	308
8.53.3.4 get	308
8.53.3.5 getdao	309
8.53.3.6 gethadByEndEffector_Robot	309
8.53.3.7 gethasRobot_Description	309
8.53.3.8 gethasRobot_Id	309
8.53.3.9 gethasRobot_MaximumLoadWeight	309
8.53.3.10 gethasRobot_WorkVolume	309
8.53.3.11 gethasWorkstation_Robot	309
8.53.3.12 getRobotID	309
8.53.3.13 set	310
8.53.3.14 set	310
8.53.3.15 setdao	310
8.53.3.16 sethadByEndEffector_Robot	310
8.53.3.17 sethasRobot_Description	310
8.53.3.18 sethasRobot_Id	310
8.53.3.19 sethasRobot_MaximumLoadWeight	310
8.53.3.20 sethasRobot_WorkVolume	310
8.53.3.21 sethasWorkstation_Robot	311
8.53.4 Member Data Documentation	311
8.53.4.1 dao	311
8.53.4.2 hadByEndEffector_Robot	311
8.53.4.3 hasRobot_Description	311
8.53.4.4 hasRobot_Id	311
8.53.4.5 hasRobot_MaximumLoadWeight	311
8.53.4.6 hasRobot_WorkVolume	311

8.53.4.7 hasWorkstation_Robot	311
8.53.4.8 RobotID	311
8.54 SetAbsoluteAccelerationMsg Class Reference	312
8.54.1 Detailed Description	314
8.54.2 Constructor & Destructor Documentation	314
8.54.2.1 SetAbsoluteAccelerationMsg	314
8.54.2.2 SetAbsoluteAccelerationMsg	314
8.54.2.3 ~SetAbsoluteAccelerationMsg	314
8.54.3 Member Data Documentation	314
8.54.3.1 acceleration	314
8.55 SetAngleUnitsMsg Class Reference	314
8.55.1 Detailed Description	317
8.55.2 Constructor & Destructor Documentation	317
8.55.2.1 SetAngleUnitsMsg	317
8.55.2.2 SetAngleUnitsMsg	317
8.55.2.3 ~SetAngleUnitsMsg	317
8.55.3 Member Data Documentation	317
8.55.3.1 unitName	317
8.56 SetEndAngleToleranceMsg Class Reference	317
8.56.1 Detailed Description	320
8.56.2 Constructor & Destructor Documentation	320
8.56.2.1 SetEndAngleToleranceMsg	320
8.56.2.2 SetEndAngleToleranceMsg	320
8.56.2.3 ~SetEndAngleToleranceMsg	320
8.56.3 Member Data Documentation	320
8.56.3.1 tolerance	320
8.57 SetEndPointToleranceMsg Class Reference	320
8.57.1 Detailed Description	323
8.57.2 Constructor & Destructor Documentation	323
8.57.2.1 SetEndPointToleranceMsg	323
8.57.2.2 SetEndPointToleranceMsg	323
8.57.2.3 ~SetEndPointToleranceMsg	323
8.57.3 Member Data Documentation	323
8.57.3.1 tolerance	323
8.58 SetIntermediatePointToleranceMsg Class Reference	323
8.58.1 Detailed Description	326
8.58.2 Constructor & Destructor Documentation	326

8.58.2.1	SetIntermediatePointToleranceMsg	326
8.58.2.2	SetIntermediatePointToleranceMsg	326
8.58.2.3	~SetIntermediatePointToleranceMsg	326
8.58.3	Member Data Documentation	326
8.58.3.1	tolerance	326
8.59	SetLengthUnitsMsg Class Reference	326
8.59.1	Detailed Description	329
8.59.2	Constructor & Destructor Documentation	329
8.59.2.1	SetLengthUnitsMsg	329
8.59.2.2	SetLengthUnitsMsg	329
8.59.2.3	~SetLengthUnitsMsg	329
8.59.3	Member Data Documentation	329
8.59.3.1	unitName	329
8.60	SetRelativeAccelerationMsg Class Reference	329
8.60.1	Detailed Description	332
8.60.2	Constructor & Destructor Documentation	332
8.60.2.1	SetRelativeAccelerationMsg	332
8.60.2.2	SetRelativeAccelerationMsg	332
8.60.2.3	~SetRelativeAccelerationMsg	332
8.60.3	Member Data Documentation	332
8.60.3.1	percent	332
8.61	SetRelativeSpeedMsg Class Reference	332
8.61.1	Detailed Description	335
8.61.2	Constructor & Destructor Documentation	335
8.61.2.1	SetRelativeSpeedMsg	335
8.61.2.2	SetRelativeSpeedMsg	335
8.61.2.3	~SetRelativeSpeedMsg	335
8.61.3	Member Data Documentation	335
8.61.3.1	percent	335
8.62	ShapeDesign Class Reference	335
8.62.1	Detailed Description	338
8.62.2	Constructor & Destructor Documentation	338
8.62.2.1	ShapeDesign	338
8.62.2.2	~ShapeDesign	338
8.62.3	Member Function Documentation	338
8.62.3.1	copy	338
8.62.3.2	Explode	339

8.62.3.3 get	339
8.62.3.4 get	339
8.62.3.5 getdao	340
8.62.3.6 gethasShapeDesign_Description	340
8.62.3.7 gethasSku_Shape	340
8.62.3.8 getShapeDesignID	340
8.62.3.9 set	340
8.62.3.10 set	340
8.62.3.11 setdao	341
8.62.3.12 sethasShapeDesign_Description	341
8.62.3.13 sethasSku_Shape	341
8.62.4 Member Data Documentation	341
8.62.4.1 dao	341
8.62.4.2 hasShapeDesign_Description	341
8.62.4.3 hasSku_Shape	341
8.62.4.4 ShapeDesignID	341
8.63 SolidObject Class Reference	342
8.63.1 Detailed Description	344
8.63.2 Constructor & Destructor Documentation	344
8.63.2.1 SolidObject	344
8.63.2.2 ~SolidObject	345
8.63.3 Member Function Documentation	345
8.63.3.1 copy	345
8.63.3.2 Explode	346
8.63.3.3 get	347
8.63.3.4 get	348
8.63.3.5 getdao	348
8.63.3.6 gethadBySolidObject_WorkTable	348
8.63.3.7 gethasPhysicalLocation_RefObject	348
8.63.3.8 gethasSolidObject_PrimaryLocation	348
8.63.3.9 gethasSolidObject_SecondaryLocation	348
8.63.3.10 getname	349
8.63.3.11 getSolidObjectID	350
8.63.3.12 set	351
8.63.3.13 set	352
8.63.3.14 setdao	352
8.63.3.15 sethadBySolidObject_WorkTable	352

8.63.3.16 sethasPhysicalLocation_RefObject	352
8.63.3.17 sethasSolidObject_PrimaryLocation	352
8.63.3.18 sethasSolidObject_SecondaryLocation	352
8.63.4 Member Data Documentation	352
8.63.4.1 dao	352
8.63.4.2 hadBySolidObject_WorkTable	353
8.63.4.3 hasPhysicalLocation_RefObject	353
8.63.4.4 hasSolidObject_PrimaryLocation	353
8.63.4.5 hasSolidObject_SecondaryLocation	353
8.63.4.6 name	353
8.63.4.7 SolidObjectID	353
8.64 StockKeepingUnit Class Reference	353
8.64.1 Detailed Description	356
8.64.2 Constructor & Destructor Documentation	356
8.64.2.1 StockKeepingUnit	356
8.64.2.2 ~StockKeepingUnit	356
8.64.3 Member Function Documentation	357
8.64.3.1 copy	357
8.64.3.2 Explode	357
8.64.3.3 get	357
8.64.3.4 get	357
8.64.3.5 getdao	358
8.64.3.6 gethadBySku_Workstation	358
8.64.3.7 gethasSku_Description	358
8.64.3.8 gethasSku_EndEffectorRefs	358
8.64.3.9 gethasSku_Id	358
8.64.3.10 gethasSku_Shape	358
8.64.3.11 gethasSku_Weight	358
8.64.3.12 getStockKeepingUnitID	358
8.64.3.13 set	359
8.64.3.14 set	359
8.64.3.15 setdao	359
8.64.3.16 sethadBySku_Workstation	359
8.64.3.17 sethasSku_Description	359
8.64.3.18 sethasSku_EndEffectorRefs	359
8.64.3.19 sethasSku_Id	359
8.64.3.20 sethasSku_Shape	359

8.64.3.21 <code>sethasSku_Weight</code>	360
8.64.4 Member Data Documentation	360
8.64.4.1 <code>dao</code>	360
8.64.4.2 <code>hadBySku_Workstation</code>	360
8.64.4.3 <code>hasSku_Description</code>	360
8.64.4.4 <code>hasSku_EndEffectorRefs</code>	360
8.64.4.5 <code>hasSku_Id</code>	360
8.64.4.6 <code>hasSku_Shape</code>	360
8.64.4.7 <code>hasSku_Weight</code>	360
8.64.4.8 <code>StockKeepingUnitID</code>	360
8.65 <code>unix_shm_struct</code> Struct Reference	361
8.65.1 Detailed Description	361
8.65.2 Member Data Documentation	361
8.65.2.1 <code>addr</code>	361
8.65.2.2 <code>id</code>	361
8.65.2.3 <code>key</code>	361
8.65.2.4 <code>size</code>	361
8.66 VacuumEffector Class Reference	362
8.66.1 Detailed Description	365
8.66.2 Constructor & Destructor Documentation	365
8.66.2.1 <code>VacuumEffector</code>	365
8.66.2.2 <code>~VacuumEffector</code>	365
8.66.3 Member Function Documentation	365
8.66.3.1 <code>copy</code>	365
8.66.3.2 <code>Explode</code>	366
8.66.3.3 <code>get</code>	366
8.66.3.4 <code>get</code>	366
8.66.3.5 <code>getdao</code>	366
8.66.3.6 <code>gethasVacuumEffector_CupDiameter</code>	367
8.66.3.7 <code>gethasVacuumEffector_Length</code>	367
8.66.3.8 <code>getVacuumEffectorID</code>	367
8.66.3.9 <code>set</code>	367
8.66.3.10 <code>set</code>	367
8.66.3.11 <code>setdao</code>	368
8.66.3.12 <code>sethasVacuumEffector_CupDiameter</code>	368
8.66.3.13 <code>sethasVacuumEffector_Length</code>	368
8.66.4 Member Data Documentation	368

8.66.4.1	dao	368
8.66.4.2	hasVacuumEffector_CupDiameter	368
8.66.4.3	hasVacuumEffector_Length	368
8.66.4.4	VacuumEffectorID	368
8.67	VacuumEffectorMultiCup Class Reference	368
8.67.1	Detailed Description	371
8.67.2	Constructor & Destructor Documentation	371
8.67.2.1	VacuumEffectorMultiCup	371
8.67.2.2	~VacuumEffectorMultiCup	371
8.67.3	Member Function Documentation	371
8.67.3.1	copy	371
8.67.3.2	Explode	372
8.67.3.3	get	372
8.67.3.4	get	372
8.67.3.5	getdao	372
8.67.3.6	gethasMultiCup_ArrayNumber	372
8.67.3.7	gethasMultiCup_ArrayRadius	373
8.67.3.8	getVacuumEffectorMultiCupID	373
8.67.3.9	set	373
8.67.3.10	set	373
8.67.3.11	setdao	373
8.67.3.12	sethasMultiCup_ArrayNumber	373
8.67.3.13	sethasMultiCup_ArrayRadius	373
8.67.4	Member Data Documentation	373
8.67.4.1	dao	373
8.67.4.2	hasMultiCup_ArrayNumber	374
8.67.4.3	hasMultiCup_ArrayRadius	374
8.67.4.4	VacuumEffectorMultiCupID	374
8.68	VacuumEffectorSingleCup Class Reference	374
8.68.1	Detailed Description	377
8.68.2	Constructor & Destructor Documentation	377
8.68.2.1	VacuumEffectorSingleCup	377
8.68.2.2	~VacuumEffectorSingleCup	377
8.68.3	Member Function Documentation	377
8.68.3.1	copy	377
8.68.3.2	Explode	378
8.68.3.3	get	378

8.68.3.4	get	378
8.68.3.5	getdao	378
8.68.3.6	getVacuumEffectorSingleCupID	378
8.68.3.7	set	378
8.68.3.8	set	378
8.68.3.9	setdao	379
8.68.4	Member Data Documentation	379
8.68.4.1	dao	379
8.68.4.2	VacuumEffectorSingleCupID	379
8.69	Vector Class Reference	379
8.69.1	Detailed Description	382
8.69.2	Constructor & Destructor Documentation	382
8.69.2.1	Vector	382
8.69.2.2	~Vector	383
8.69.3	Member Function Documentation	383
8.69.3.1	copy	383
8.69.3.2	Explode	383
8.69.3.3	get	384
8.69.3.4	get	384
8.69.3.5	getdao	384
8.69.3.6	gethasPartRefAndPose_XAxis	384
8.69.3.7	gethasPartRefAndPose_ZAxis	384
8.69.3.8	gethasPoseLocation_XAxis	384
8.69.3.9	gethasPoseLocation_ZAxis	385
8.69.3.10	gethasVector_I	385
8.69.3.11	gethasVector_J	385
8.69.3.12	gethasVector_K	385
8.69.3.13	getVectorID	385
8.69.3.14	set	385
8.69.3.15	set	385
8.69.3.16	setdao	386
8.69.3.17	sethasPartRefAndPose_XAxis	386
8.69.3.18	sethasPartRefAndPose_ZAxis	386
8.69.3.19	sethasPoseLocation_XAxis	386
8.69.3.20	sethasPoseLocation_ZAxis	386
8.69.3.21	sethasVector_I	386
8.69.3.22	sethasVector_J	386

8.69.3.23 sethasVector_K	387
8.69.4 Member Data Documentation	387
8.69.4.1 dao	387
8.69.4.2 hasPartRefAndPose_XAxis	387
8.69.4.3 hasPartRefAndPose_ZAxis	387
8.69.4.4 hasPoseLocation_XAxis	387
8.69.4.5 hasPoseLocation_ZAxis	387
8.69.4.6 hasVector_I	387
8.69.4.7 hasVector_J	387
8.69.4.8 hasVector_K	387
8.69.4.9 VectorID	387
8.70 WorkTable Class Reference	388
8.70.1 Detailed Description	390
8.70.2 Constructor & Destructor Documentation	390
8.70.2.1 WorkTable	390
8.70.2.2 ~WorkTable	390
8.70.3 Member Function Documentation	390
8.70.3.1 copy	390
8.70.3.2 Explode	391
8.70.3.3 get	391
8.70.3.4 get	391
8.70.3.5 getdao	392
8.70.3.6 gethadBySolidObject_WorkTable	392
8.70.3.7 gethasWorkstation_WorkTable	392
8.70.3.8 getWorkTableID	392
8.70.3.9 set	392
8.70.3.10 set	392
8.70.3.11 setdao	393
8.70.3.12 sethadBySolidObject_WorkTable	393
8.70.3.13 sethasWorkstation_WorkTable	393
8.70.4 Member Data Documentation	393
8.70.4.1 dao	393
8.70.4.2 hadBySolidObject_WorkTable	393
8.70.4.3 hasWorkstation_WorkTable	393
8.70.4.4 WorkTableID	394

9.1	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controllercanonicalMsg.hh File Reference	395
9.1.1	Detailed Description	397
9.1.2	Enumeration Type Documentation	397
9.1.2.1	CanonicalType	397
9.2	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controllercontroller.cpp File Reference	398
9.3	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controllercontroller.hh File Reference	399
9.3.1	Detailed Description	400
9.4	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controllermyController.cpp File Reference	401
9.5	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controllermyController.hh File Reference	401
9.5.1	Detailed Description	403
9.6	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controllerulapi.cpp File Reference	404
9.6.1	Typedef Documentation	405
9.6.1.1	pthread_task_code	405
9.6.2	Function Documentation	405
9.6.2.1	ulapi_exit	405
9.6.2.2	ulapi_init	405
9.6.2.3	ulapi_mutex_delete	406
9.6.2.4	ulapi_mutex_give	406
9.6.2.5	ulapi_mutex_new	406
9.6.2.6	ulapi_mutex_take	406
9.6.2.7	ulapi_prio_highest	406
9.6.2.8	ulapi_prio_lowest	406
9.6.2.9	ulapi_prio_next_higher	407
9.6.2.10	ulapi_prio_next_lower	407
9.6.2.11	ulapi_socket_close	407
9.6.2.12	ulapi_socket_get_client_id	408
9.6.2.13	ulapi_socket_read	408
9.6.2.14	ulapi_socket_write	408
9.6.2.15	ulapi_task_delete	408
9.6.2.16	ulapi_task_init	408
9.6.2.17	ulapi_task_new	408
9.6.2.18	ulapi_task_pause	408
9.6.2.19	ulapi_task_resume	408
9.6.2.20	ulapi_task_set_period	408
9.6.2.21	ulapi_task_start	408

9.6.2.22	ulapi_task_stop	409
9.6.2.23	ulapi_time	409
9.6.2.24	unix_ulapi_exit	409
9.6.2.25	unix_ulapi_fifo_delete	409
9.6.2.26	unix_ulapi_fifo_new	409
9.6.2.27	unix_ulapi_fifo_read	409
9.6.2.28	unix_ulapi_fifo_write	409
9.6.2.29	unix_ulapi_shm_addr	409
9.6.2.30	unix_ulapi_shm_delete	409
9.6.2.31	unix_ulapi_shm_new	410
9.7	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/ulapi.hh File Reference . . .	410
9.7.1	Detailed Description	413
9.7.2	Macro Definition Documentation	414
9.7.2.1	DIGITS_IN	414
9.7.2.2	ULAPI_DECL_SHARED	414
9.7.2.3	ulapi_print	414
9.7.2.4	ulapi_snprintf	414
9.7.2.5	ulapi_sscanf	414
9.7.2.6	ulapi_strcasecmp	414
9.7.2.7	ulapi_strdup	414
9.7.2.8	ulapi_strncpy	414
9.7.3	Typedef Documentation	414
9.7.3.1	ulapi_flag	414
9.7.3.2	ulapi_id	414
9.7.3.3	ulapi_integer	415
9.7.3.4	ulapi_prio	415
9.7.3.5	ulapi_real	415
9.7.3.6	ulapi_result	415
9.7.3.7	ulapi_task_code	415
9.7.4	Enumeration Type Documentation	415
9.7.4.1	anonymous enum	415
9.7.4.2	anonymous enum	415
9.7.4.3	anonymous enum	416
9.7.4.4	ulapi_stdio	416
9.7.5	Function Documentation	416
9.7.5.1	ulapi_address_to_hostname	416
9.7.5.2	ulapi_cond_broadcast	416

9.7.5.3	ulapi_cond_delete	416
9.7.5.4	ulapi_cond_new	416
9.7.5.5	ulapi_cond_signal	416
9.7.5.6	ulapi_cond_wait	416
9.7.5.7	ulapi_exit	416
9.7.5.8	ulapi_fd_close	417
9.7.5.9	ulapi_fd_delete	417
9.7.5.10	ulapi_fd_new	417
9.7.5.11	ulapi_fd_open	417
9.7.5.12	ulapi_fd_read	417
9.7.5.13	ulapi_fd_set_blocking	417
9.7.5.14	ulapi_fd_set_nonblocking	417
9.7.5.15	ulapi_fd_write	417
9.7.5.16	ulapi_fifo_delete	417
9.7.5.17	ulapi_fifo_new	417
9.7.5.18	ulapi_fifo_read	417
9.7.5.19	ulapi_fifo_write	417
9.7.5.20	ulapi_free_argv	417
9.7.5.21	ulapi_get_host_address	417
9.7.5.22	ulapi_hostname_to_address	417
9.7.5.23	ulapi_init	418
9.7.5.24	ulapi_mutex_delete	418
9.7.5.25	ulapi_mutex_give	418
9.7.5.26	ulapi_mutex_new	418
9.7.5.27	ulapi_mutex_take	418
9.7.5.28	ulapi_prio_highest	418
9.7.5.29	ulapi_prio_lowest	419
9.7.5.30	ulapi_prio_next_higher	419
9.7.5.31	ulapi_prio_next_lower	419
9.7.5.32	ulapi_self_set_period	419
9.7.5.33	ulapi_sem_delete	420
9.7.5.34	ulapi_sem_give	420
9.7.5.35	ulapi_sem_new	420
9.7.5.36	ulapi_sem_take	420
9.7.5.37	ulapi_serial_close	420
9.7.5.38	ulapi_serial_delete	420
9.7.5.39	ulapi_serial_new	420

9.7.5.40 ulapi_serial_open	420
9.7.5.41 ulapi_serial_read	420
9.7.5.42 ulapi_serial_set_blocking	420
9.7.5.43 ulapi_serial_set_nonblocking	420
9.7.5.44 ulapi_serial_write	420
9.7.5.45 ulapi_set_debug	420
9.7.5.46 ulapi_shm_addr	420
9.7.5.47 ulapi_shm_delete	421
9.7.5.48 ulapi_shm_new	421
9.7.5.49 ulapi_sleep	421
9.7.5.50 ulapi_socket_broadcast	421
9.7.5.51 ulapi_socket_close	421
9.7.5.52 ulapi_socket_get_broadcastee_id	421
9.7.5.53 ulapi_socket_get_broadcaster_id	421
9.7.5.54 ulapi_socket_get_client_id	421
9.7.5.55 ulapi_socket_get_connection_id	421
9.7.5.56 ulapi_socket_get_server_id	421
9.7.5.57 ulapi_socket_read	422
9.7.5.58 ulapi_socket_set_blocking	422
9.7.5.59 ulapi_socket_set_nonblocking	422
9.7.5.60 ulapi_socket_write	422
9.7.5.61 ulapi_std_open	422
9.7.5.62 ulapi_system	422
9.7.5.63 ulapi_task_delete	422
9.7.5.64 ulapi_task_exit	422
9.7.5.65 ulapi_task_id	422
9.7.5.66 ulapi_task_init	422
9.7.5.67 ulapi_task_join	422
9.7.5.68 ulapi_task_new	423
9.7.5.69 ulapi_task_pause	423
9.7.5.70 ulapi_task_resume	423
9.7.5.71 ulapi_task_set_period	423
9.7.5.72 ulapi_task_start	423
9.7.5.73 ulapi_task_stop	423
9.7.5.74 ulapi_time	423
9.7.5.75 ulapi_to_argv	423
9.7.5.76 ulapi_wait	423

9.8	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/BoxVolume.cpp File Reference	424
9.9	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/BoxVolume.h File Reference	424
9.10	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/BoxyObject.cpp File Reference	425
9.11	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/BoxyObject.h File Reference	425
9.12	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Connection.cpp File Reference	426
9.13	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Connection.h File Reference	427
9.14	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/DAO.cpp File Reference . . .	427
9.15	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/DAO.h File Reference . . .	428
9.16	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/DataThing.cpp File Reference	428
9.17	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/DataThing.h File Reference .	428
9.18	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffector.cpp File Reference	429
9.19	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffector.h File Reference	430
9.20	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffectorChanging-Station.cpp File Reference	431
9.21	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffectorChanging-Station.h File Reference	431
9.22	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffectorHolder.cpp File Reference	432
9.23	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffectorHolder.h File Reference	432
9.24	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/GripperEffector.cpp File Reference	433
9.25	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/GripperEffector.h File Reference	434
9.26	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Kit.cpp File Reference	435
9.27	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Kit.h File Reference	435
9.28	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KitDesign.cpp File Reference	436
9.29	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KitDesign.h File Reference .	437
9.30	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KittingWorkstation.cpp File Reference	438
9.31	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KittingWorkstation.h File Reference	438
9.32	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KitTray.cpp File Reference . . .	439
9.33	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KitTray.h File Reference . . .	440
9.34	C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/LargeBoxWithEmptyKit-Trays.cpp File Reference	441

9.35 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/LargeBoxWithEmptyKit-Trays.h File Reference	441
9.36 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/LargeBoxWithKits.cpp File Reference	442
9.37 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/LargeBoxWithKits.h File Reference	442
9.38 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/LargeContainer.cpp File Reference	443
9.39 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/LargeContainer.h File Reference	444
9.40 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Part.cpp File Reference	445
9.41 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Part.h File Reference	445
9.42 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartRefAndPose.cpp File Reference	446
9.43 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartRefAndPose.h File Reference	447
9.44 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsBin.cpp File Reference	448
9.45 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsBin.h File Reference	448
9.46 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsTray.cpp File Reference	449
9.47 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsTray.h File Reference	449
9.48 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsTrayWithParts.cpp File Reference	451
9.49 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsTrayWithParts.h File Reference	451
9.50 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PhysicalLocation.cpp File Reference	452
9.51 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PhysicalLocation.h File Reference	452
9.52 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Point.cpp File Reference	453
9.53 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Point.h File Reference	454
9.54 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseLocation.cpp File Reference	455
9.55 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseLocation.h File Reference	455
9.56 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseLocationIn.cpp File Reference	456
9.57 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseLocationIn.h File Reference	456
9.58 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseLocationOn.cpp File Reference	457
9.59 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseLocationOn.h File Reference	458

9.60 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseOnlyLocation.cpp File Reference	459
9.61 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseOnlyLocation.h File Reference	459
9.62 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/RelativeLocation.cpp File Reference	461
9.63 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/RelativeLocation.h File Reference	461
9.64 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/RelativeLocationIn.cpp File Reference	462
9.65 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/RelativeLocationIn.h File Reference	462
9.66 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/RelativeLocationOn.cpp File Reference	464
9.67 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/RelativeLocationOn.h File Reference	464
9.68 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Robot.cpp File Reference	465
9.69 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Robot.h File Reference	465
9.70 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ShapeDesign.cpp File Reference	466
9.71 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ShapeDesign.h File Reference	467
9.72 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/SolidObject.cpp File Reference	468
9.73 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/SolidObject.h File Reference	468
9.74 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/StockKeepingUnit.cpp File Reference	469
9.75 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/StockKeepingUnit.h File Reference	470
9.76 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/VacuumEffector.cpp File Reference	471
9.77 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/VacuumEffector.h File Reference	471
9.78 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/VacuumEffectorMultiCup.cpp File Reference	472
9.79 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/VacuumEffectorMultiCup.h File Reference	472
9.80 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/VacuumEffectorSingleCup.cpp File Reference	474
9.81 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/VacuumEffectorSingleCup.h File Reference	474
9.82 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Vector.cpp File Reference	475
9.83 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Vector.h File Reference	475

9.84 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/WorkTable.cpp File Reference	476
9.85 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/WorkTable.h File Reference	477
9.86 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/CanonicalRobotCommand.cc File Reference	478
9.87 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/CanonicalRobotCommand.h File Reference	478
9.87.1 Detailed Description	480
9.88 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/Config.h File Reference	480
9.88.1 Macro Definition Documentation	480
9.88.1.1 PDDL_DOMAIN	481
9.88.1.2 PDDL_FOLDER	481
9.88.1.3 PDDL_PROBLEM	481
9.88.1.4 PLAN_FILE	481
9.88.1.5 PLAN_FOLDER	481
9.88.1.6 ROBOT_COMMANDS_FILE	481
9.88.1.7 ROBOT_COMMANDS_FOLDER	481
9.89 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/IniFile.cpp File Reference	482
9.89.1 Function Documentation	482
9.89.1.1 afterequal	482
9.89.1.2 skipwhite	483
9.90 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/IniFile.h File Reference	483
9.90.1 Macro Definition Documentation	485
9.90.1.1 COMMENT_CHAR	485
9.90.1.2INI_DEFAULT	485
9.90.1.3INI_INVALID	485
9.90.1.4INI_OK	485
9.90.1.5INIFILE_MAX_LINELEN	485
9.91 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/KittingPDDLProblem.cc File Reference	486
9.92 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/KittingPDDLProblem.h File Reference	486
9.93 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/KittingPlan.cc File Reference	487
9.94 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/KittingPlan.h File Reference	488
9.95 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/Operator.cc File Reference	489
9.96 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/Operator.h File Reference	490
9.97 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/Structdef.h File Reference	491
9.97.1 Detailed Description	491
9.98 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/Tools.h File Reference	492

9.99 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/main.cc File Reference	493
9.99.1 Function Documentation	493
9.99.1.1 main	494

Index	494
--------------	------------

Chapter 1

Main Page

The interpreter reads a plan file PLAN_INSTANCE and generate a text file that contains Canonical **Robot** Commands.

Author

Zeid Kootbally zeid.kootbally@nist.gov

Tom Kramer kramer@cme.nist.gov

Stephen Balakirsky stephen.balakirsky@nist.gov

Anthony Pietromartire pietromartire.anthony@nist.gov

National Institute of Standards and Technology

Intelligent Systems Division

Chapter 2

Todo List

Member `CanonicalRobotCommand::attach_eff (vector< string > paramList, KittingPlan *kittingplan)`

This function will be written once the **Canonical Robot Command** for the action **attach-eff** is implemented in ROS/USARSim

Member `CanonicalRobotCommand::create_kit (vector< string > paramList, KittingPlan *kittingplan)`

This function will be written once the **Canonical Robot Command** for the action **create-kit** is implemented

Member `CanonicalRobotCommand::put_kit (vector< string > paramList)`

This function will be written once the **Canonical Robot Command** for the action **put-kit** is implemented in ROS/USARSim

Member `CanonicalRobotCommand::put_kit_tray (vector< string > paramList)`

This function will be written once the **Canonical Robot Command** for the action **put-kit-tray** is implemented in ROS/USARSim

Member `CanonicalRobotCommand::remove_eff (vector< string > paramList)`

This function will be written once the **Canonical Robot Command** for the action **remove-eff** is implemented in ROS/USARSim

Member `CanonicalRobotCommand::take_kit (vector< string > paramList)`

This function will be written once the **Canonical Robot Command** for the action **take-kit** is implemented in ROS/USARSim

Member `CanonicalRobotCommand::take_kit_tray (vector< string > paramList)`

This function will be written once the **Canonical Robot Command** for the action **take-kit-tray** is implemented in ROS/USARSim

Member `CanonicalRobotCommand::take_part (vector< string > paramList, KittingPlan *kittingplan)`

This function will be written once the **Canonical Robot Command** for the action **take-part** is implemented in ROS/USARSim

Chapter 3

Module Index

3.1 Modules

Here is a list of all modules:

User-Level API	15
----------------	-------	----

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

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

CanonicalHdr	31
CanonicalMsg	32
CloseGripperMsg	57
DwellMsg	91
EndCanonMsg	94
InitCanonMsg	133
MessageMsg	194
MoveSmoothlyToMsg	196
MoveStraightToMsg	199
MoveToMsg	202
OpenGripperMsg	209
SetAbsoluteAccelerationMsg	312
SetAngleUnitsMsg	314
SetEndAngleToleranceMsg	317
SetEndPointToleranceMsg	320
SetIntermediatePointToleranceMsg	323
SetLengthUnitsMsg	326
SetRelativeAccelerationMsg	329
SetRelativeSpeedMsg	332
CanonicalRobotCommand	35
Connection	60
Controller	63
MyController	204
DAO	74
DataThing	84
BoxVolume	17
KitDesign	143
PartRefAndPose	221
PhysicalLocation	249
PoseLocation	265
PoseLocationIn	273
PoseLocationOn	278
PoseOnlyLocation	283

RelativeLocation	288
RelativeLocationIn	294
RelativeLocationOn	299
Point	256
ShapeDesign	335
StockKeepingUnit	353
Vector	379
FileOperator	117
IniFile	128
INIFILE_ENTRY	132
KittingPDDLProblem	150
KittingPlan	155
KitTrayLocStruct	172
PartLocStruct	219
PartsTrayLocStruct	240
SolidObject	342
BoxyObject	24
KitTray	165
LargeContainer	187
PartsBin	228
PartsTray	234
WorkTable	388
EndEffector	97
GripperEffector	123
VacuumEffector	362
VacuumEffectorMultiCup	368
VacuumEffectorSingleCup	374
EndEffectorChangingStation	105
EndEffectorHolder	111
Kit	135
KittingWorkstation	156
LargeBoxWithEmptyKitTrays	174
LargeBoxWithKits	180
Part	212
PartsTrayWithParts	242
Robot	304
unix_shm_struct	361

Chapter 5

Class Index

5.1 Class List

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

BoxVolume	17
BoxyObject	24
CanonicalHdr	31
CanonicalMsg	32
CanonicalRobotCommand	
This class provides functions to build Canonical Robot Commands from PDDL actions in the Plan	35
CloseGripperMsg	57
Connection	60
Controller	63
DAO	74
DataThing	84
DwellMsg	91
EndCanonMsg	94
EndEffector	97
EndEffectorChangingStation	105
EndEffectorHolder	111
FileOperator	
Class for operations on files	117
GripperEffector	123
IniFile	128
INIFILE_ENTRY	132
InitCanonMsg	133
Kit	135
KitDesign	143
KittingPDDLProblem	
Class for the Kitting PDDL Problem	150
KittingPlan	
Class for the Kitting Plan	155
KittingWorkstation	156
KitTray	165
KitTrayLocStruct	
This structure represents a kit tray location	172
LargeBoxWithEmptyKitTrays	174
LargeBoxWithKits	180

LargeContainer	187
MessageMsg	194
MoveSmoothlyToMsg	196
MoveStraightToMsg	199
MoveToMsg	202
MyController	204
OpenGripperMsg	209
Part	212
PartLocStruct Represents a part location	219
PartRefAndPose	221
PartsBin	228
PartsTray	234
PartsTrayLocStruct This structure represents a parts tray location	240
PartsTrayWithParts	242
PhysicalLocation	249
Point	256
PoseLocation	265
PoseLocationIn	273
PoseLocationOn	278
PoseOnlyLocation	283
RelativeLocation	288
RelativeLocationIn	294
RelativeLocationOn	299
Robot	304
SetAbsoluteAccelerationMsg	312
SetAngleUnitsMsg	314
SetEndAngleToleranceMsg	317
SetEndPointToleranceMsg	320
SetIntermediatePointToleranceMsg	323
SetLengthUnitsMsg	326
SetRelativeAccelerationMsg	329
SetRelativeSpeedMsg	332
ShapeDesign	335
SolidObject	342
StockKeepingUnit	353
unix_shm_struct	361
VacuumEffector	362
VacuumEffectorMultiCup	368
VacuumEffectorSingleCup	374
Vector	379
WorkTable	388

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/main.cc	493
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/canonicalMsg.hh	
Provide a generic class to base command processing routines for the canonical robot command language. This class will provide virtual routines that may be specialized for individual robot controllers	395
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/controller.cpp	398
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/controller.hh	
Provide a generic class to base executor controller modules off of. This class will provide routines to queue and dequeue commands as well as virtual functions for items such as emergency conditions and command handlers	399
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/myController.cpp	401
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/myController.hh	
Include file for myController class	401
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/ulapi.cpp	404
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/ulapi.hh	
Declarations of the user-level application programming interface, ULAPI, for portable communication between user-level processes and real-time tasks conforming to the RTAPI real-time API	410
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/BoxVolume.cpp	424
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/BoxVolume.h	424
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/BoxyObject.cpp	425
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/BoxyObject.h	425
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Connection.cpp	426
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Connection.h	427
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/DAO.cpp	427
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/DAO.h	428
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/DataThing.cpp	428
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/DataThing.h	428
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffector.cpp	429
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffector.h	430
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffectorChangingStation.cpp .	431
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffectorChangingStation.h .	431
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffectorHolder.cpp	432
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffectorHolder.h	432
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/GripperEffector.cpp	433
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/GripperEffector.h	434

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ Kit.cpp	435
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ Kit.h	435
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ KitDesign.cpp	436
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ KitDesign.h	437
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ KittingWorkstation.cpp	438
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ KittingWorkstation.h	438
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ KitTray.cpp	439
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ KitTray.h	440
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ LargeBoxWithEmptyKitTrays.cpp	441
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ LargeBoxWithEmptyKitTrays.h	441
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ LargeBoxWithKits.cpp	442
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ LargeBoxWithKits.h	442
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ LargeContainer.cpp	443
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ LargeContainer.h	444
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ Part.cpp	445
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ Part.h	445
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ PartRefAndPose.cpp	446
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ PartRefAndPose.h	447
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ PartsBin.cpp	448
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ PartsBin.h	448
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ PartsTray.cpp	449
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ PartsTray.h	449
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ PartsTrayWithParts.cpp	451
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ PartsTrayWithParts.h	451
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ PhysicalLocation.cpp	452
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ PhysicalLocation.h	452
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ Point.cpp	453
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ Point.h	454
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ PoseLocation.cpp	455
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ PoseLocation.h	455
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ PoseLocationIn.cpp	456
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ PoseLocationIn.h	456
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ PoseLocationOn.cpp	457
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ PoseLocationOn.h	458
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ PoseOnlyLocation.cpp	459
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ PoseOnlyLocation.h	459
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ RelativeLocation.cpp	461
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ RelativeLocation.h	461
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ RelativeLocationIn.cpp	462
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ RelativeLocationIn.h	462
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ RelativeLocationOn.cpp	464
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ RelativeLocationOn.h	464
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ Robot.cpp	465
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ Robot.h	465
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ ShapeDesign.cpp	466
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ ShapeDesign.h	467
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ SolidObject.cpp	468
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ SolidObject.h	468
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ StockKeepingUnit.cpp	469
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ StockKeepingUnit.h	470
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ VacuumEffector.cpp	471
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ VacuumEffector.h	471
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ VacuumEffectorMultiCup.cpp	472
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ VacuumEffectorMultiCup.h	472

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ VacuumEffectorSingleCup.cpp	474
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ VacuumEffectorSingleCup.h	474
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ Vector.cpp	475
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ Vector.h	475
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ WorkTable.cpp	476
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ WorkTable.h	477
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/ CanonicalRobotCommand.cc	478
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/ CanonicalRobotCommand.h	
Test	478
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/ Config.h	480
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/ IniFile.cpp	482
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/ IniFile.h	483
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/ KittingPDDLProblem.cc	486
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/ KittingPDDLProblem.h	486
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/ KittingPlan.cc	487
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/ KittingPlan.h	488
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/ Operator.cc	489
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/ Operator.h	490
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/ Structdef.h	
Structures used to define the location for different structures	491
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/ Tools.h	492

Chapter 7

Module Documentation

7.1 User-Level API

The User-Level Application Programming Interface (ULAPI) provides a portable interface for communicating with real-time tasks conforming to the RTAPI real-time API.

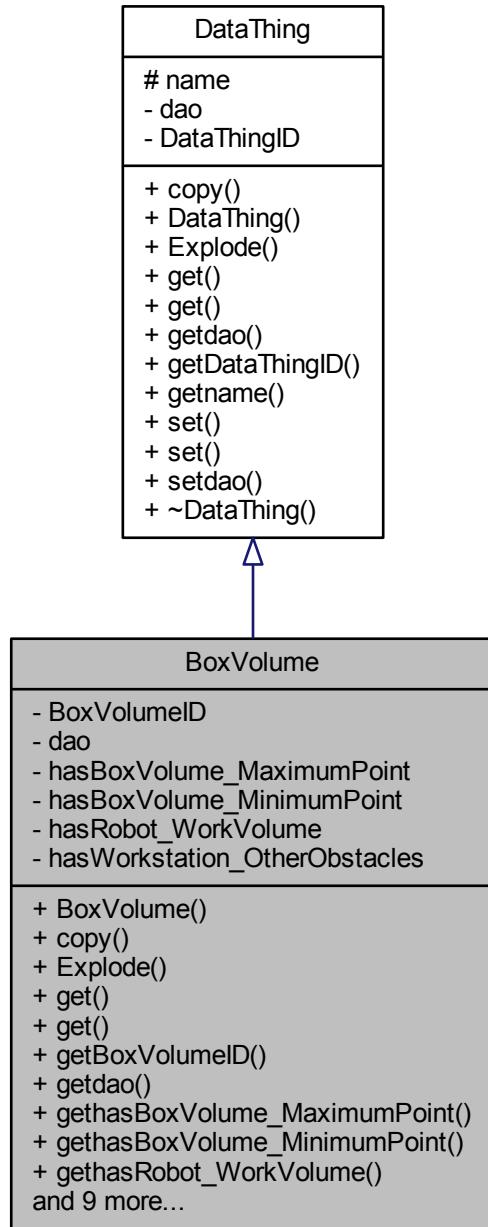
Chapter 8

Class Documentation

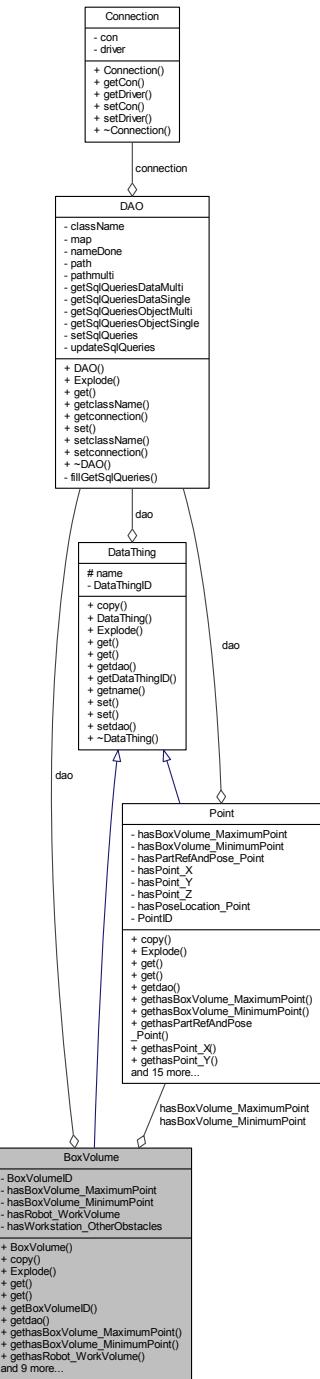
8.1 BoxVolume Class Reference

```
#include <BoxVolume.h>
```

Inheritance diagram for BoxVolume:



Collaboration diagram for BoxVolume:



Public Member Functions

- `BoxVolume (std::string name)`
- void `copy (std::map< std::string, std::string > object)`

- std::vector< std::string > [Explode](#) (const std::string &str, char separator)
- void [get](#) (int id)
- void [get](#) (std::string name)
- int [getBoxVolumeID](#) ()
- DAO * [getdao](#) ()
- Point * [gethasBoxVolume_MaximumPoint](#) ()
- Point * [gethasBoxVolume_MinimumPoint](#) ()
- std::vector< Robot * > [gethasRobot_WorkVolume](#) ()
- std::vector< KittingWorkstation * > [gethasWorkstation_OtherObstacles](#) ()
- void [set](#) (int id, BoxVolume *obj)
- void [set](#) (std::string name)
- void [setdao](#) (DAO *_dao)
- void [sethasBoxVolume_MaximumPoint](#) (Point *_hasBoxVolume_MaximumPoint)
- void [sethasBoxVolume_MinimumPoint](#) (Point *_hasBoxVolume_MinimumPoint)
- void [sethasRobot_WorkVolume](#) (std::vector< Robot * > _hasRobot_WorkVolume)
- void [sethasWorkstation_OtherObstacles](#) (std::vector< KittingWorkstation * > _hasWorkstation_OtherObstacles)
- [~BoxVolume](#) ()

Private Attributes

- int [BoxVolumeID](#)
- DAO * [dao](#)
- Point * [hasBoxVolume_MaximumPoint](#)
- Point * [hasBoxVolume_MinimumPoint](#)
- std::vector< Robot * > [hasRobot_WorkVolume](#)
- std::vector< KittingWorkstation * > [hasWorkstation_OtherObstacles](#)

Additional Inherited Members

8.1.1 Detailed Description

Definition at line 30 of file BoxVolume.h.

8.1.2 Constructor & Destructor Documentation

8.1.2.1 BoxVolume::BoxVolume (std::string name)

Definition at line 22 of file BoxVolume.cpp.

8.1.2.2 BoxVolume::~BoxVolume ()

Definition at line 27 of file BoxVolume.cpp.

8.1.3 Member Function Documentation

8.1.3.1 void BoxVolume::copy (std::map< std::string, std::string > *object*)

Definition at line 109 of file BoxVolume.cpp.

Here is the call graph for this function:



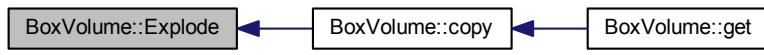
Here is the caller graph for this function:



8.1.3.2 std::vector< std::string > BoxVolume::Explode (const std::string & str, char separator)

Definition at line 136 of file BoxVolume.cpp.

Here is the caller graph for this function:

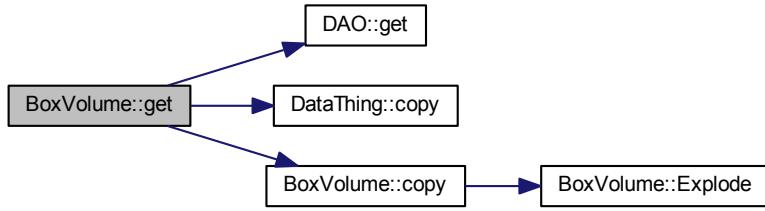


8.1.3.3 void BoxVolume::get (int id)

8.1.3.4 void BoxVolume::get (std::string name)

Definition at line 69 of file BoxVolume.cpp.

Here is the call graph for this function:



8.1.3.5 int BoxVolume::getBoxVolumeID ()

Definition at line 36 of file BoxVolume.cpp.

8.1.3.6 DAO * BoxVolume::getdao ()

Definition at line 39 of file BoxVolume.cpp.

8.1.3.7 Point * BoxVolume::gethasBoxVolume_MaximumPoint ()

Definition at line 42 of file BoxVolume.cpp.

8.1.3.8 Point * BoxVolume::gethasBoxVolume_MinimumPoint ()

Definition at line 45 of file BoxVolume.cpp.

8.1.3.9 std::vector< Robot * > BoxVolume::gethasRobot_WorkVolume ()

Definition at line 51 of file BoxVolume.cpp.

8.1.3.10 std::vector< KittingWorkstation * > BoxVolume::gethasWorkstation_OtherObstacles ()

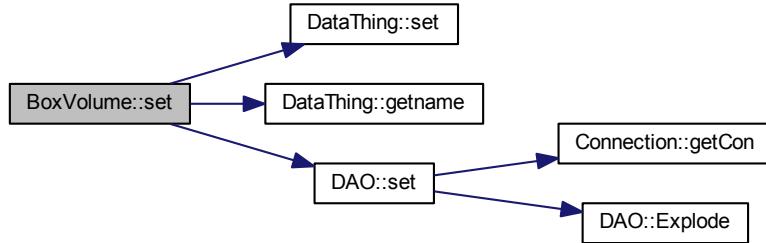
Definition at line 48 of file BoxVolume.cpp.

8.1.3.11 void BoxVolume::set (int id, BoxVolume * obj)

8.1.3.12 void BoxVolume::set (std::string name)

Definition at line 79 of file BoxVolume.cpp.

Here is the call graph for this function:



8.1.3.13 void BoxVolume::setdao (DAO * _dao)

Definition at line 54 of file BoxVolume.cpp.

8.1.3.14 void BoxVolume::sethasBoxVolume_MaximumPoint (Point * _hasBoxVolume_MaximumPoint)

Definition at line 57 of file BoxVolume.cpp.

8.1.3.15 void BoxVolume::sethasBoxVolume_MinimumPoint (Point * _hasBoxVolume_MinimumPoint)

Definition at line 60 of file BoxVolume.cpp.

8.1.3.16 void BoxVolume::sethasRobot_WorkVolume (std::vector< Robot * > _hasRobot_WorkVolume)

Definition at line 66 of file BoxVolume.cpp.

8.1.3.17 void BoxVolume::sethasWorkstation_OtherObstacles (std::vector< KittingWorkstation * > _hasWorkstation_OtherObstacles)

Definition at line 63 of file BoxVolume.cpp.

8.1.4 Member Data Documentation

8.1.4.1 int BoxVolume::BoxVolumeID [private]

Definition at line 31 of file BoxVolume.h.

8.1.4.2 DAO* BoxVolume::dao [private]

Definition at line 32 of file BoxVolume.h.

8.1.4.3 Point* BoxVolume::hasBoxVolume_MaximumPoint [private]

Definition at line 33 of file BoxVolume.h.

8.1.4.4 Point* BoxVolume::hasBoxVolume_MinimumPoint [private]

Definition at line 34 of file BoxVolume.h.

8.1.4.5 std::vector<Robot*> BoxVolume::hasRobot_WorkVolume [private]

Definition at line 36 of file BoxVolume.h.

8.1.4.6 std::vector<KittingWorkstation*> BoxVolume::hasWorkstation_OtherObstacles [private]

Definition at line 35 of file BoxVolume.h.

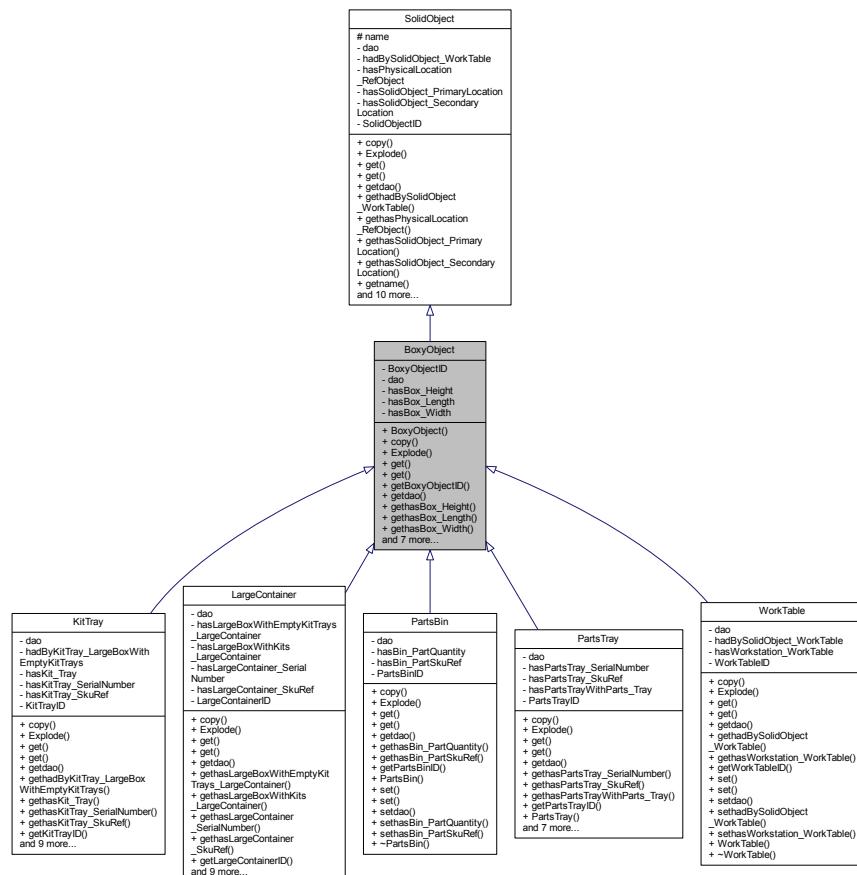
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[BoxVolume.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[BoxVolume.cpp](#)

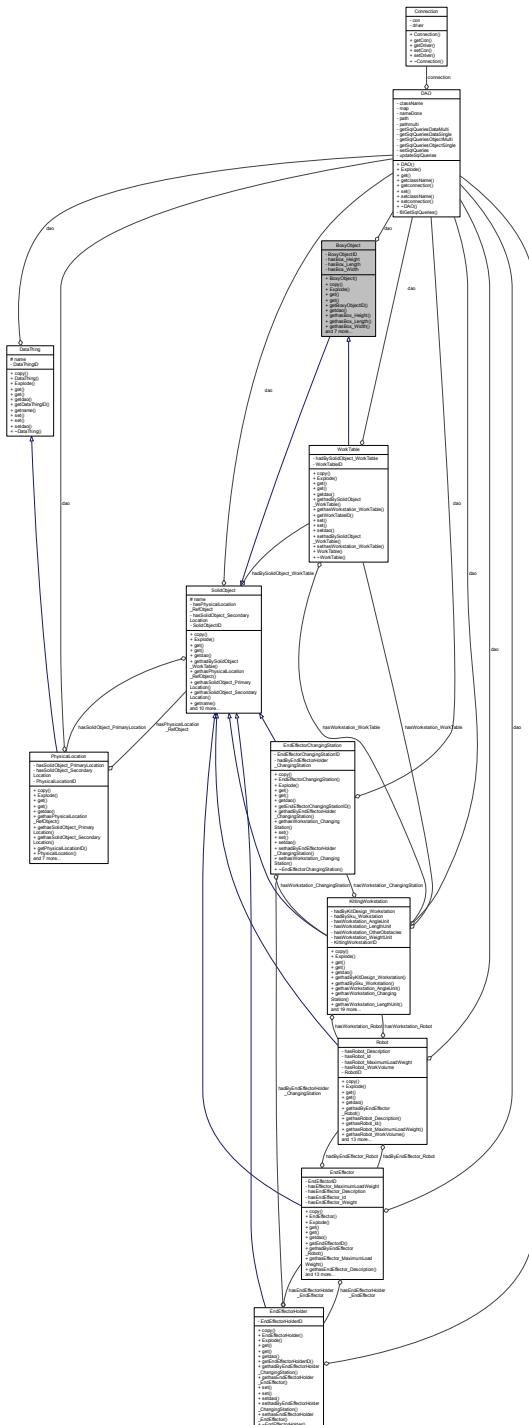
8.2 BoxyObject Class Reference

```
#include <BoxyObject.h>
```

Inheritance diagram for BoxyObject:



Collaboration diagram for BoxyObject:



Public Member Functions

- **BoxyObject (std::string name)**
- void **copy (std::map< std::string, std::string > object)**

- std::vector< std::string > [Explode](#) (const std::string &str, char separator)
- void [get](#) (int id)
- void [get](#) (std::string name)
- int [getBoxyObjectID](#) ()
- DAO * [getdao](#) ()
- double [gethasBox_Height](#) ()
- double [gethasBox_Length](#) ()
- double [gethasBox_Width](#) ()
- void [set](#) (int id, [BoxyObject](#) *obj)
- void [set](#) (std::string name)
- void [setdao](#) (DAO *_dao)
- void [sethasBox_Height](#) (double _hasBox_Height)
- void [sethasBox_Length](#) (double _hasBox_Length)
- void [sethasBox_Width](#) (double _hasBox_Width)
- ~[BoxyObject](#) ()

Private Attributes

- int [BoxyObjectID](#)
- DAO * [dao](#)
- double [hasBox_Height](#)
- double [hasBox_Length](#)
- double [hasBox_Width](#)

Additional Inherited Members

8.2.1 Detailed Description

Definition at line 27 of file BoxyObject.h.

8.2.2 Constructor & Destructor Documentation

8.2.2.1 BoxyObject::BoxyObject (std::string name)

Definition at line 19 of file BoxyObject.cpp.

8.2.2.2 BoxyObject::~BoxyObject ()

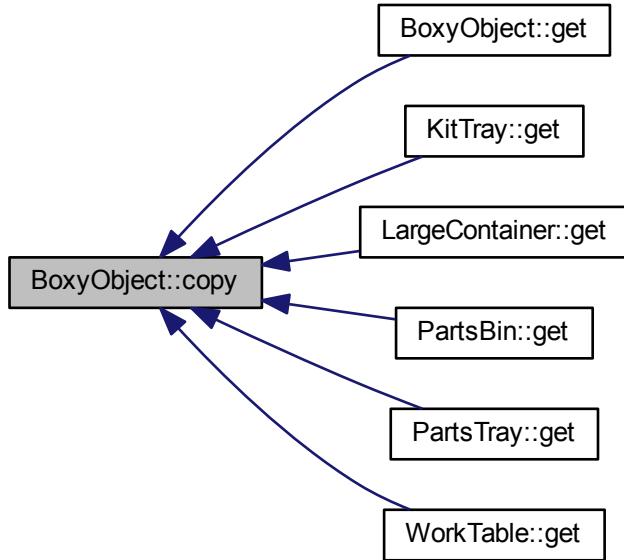
Definition at line 22 of file BoxyObject.cpp.

8.2.3 Member Function Documentation

8.2.3.1 void BoxyObject::copy (std::map< std::string, std::string > object)

Definition at line 85 of file BoxyObject.cpp.

Here is the caller graph for this function:



8.2.3.2 `std::vector< std::string > BoxyObject::Explode (const std::string & str, char separator)`

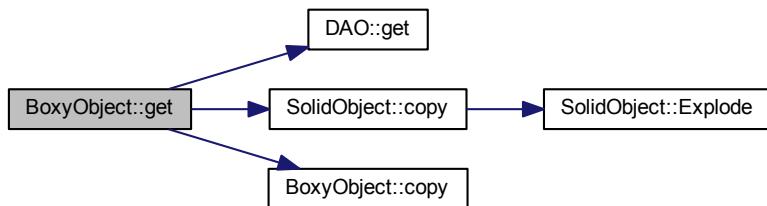
Definition at line 97 of file BoxyObject.cpp.

8.2.3.3 `void BoxyObject::get (int id)`

8.2.3.4 `void BoxyObject::get (std::string name)`

Definition at line 52 of file BoxyObject.cpp.

Here is the call graph for this function:



8.2.3.5 int BoxyObject::getBoxyObjectID ()

Definition at line 34 of file BoxyObject.cpp.

8.2.3.6 DAO * BoxyObject::getdao ()

Definition at line 37 of file BoxyObject.cpp.

8.2.3.7 double BoxyObject::gethasBox_Height ()

Definition at line 25 of file BoxyObject.cpp.

8.2.3.8 double BoxyObject::gethasBox_Length ()

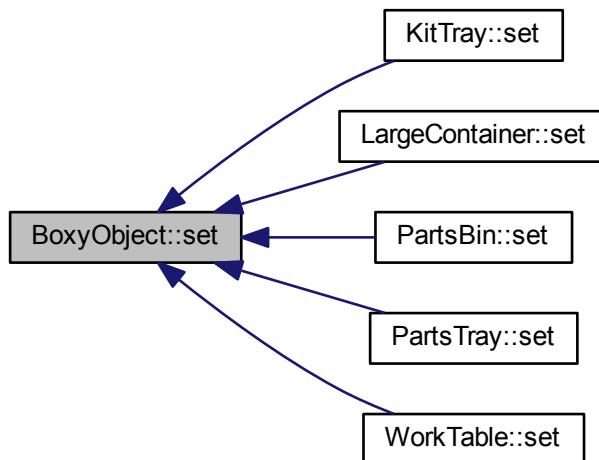
Definition at line 31 of file BoxyObject.cpp.

8.2.3.9 double BoxyObject::gethasBox_Width ()

Definition at line 28 of file BoxyObject.cpp.

8.2.3.10 void BoxyObject::set (int *id*, BoxyObject * *obj*)

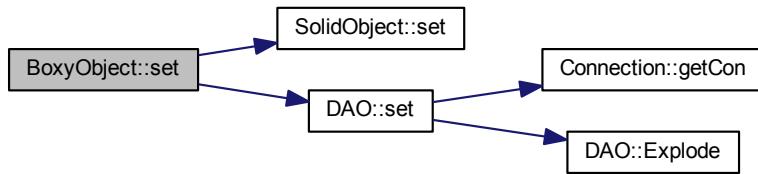
Here is the caller graph for this function:



8.2.3.11 void BoxyObject::set (std::string name)

Definition at line 62 of file BoxyObject.cpp.

Here is the call graph for this function:



8.2.3.12 void BoxyObject::setdao (DAO * _dao)

Definition at line 49 of file BoxyObject.cpp.

8.2.3.13 void BoxyObject::sethasBox_Height (double _hasBox_Height)

Definition at line 40 of file BoxyObject.cpp.

8.2.3.14 void BoxyObject::sethasBox_Length (double _hasBox_Length)

Definition at line 46 of file BoxyObject.cpp.

8.2.3.15 void BoxyObject::sethasBox_Width (double _hasBox_Width)

Definition at line 43 of file BoxyObject.cpp.

8.2.4 Member Data Documentation

8.2.4.1 int BoxyObject::BoxyObjectID [private]

Definition at line 31 of file BoxyObject.h.

8.2.4.2 DAO* BoxyObject::dao [private]

Definition at line 32 of file BoxyObject.h.

8.2.4.3 double BoxyObject::hasBox_Height [private]

Definition at line 28 of file BoxyObject.h.

8.2.4.4 double BoxyObject::hasBox_Length [private]

Definition at line 30 of file BoxyObject.h.

8.2.4.5 double BoxyObject::hasBox_Width [private]

Definition at line 29 of file BoxyObject.h.

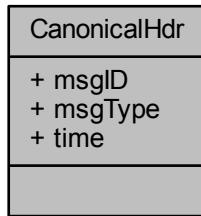
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[BoxyObject.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[BoxyObject.cpp](#)

8.3 CanonicalHdr Struct Reference

```
#include <canonicalMsg.hh>
```

Collaboration diagram for CanonicalHdr:



Public Attributes

- unsigned int [msgID](#)
- [CanonicalType](#) [msgType](#)
- double [time](#)

8.3.1 Detailed Description

Definition at line 46 of file canonicalMsg.hh.

8.3.2 Member Data Documentation

8.3.2.1 unsigned int CanonicalHdr::msgID

Definition at line 49 of file canonicalMsg.hh.

8.3.2.2 CanonicalType CanonicalHdr::msgType

Definition at line 48 of file canonicalMsg.hh.

8.3.2.3 double CanonicalHdr::time

Definition at line 50 of file canonicalMsg.hh.

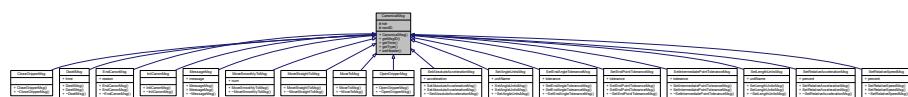
The documentation for this struct was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controllercanonicalMsg.hh

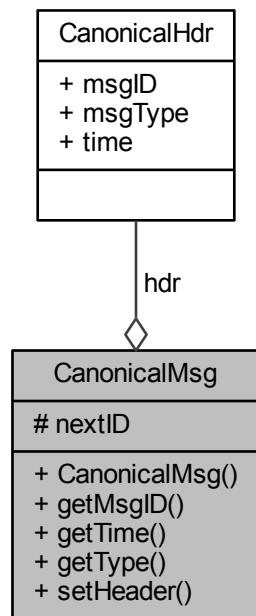
8.4 CanonicalMsg Class Reference

```
#include <canonicalMsq.hh>
```

Inheritance diagram for CanonicalMsg:



Collaboration diagram for CanonicalMsg:



Public Member Functions

- CanonicalMsg (CanonicalType msgType)
- int [getMsgID \(\)](#)
- double [getTime \(\)](#)
- CanonicalType [getType \(\)](#)
- void [setHeader \(\)](#)

Protected Attributes

- CanonicalHdr [hdr](#)

Static Protected Attributes

- static int [nextID](#) = 1

8.4.1 Detailed Description

Definition at line 53 of file canonicalMsg.hh.

8.4.2 Constructor & Destructor Documentation

8.4.2.1 CanonicalMsg::CanonicalMsg (CanonicalType *msgType*) [inline]

Definition at line 55 of file canonicalMsg.hh.

8.4.3 Member Function Documentation

8.4.3.1 int CanonicalMsg::getMsgID () [inline]

Definition at line 57 of file canonicalMsg.hh.

Here is the caller graph for this function:



8.4.3.2 double CanonicalMsg::getTime () [inline]

Definition at line 58 of file canonicalMsg.hh.

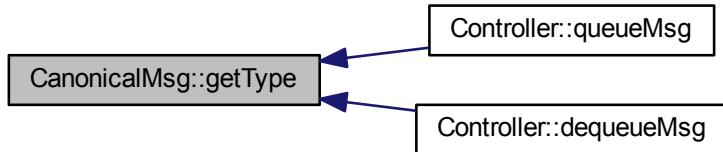
Here is the caller graph for this function:



8.4.3.3 CanonicalType CanonicalMsg::getType() [inline]

Definition at line 59 of file canonicalMsg.hh.

Here is the caller graph for this function:



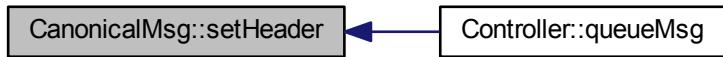
8.4.3.4 void CanonicalMsg::setHeader() [inline]

Definition at line 56 of file canonicalMsg.hh.

Here is the call graph for this function:



Here is the caller graph for this function:



8.4.4 Member Data Documentation

8.4.4.1 CanonicalHdr CanonicalMsg::hdr [protected]

Definition at line 62 of file canonicalMsg.hh.

8.4.4.2 int CanonicalMsg::nextID = 1 [static], [protected]

Definition at line 59 of file canonicalMsg.hh.

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

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[canonicalMsg.hh](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[controller.cpp](#)

8.5 CanonicalRobotCommand Class Reference

This class provides functions to build **Canonical Robot Commands** from PDDL actions in the Plan.

```
#include <CanonicalRobotCommand.h>
```

Collaboration diagram for CanonicalRobotCommand:

CanonicalRobotCommand
- m_dwell - m_kit_tray - m_safe_z
+ actionInterpreter() + attach_eff() + canon_put_part() + canon_take_part() + CanonicalRobotCommand() + create_kit() + interpretPlan() + put_kit() + put_kit_tray() + put_part() + remove_eff() + take_kit() + take_kit_tray() + take_part() + ~CanonicalRobotCommand() - getKitTrayLocation() - getPartLocation() - getPartsTrayLocation() - print_closegripper() - print_dwell() - print_endcannon() - print_initcannon() - print_moveto() - print_opengripper()

Public Member Functions

- void [actionInterpreter](#) (string action, vector< string > paramName, [KittingPlan](#) *kittingplan)
Match each action found in the plan with a function.
- void [attach_eff](#) (vector< string > paramList, [KittingPlan](#) *kittingplan)
*A canonical robot command for the PDDL action **attach-eff***
- void [canon_put_part](#) (vector< double > xyz, [Vector](#) *z_axis, [Vector](#) *x_axis)
*Compile a set of **Canonical Robot Commands** to perform the PDDL action **put-part***
- void [canon_take_part](#) (vector< double > xyz, [Vector](#) *z_axis, [Vector](#) *x_axis)
*Generate the canonical robot commands for the PDDL action **take-part***
- [CanonicalRobotCommand](#) ()
Constructor.
- void [create_kit](#) (vector< string > paramList, [KittingPlan](#) *kittingplan)

- A canonical robot command for the PDDL action **create-kit**
- void `interpretPlan (KittingPlan *kittingplan)`
 - Read the plan stored in KittingPlan::m_actionParamList and interpret each action. The different steps are:*
- void `put_kit (vector< string > paramList)`
 - A canonical robot command for the PDDL action **put-kit**
- void `put_kit_tray (vector< string > paramList)`
 - A canonical robot command for the PDDL action **put-kit-tray**
- void `put_part (vector< string > paramList, KittingPlan *kittingplan)`
 - A canonical robot command for the PDDL action **put-part**
- void `remove_eff (vector< string > paramList)`
 - A canonical robot command for the PDDL action **remove-eff**
- void `take_kit (vector< string > paramList)`
 - A canonical robot command for the PDDL action **take-kit**
- void `take_kit_tray (vector< string > paramList)`
 - A canonical robot command for the PDDL action **take-kit-tray**
- void `take_part (vector< string > paramList, KittingPlan *kittingplan)`
 - A canonical robot command for the PDDL action **take-part**
- virtual ~CanonicalRobotCommand ()
Auto-generated destructor stub.

Private Member Functions

- `KitTrayLocStruct getKitTrayLocation (string kit_tray_name)`
 - Retrieve the location of the kit tray kit_tray_name.*
- `PartLocStruct getPartLocation (string part_name)`
 - Retrieve the location of the part part_name.*
- `PartsTrayLocStruct getPartsTrayLocation (string parts_tray_name)`
 - Retrieve the location of the parts tray part_tray_name.*
- void `print_closegripper ()`
 - The CloseGripper command.*
- void `print_dwell (double time)`
 - The Dwell command.*
- void `print_endcannon (int id)`
 - The EndCanon command.*
- void `print_initcannon ()`
 - The InitCanon command.*
- void `print_moveto (double x, double y, double z, Vector *z_axis, Vector *x_axis)`
 - The MoveTo command.*
- void `print.opengripper ()`
 - The OpenGripper command.*

Private Attributes

- double `m_dwell`
- string `m_kit_tray`
- double `m_safe_z`

8.5.1 Detailed Description

This class provides functions to build **Canonical Robot Commands** from PDDL actions in the Plan.

Author

Zeid Kootbally zeid.kootbally@nist.gov

Date

May 17, 2012

Definition at line 52 of file CanonicalRobotCommand.h.

8.5.2 Constructor & Destructor Documentation

8.5.2.1 CanonicalRobotCommand::CanonicalRobotCommand ()

Constructor.

Definition at line 19 of file CanonicalRobotCommand.cc.

8.5.2.2 CanonicalRobotCommand::~CanonicalRobotCommand () [virtual]

Auto-generated destructor stub.

Definition at line 26 of file CanonicalRobotCommand.cc.

8.5.3 Member Function Documentation

8.5.3.1 void CanonicalRobotCommand::actionInterpreter (string *action_name*, vector< string > *paramList*, KittingPlan * *kittingplan*)

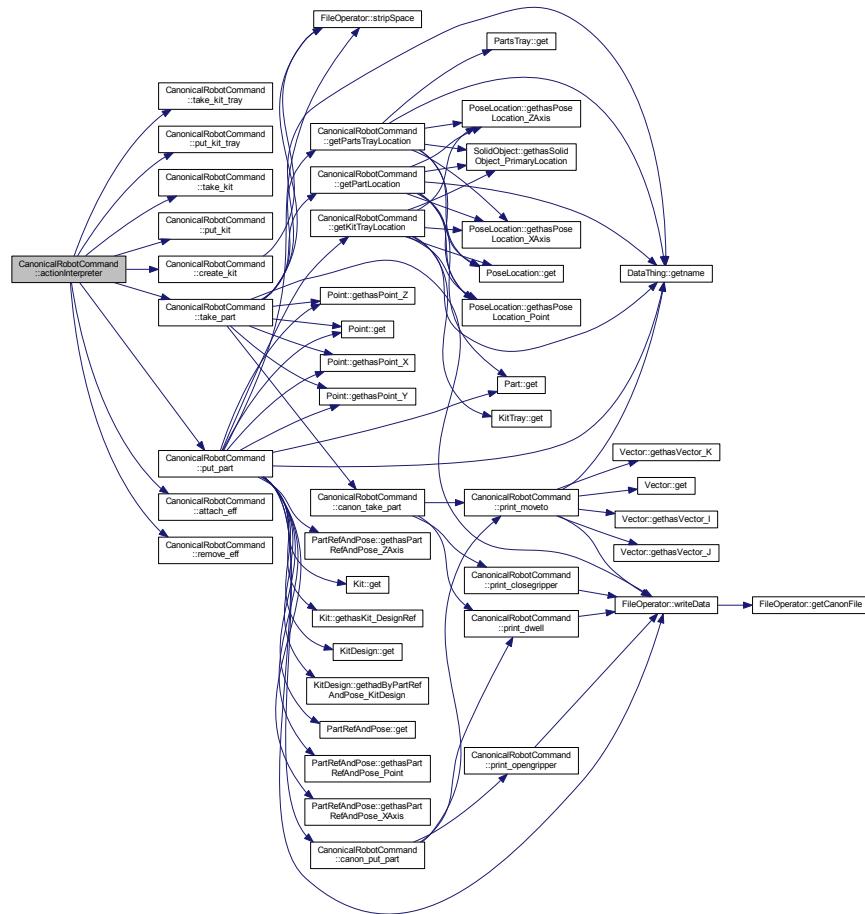
Match each action found in the plan with a function.

Parameters

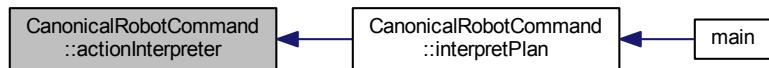
<i>action_name</i>	Action from the plan
<i>paramList</i>	List of parameters for the action <i>action_name</i>
<i>kittingplan</i>	Instance of KittingPlan

Definition at line 36 of file CanonicalRobotCommand.cc.

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.3.2 void CanonicalRobotCommand::attach_eff (`vector< string > paramList, KittingPlan * kittingplan`)

A canonical robot command for the PDDL action **attach-eff**

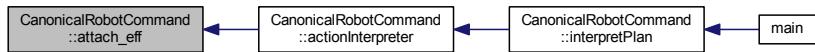
Parameters

<i>paramList</i>	List of parameters for the action attach-eff
<i>kittingplan</i>	Instance of KittingPlan

Todo This function will be written once the **Canonical Robot Command** for the action **attach-eff** is implemented in ROS/USARSim

Definition at line 57 of file CanonicalRobotCommand.cc.

Here is the caller graph for this function:



8.5.3.3 void CanonicalRobotCommand::canon_put_part (vector< double > xyz, Vector * z_axis, Vector * x_axis)

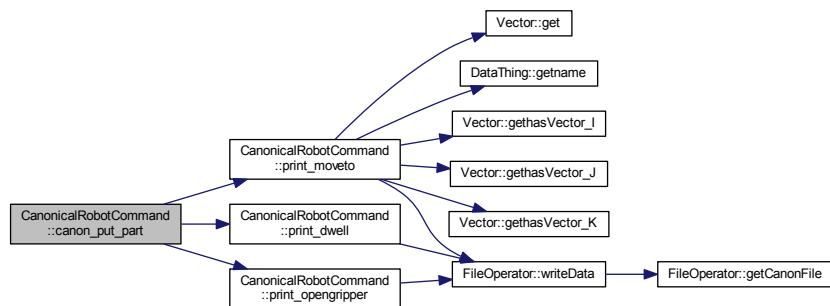
Compile a set of **Canonical Robot Commands** to perform the PDDL action **put-part**

Parameters

<i>xyz</i>	Vector that contains the coordinates of the part in the kit
<i>z_axis</i>	Z axis used by the robot to put the part in the kit
<i>x_axis</i>	X axis used by the robot to put the part in the kit

Definition at line 66 of file CanonicalRobotCommand.cc.

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.3.4 void CanonicalRobotCommand::canon_take_part (vector< double > xyz, Vector * z_axis, Vector * x_axis)

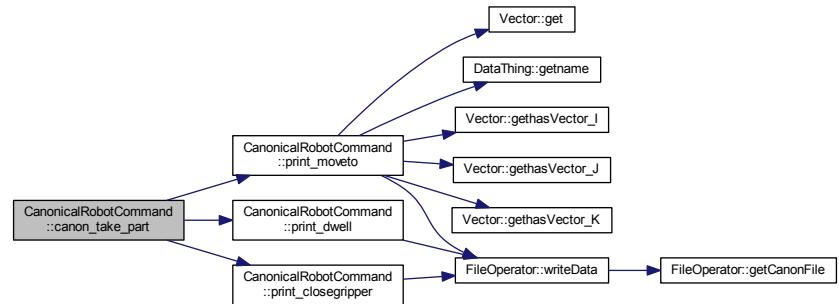
Generate the canonical robot commands for the PDDL action **take-part**

Parameters

<i>xyz</i>	<code>Vector</code> that contains the coordinates of the part in the parts tray
<i>z_axis</i>	Z axis used by the robot to take the part from the parts tray
<i>x_axis</i>	X axis used by the robot to take the part from the parts tray

Definition at line 86 of file CanonicalRobotCommand.cc.

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.3.5 void CanonicalRobotCommand::create_kit (vector< string > paramList, KittingPlan * kittingplan)

A canonical robot command for the PDDL action **create-kit**

Parameters

<i>paramList</i>	List of parameters for the action create-kit
------------------	---

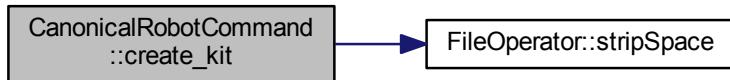
The data to be inserted in the table **Kit** are:

- KitID: ID of the new **Kit**.
- _NAME: Name of the new **Kit**.
- hasKit_DesignRef: **Kit** design associated to this **Kit**.
- isKit_Finished: 1 if the **Kit** is finished, 0 otherwise.
- hadByKit_LargeBoxWithKits: The **LargeBoxWithKits** that is capable of containing this **Kit**.
- hasKit_Tray: **Kit** Tray associated to this **Kit**.

Todo This function will be written once the **Canonical Robot Command** for the action **create-kit** is implemented

Definition at line 115 of file CanonicalRobotCommand.cc.

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.3.6 KitTrayLocStruct CanonicalRobotCommand::getKitTrayLocation (string *kit_tray_name*) [private]

Retrieve the location of the kit tray *kit_tray_name*.

Parameters

<i>kit_tray_name</i>	Name of the kit tray
----------------------	----------------------

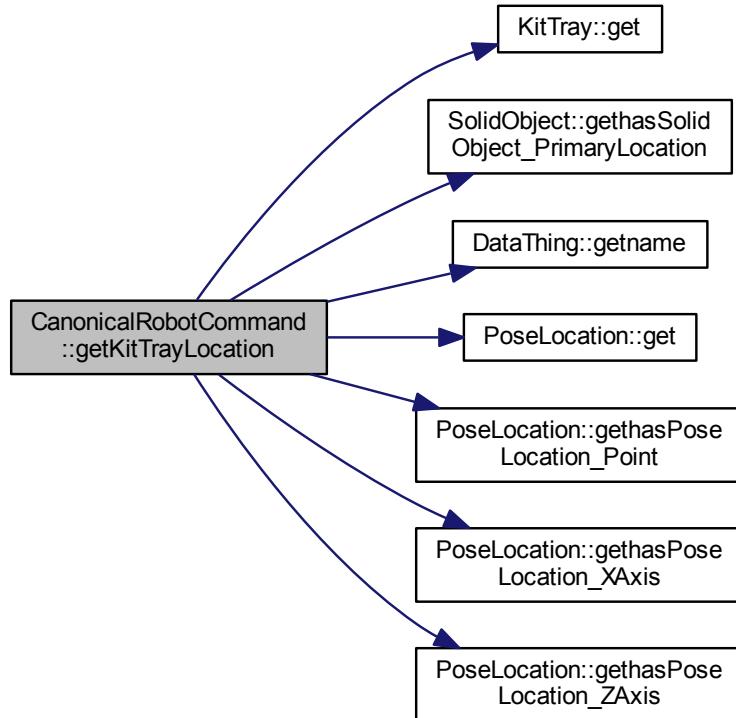
Returns

`KitTrayLocStruct` Pose (`Point`), the Z axis (`Vector`), and the X axis (`Vector`) of the kit tray `kit_tray_name`

1. Query the field from the table `PoseLocation` to retrieve the pose location for the kit tray
2. Query the field `hasPoseLocation_Point` from the table `PoseLocation` to retrieve the name of the point for the kit tray
 - Query the table `Point` to retrieve the coordinates of the parts tray
3. Query the field `hasPoseLocation_XAxis` from the table `PoseLocation` to retrieve the name of the X axis for the kit tray
 - Query the table `Vector` to retrieve the vector for the X axis
4. Query the field `hasPoseLocation_ZAxis` from the table `PoseLocation` to retrieve the name of the Z axis for the kit tray
 - Query the table `Vector` to retrieve the vector for the Z axis

Definition at line 393 of file CanonicalRobotCommand.cc.

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.3.7 PartLocStruct CanonicalRobotCommand::getPartLocation (string *part_name*) [private]

Retrieve the location of the part *part_name*.

Parameters

<i>partName</i>	Name of the part
-----------------	------------------

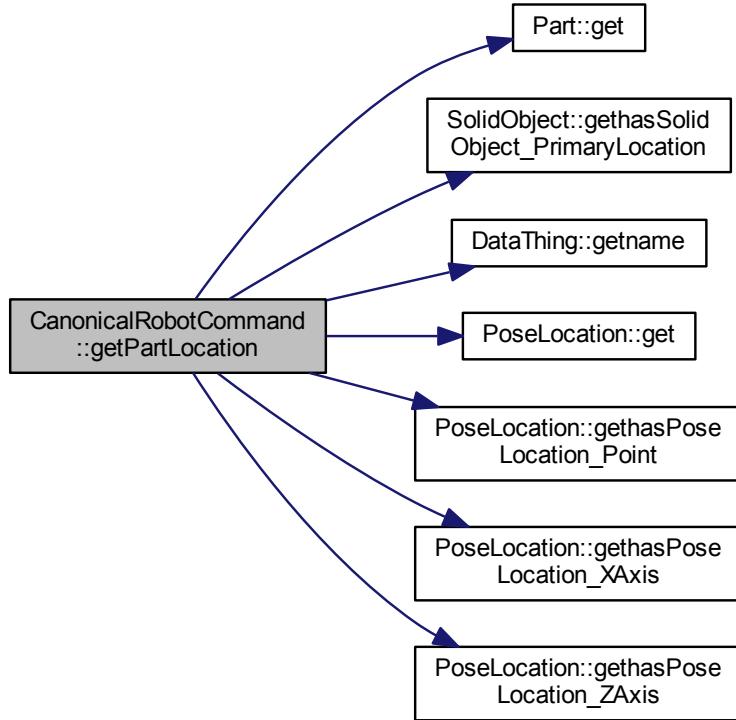
Returns

PartLocStruct Pose ([Point](#)), the Z axis ([Vector](#)), and the X axis ([Vector](#)) of the part *part_name*

1. Query the field *hasSolidObject_PrimaryLocation* from the table [SolidObject](#) to retrieve the name of the pose for the part
2. Query the field *hasPoseLocation_Point* from the table [PoseLocation](#) to retrieve the name of the point for the part
 - Query the table [Point](#) to retrieve the coordinates of the part
3. Query the field *hasPoseLocation_XAxis* from the table [PoseLocation](#) to retrieve the name of the X axis for the part
 - Query the table [Vector](#) to retrieve the vector for the X axis
4. Query the field *hasPoseLocation_ZAxis* from the table [PoseLocation](#) to retrieve the name of the Z axis for the part
 - Query the table [Vector](#) to retrieve the vector for the Z axis

Definition at line 467 of file CanonicalRobotCommand.cc.

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.3.8 PartsTrayLocStruct CanonicalRobotCommand::getPartsTrayLocation (string *part_tray_name*) [private]

Retrieve the location of the parts tray *part_tray_name*.

Parameters

<i>part_tray_name</i>	Name of the parts tray
-----------------------	------------------------

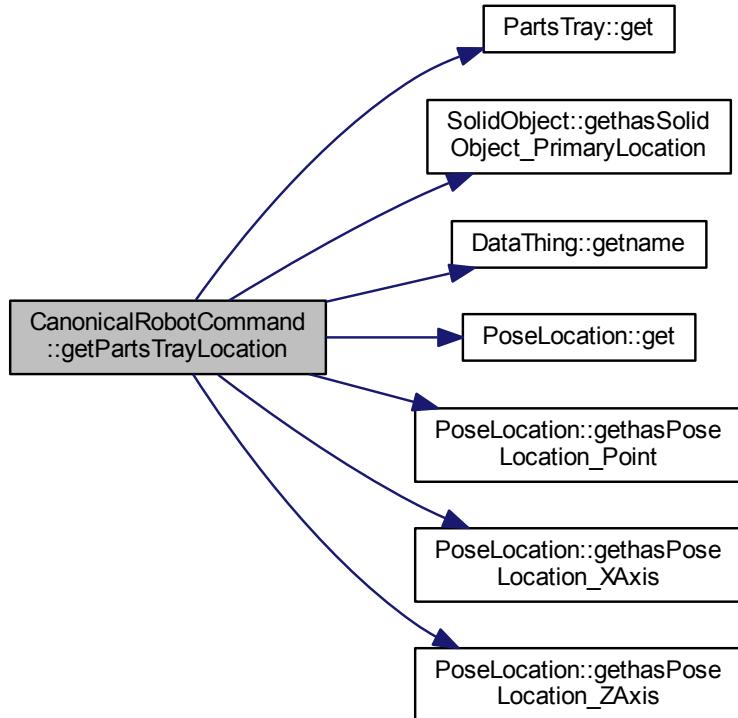
Returns

`PartsTrayLocStruct` Pose (`Point`), the Z axis (`Vector`), and the X axis (`Vector`) of the parts tray `part_tray_name`

1. Query the field `hasPoseLocation_Point` from the table `PoseLocation` to retrieve the name of the point for the parts tray
 - Query the table `Point` to retrieve the coordinates of the parts tray
2. Query the field `hasPoseLocation_XAxis` from the table `PoseLocation` to retrieve the name of the X axis for the parts tray
 - Query the table `Vector` to retrieve the vector for the X axis
3. Query the field `hasPoseLocation_ZAxis` from the table `PoseLocation` to retrieve the name of the Z axis for the parts tray
 - Query the table `Vector` to retrieve the vector for the Z axis

Definition at line 429 of file CanonicalRobotCommand.cc.

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.3.9 void CanonicalRobotCommand::interpretPlan (*KittingPlan* * *kittingplan*)

Read the plan stored in KittingPlan::m_actionParamList and interpret each action. The different steps are:

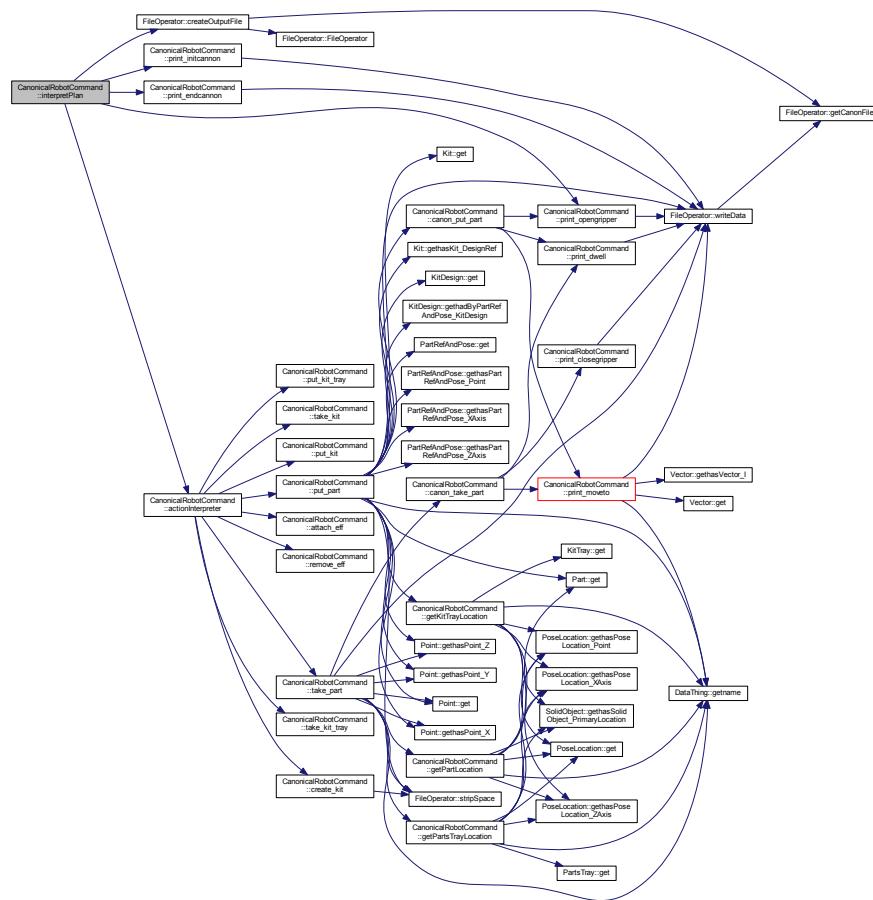
- Create the output file that will contain the canonical robot commands
- Write the command InitCanon () in the output file
- Write the command OpenGripper () in the output file
- Retrieve each action and their parameters from KittingPlan::m_actionParamList and call the appropriate functions to print the robot commands in the output file
- Write the command EndCanon () in the output file

Parameters

<i>kittingplan</i>	Instance of KittingPlan
--------------------	---

Definition at line 148 of file CanonicalRobotCommand.cc.

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.3.10 void CanonicalRobotCommand::print_closegripper() [private]

The **CloseGripper** command.

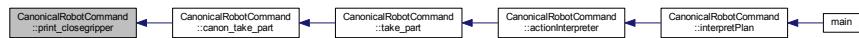
Close the gripper.

Definition at line 499 of file CanonicalRobotCommand.cc.

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.3.11 void CanonicalRobotCommand::print_dwell (double time) [private]

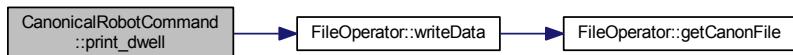
The **Dwell** command.

Parameters

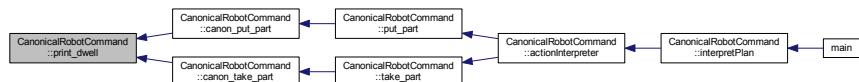
<i>time</i>	Stay motionless for the given amount of <i>time</i> in seconds
-------------	--

Definition at line 510 of file CanonicalRobotCommand.cc.

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.3.12 void CanonicalRobotCommand::print_endcannon (int reason) [private]

The **EndCanon** command.

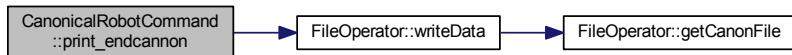
Do whatever is necessary to stop executing canonical robot commands. No specific action is required. The robot controller should not execute any canonical robot command except InitCanon after executing EndCanon and should signal an error if it is given one. This command will normally be given when execution of a plan is complete. It may also be given if the plan interpreter detects an error in the plan or is unable to proceed for any other reason. A value of 0 for reason indicates that execution of a plan has completed successfully. A positive value of reason indicates not.

Parameters

<code>reason</code>	Reason used to indicate if the execution of a plan has been completed or not
---------------------	--

Definition at line 535 of file CanonicalRobotCommand.cc.

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.3.13 void CanonicalRobotCommand::print_initcannon() [private]

The **InitCanon** command.

Do whatever is necessary to get ready to move. Length and angle units are set to the default units. This command will normally be given when the plan interpreter opens a plan to be executed.

Definition at line 552 of file CanonicalRobotCommand.cc.

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.3.14 void CanonicalRobotCommand::print_moveto (double part_point_x, double part_point_y, double part_point_z, Vector * z_axis, Vector * x_axis) [private]

The **MoveTo** command.

Move the controlled point along any convenient trajectory from the current pose to the given pose, and stop there.

Before generating the **MoveTo** command, the two following steps are performed:

- Retrieve the *part_z_axis_i*, *part_z_axis_j*, and *part_z_axis_k* components for the Z axis from *z_axis*
- Retrieve the *part_x_axis_i*, *part_x_axis_j*, and *part_x_axis_k* components for the X axis from *x_axis*

The structure of the **MoveTo** command is then built as follows:

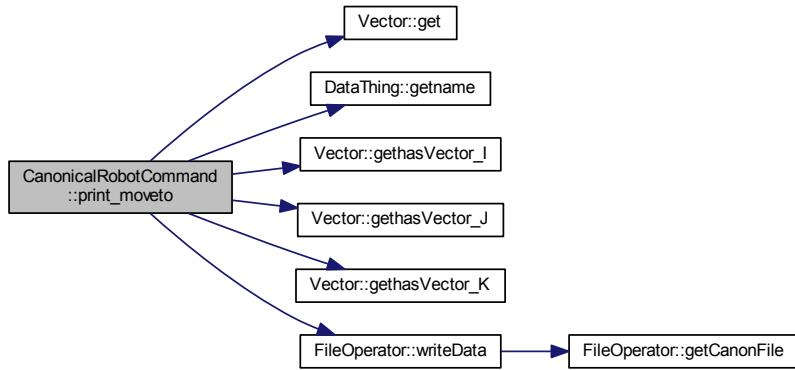
- **MoveTo({{part_point_x, part_point_y, part_point_z},{part_z_axis_i, part_z_axis_j, part_z_axis_k},{part_x_axis_i, part_x_axis_j, part_x_axis_k}}}**

Parameters

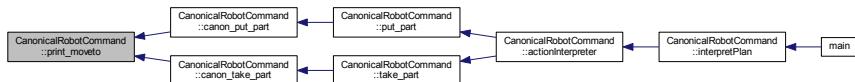
<i>part_point_x</i>	X coordinate of the given pose
<i>part_point_y</i>	Y coordinate of the given pose
<i>part_point_z</i>	Z coordinate of the given pose
<i>z_axis</i>	Z axis of the given pose
<i>x_axis</i>	X axis of the given pose

Definition at line 581 of file CanonicalRobotCommand.cc.

Here is the call graph for this function:



Here is the caller graph for this function:



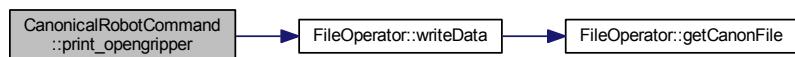
8.5.3.15 void CanonicalRobotCommand::print_opengripper() [private]

The **OpenGripper** command.

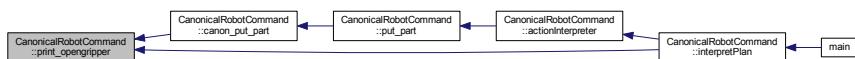
Open the gripper.

Definition at line 656 of file CanonicalRobotCommand.cc.

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.3.16 void CanonicalRobotCommand::put_kit (vector< string > *paramList*)

A canonical robot command for the PDDL action **put-kit**

Parameters

<i>paramList</i>	List of parameters for the action put-kit
------------------	--

Todo This function will be written once the **Canonical Robot Command** for the action **put-kit** is implemented in ROS/USARSim

Definition at line 175 of file CanonicalRobotCommand.cc.

Here is the caller graph for this function:



8.5.3.17 void CanonicalRobotCommand::put_kit_tray (vector< string > *paramList*)

A canonical robot command for the PDDL action **put-kit-tray**

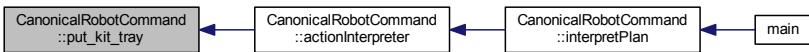
Parameters

<i>paramList</i>	List of parameters for the action put-kit-tray
------------------	---

Todo This function will be written once the **Canonical Robot Command** for the action **put-kit-tray** is implemented in ROS/USARSim

Definition at line 185 of file CanonicalRobotCommand.cc.

Here is the caller graph for this function:



8.5.3.18 void CanonicalRobotCommand::put_part (vector< string > *paramList*, KittingPlan * *kittingplan*)

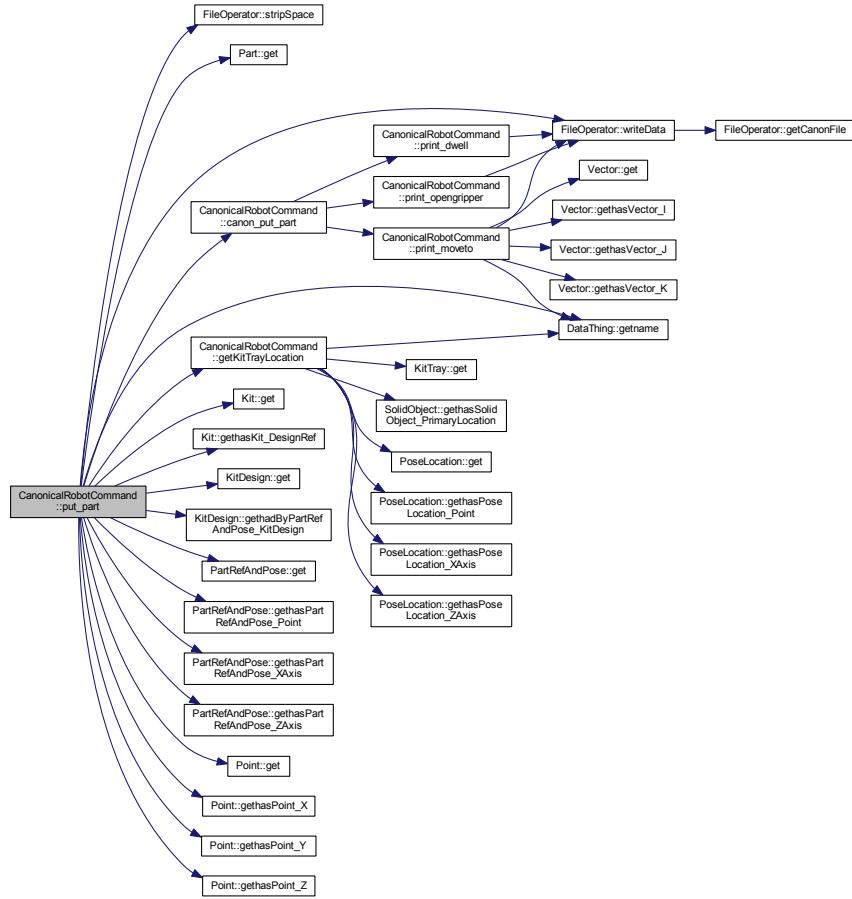
A canonical robot command for the PDDL action **put-part**

Parameters

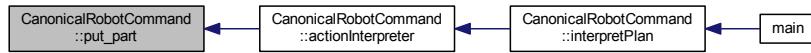
<i>paramList</i>	List of parameters for the action put-part
<i>kittingplan</i>	Instance of KittingPlan

Definition at line 195 of file CanonicalRobotCommand.cc.

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.3.19 void CanonicalRobotCommand::remove_eff (vector< string > paramList)

A canonical robot command for the PDDL action **remove-eff**

Parameters

<code>paramList</code>	List of parameters for the action remove-eff
------------------------	---

Todo This function will be written once the **Canonical Robot Command** for the action **remove-eff** is implemented in ROS/USARSim

Definition at line 295 of file CanonicalRobotCommand.cc.

Here is the caller graph for this function:



8.5.3.20 void CanonicalRobotCommand::take_kit (vector< string > paramList)

A canonical robot command for the PDDL action **take-kit**

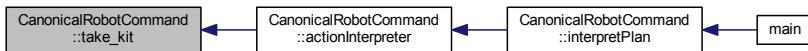
Parameters

<i>paramList</i>	List of parameters for the action take-kit
------------------	---

Todo This function will be written once the **Canonical Robot Command** for the action **take-kit** is implemented in ROS/USARSim

Definition at line 304 of file CanonicalRobotCommand.cc.

Here is the caller graph for this function:



8.5.3.21 void CanonicalRobotCommand::take_kit_tray (vector< string > paramList)

A canonical robot command for the PDDL action **take-kit-tray**

Parameters

<i>paramList</i>	List of parameters for the action take-kit-tray
<i>kittingplan</i>	Instance of KittingPlan

Todo This function will be written once the **Canonical Robot Command** for the action **take-kit-tray** is implemented in ROS/USARSim

Definition at line 315 of file CanonicalRobotCommand.cc.

Here is the caller graph for this function:



8.5.3.22 void CanonicalRobotCommand::take_part (vector< string > paramList, KittingPlan * kittingplan)

A canonical robot command for the PDDL action **take-part**

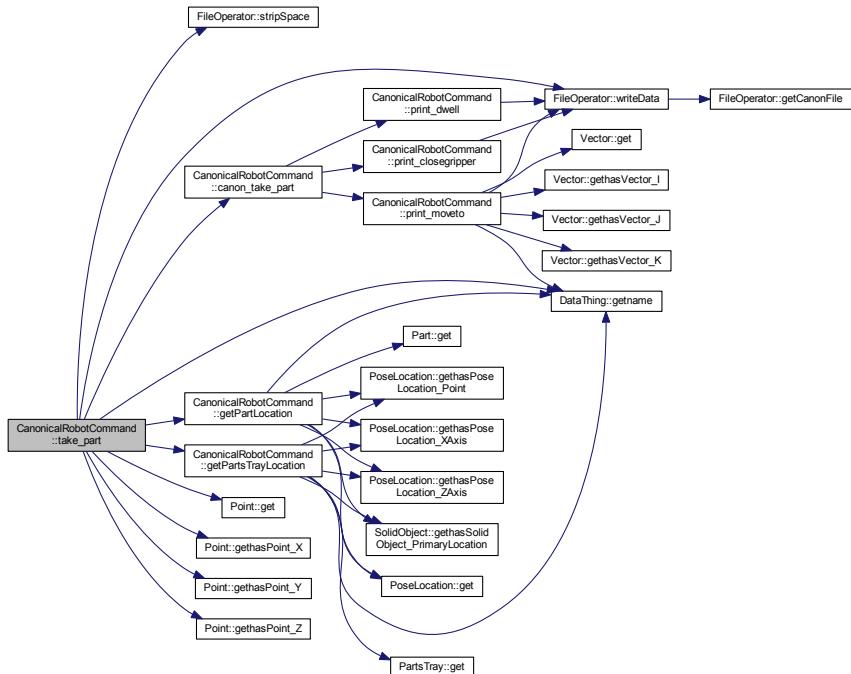
Parameters

<i>paramList</i>	List of parameters for the action take-part
<i>kittingplan</i>	Instance of KittingPlan

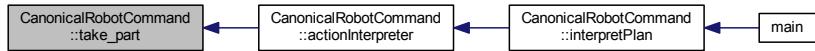
Todo This function will be written once the **Canonical Robot Command** for the action **take-part** is implemented in ROS/USARSim

Definition at line 326 of file CanonicalRobotCommand.cc.

Here is the call graph for this function:



Here is the caller graph for this function:



8.5.4 Member Data Documentation

8.5.4.1 double CanonicalRobotCommand::m_dwell [private]

Definition at line 88 of file CanonicalRobotCommand.h.

8.5.4.2 string CanonicalRobotCommand::m_kit_tray [private]

Definition at line 89 of file CanonicalRobotCommand.h.

8.5.4.3 double CanonicalRobotCommand::m_safe_z [private]

Definition at line 87 of file CanonicalRobotCommand.h.

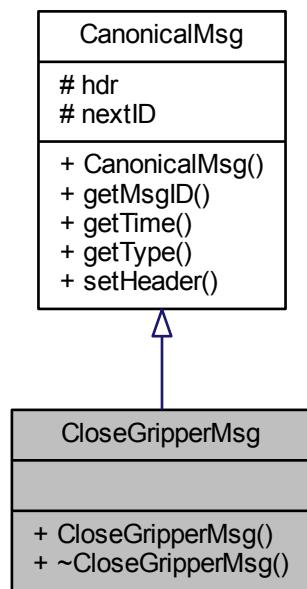
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/[CanonicalRobotCommand.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/[CanonicalRobotCommand.cc](#)

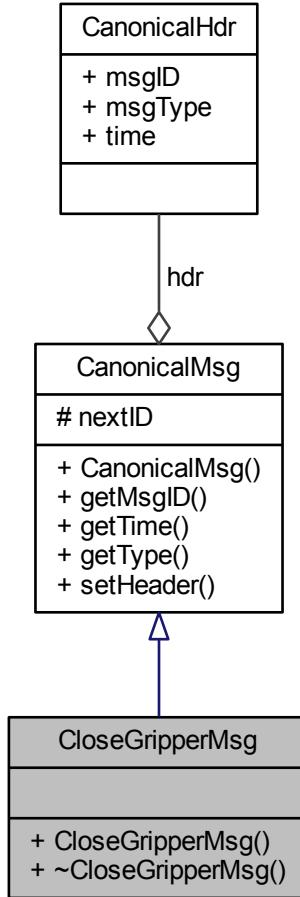
8.6 CloseGripperMsg Class Reference

```
#include <canonicalMsg.hh>
```

Inheritance diagram for CloseGripperMsg:



Collaboration diagram for CloseGripperMsg:



Public Member Functions

- [CloseGripperMsg \(\)](#)
- [~CloseGripperMsg \(\)](#)

Additional Inherited Members

8.6.1 Detailed Description

Definition at line 65 of file canonicalMsg.hh.

8.6.2 Constructor & Destructor Documentation

8.6.2.1 CloseGripperMsg::CloseGripperMsg () [inline]

Definition at line 67 of file canonicalMsg.hh.

8.6.2.2 CloseGripperMsg::~CloseGripperMsg () [inline]

Definition at line 68 of file canonicalMsg.hh.

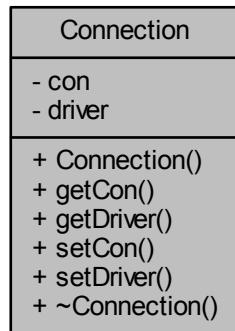
The documentation for this class was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[canonicalMsg.hh](#)

8.7 Connection Class Reference

```
#include <Connection.h>
```

Collaboration diagram for Connection:



Public Member Functions

- [Connection](#) (std::string url, std::string user, std::string pass, std::string name)
- sql::Connection * [getCon](#) () const
- sql::mysql::MySQL_Driver * [getDriver](#) () const
- void [setCon](#) (sql::Connection *con)
- void [setDriver](#) (sql::mysql::MySQL_Driver *driver)
- virtual [~Connection](#) ()

Private Attributes

- sql::Connection * [con](#)
- sql::mysql::MySQL_Driver * [driver](#)

8.7.1 Detailed Description

Definition at line 27 of file Connection.h.

8.7.2 Constructor & Destructor Documentation

8.7.2.1 Connection::Connection (std::string *url*, std::string *user*, std::string *pass*, std::string *name*)

Definition at line 15 of file Connection.cpp.

8.7.2.2 Connection::~Connection () [virtual]

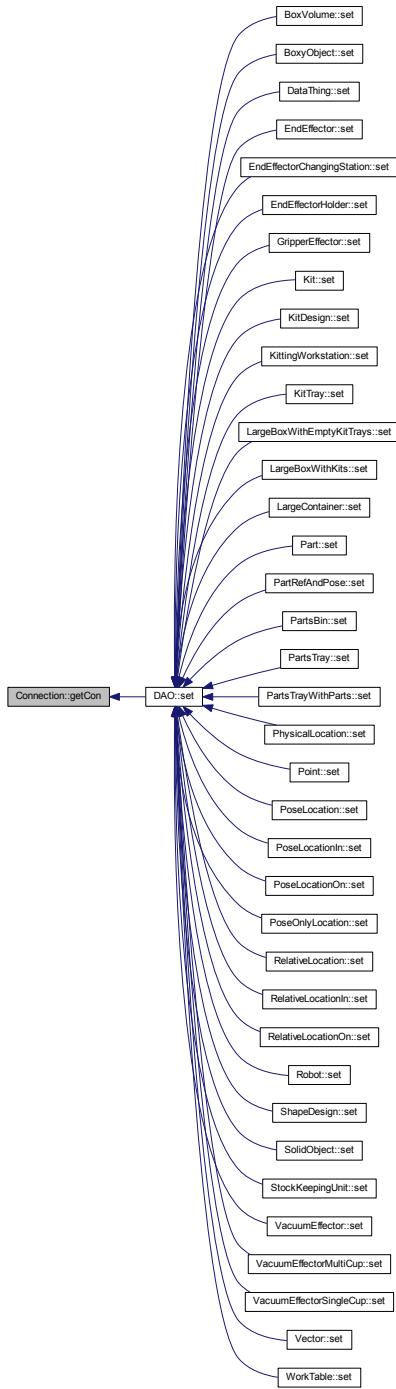
Definition at line 26 of file Connection.cpp.

8.7.3 Member Function Documentation

8.7.3.1 sql::Connection * Connection::getCon () const

Definition at line 29 of file Connection.cpp.

Here is the caller graph for this function:



8.7.3.2 `sql::mysql::MySQL_Driver * Connection::getDriver() const`

Definition at line 32 of file `Connection.cpp`.

8.7.3.3 void Connection::setCon (sql::Connection * *con*)

Definition at line 35 of file Connection.cpp.

8.7.3.4 void Connection::setDriver (sql::mysql::MySQL_Driver * *driver*)

Definition at line 38 of file Connection.cpp.

8.7.4 Member Data Documentation

8.7.4.1 sql::Connection* Connection::con [private]

Definition at line 35 of file Connection.h.

8.7.4.2 sql::mysql::MySQL_Driver* Connection::driver [private]

Definition at line 35 of file Connection.h.

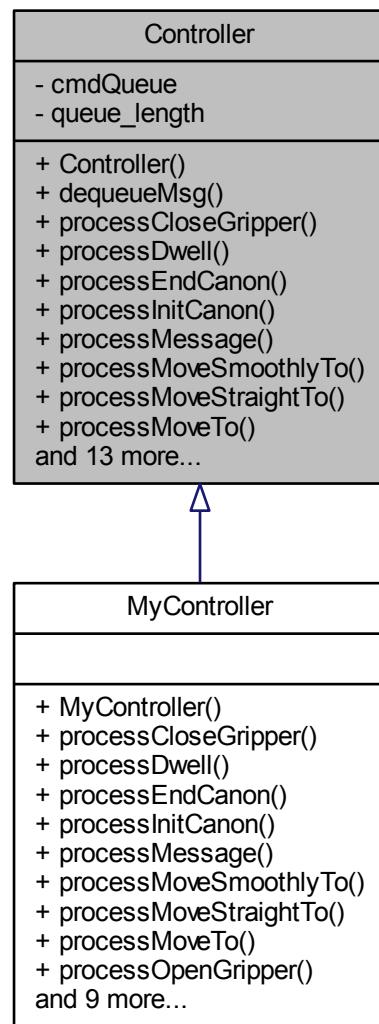
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[Connection.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[Connection.cpp](#)

8.8 Controller Class Reference

```
#include <controller.hh>
```

Inheritance diagram for Controller:



Collaboration diagram for Controller:

Controller
<ul style="list-style-type: none"> - cmdQueue - queue_length
<ul style="list-style-type: none"> + Controller() + dequeueMsg() + processCloseGripper() + processDwell() + processEndCanon() + processInitCanon() + processMessage() + processMoveSmoothlyTo() + processMoveStraightTo() + processMoveTo() and 13 more...

Public Member Functions

- [Controller \(\)](#)
- [int dequeueMsg \(\)](#)
- [virtual int processCloseGripper \(CloseGripperMsg *closeGripperMsg\)=0](#)
- [virtual int processDwell \(DwellMsg *dwellMsg\)=0](#)
- [virtual int processEndCanon \(EndCanonMsg *endCanonMsg\)=0](#)
- [virtual int processInitCanon \(InitCanonMsg *initCanonMsg\)=0](#)
- [virtual int processMessage \(MessageMsg *messageMsg\)=0](#)
- [virtual int processMoveSmoothlyTo \(MoveSmoothlyToMsg *moveSmoothlyToMsg\)=0](#)
- [virtual int processMoveStraightTo \(MoveStraightToMsg *moveStraightToMsg\)=0](#)
- [virtual int processMoveTo \(MoveToMsg *moveToMsg\)=0](#)
- [virtual int processOpenGripper \(OpenGripperMsg *openGripperMsg\)=0](#)
- [virtual int processSetAbsoluteAcceleration \(SetAbsoluteAccelerationMsg *setAbsoluteAccelerationMsg\)=0](#)
- [virtual int processSetAngleUnits \(SetAngleUnitsMsg *setAngleUnitsMsg\)=0](#)
- [virtual int processSetEndAngleTolerance \(SetEndAngleToleranceMsg *setEndAngleToleranceMsg\)=0](#)
- [virtual int processSetEndPointTolerance \(SetEndPointToleranceMsg *setEndPointToleranceMsg\)=0](#)
- [virtual int processSetIntermediatePointTolerance \(SetIntermediatePointToleranceMsg *setIntermediatePointToleranceMsg\)=0](#)
- [virtual int processSetLengthUnits \(SetLengthUnitsMsg *setLengthUnitsMsg\)=0](#)
- [virtual int processSetRelativeAcceleration \(SetRelativeAccelerationMsg *setRelativeAccelerationMsg\)=0](#)
- [virtual int processSetRelativeSpeed \(SetRelativeSpeedMsg *setRelativeSpeedMsg\)=0](#)
- [int queueCloseGripper \(CloseGripperMsg closeGripperMsg\)](#)
- [int queueDwell \(DwellMsg dwellMsg\)](#)
- [int queueMsg \(CanonicalMsg *msgIn\)](#)
- [~Controller \(\)](#)

Private Attributes

- std::deque< void * > [cmdQueue](#)
- int [queue_length](#)

8.8.1 Detailed Description

Definition at line 25 of file controller.hh.

8.8.2 Constructor & Destructor Documentation

8.8.2.1 Controller::Controller()

Definition at line 26 of file controller.cpp.

8.8.2.2 Controller::~Controller()

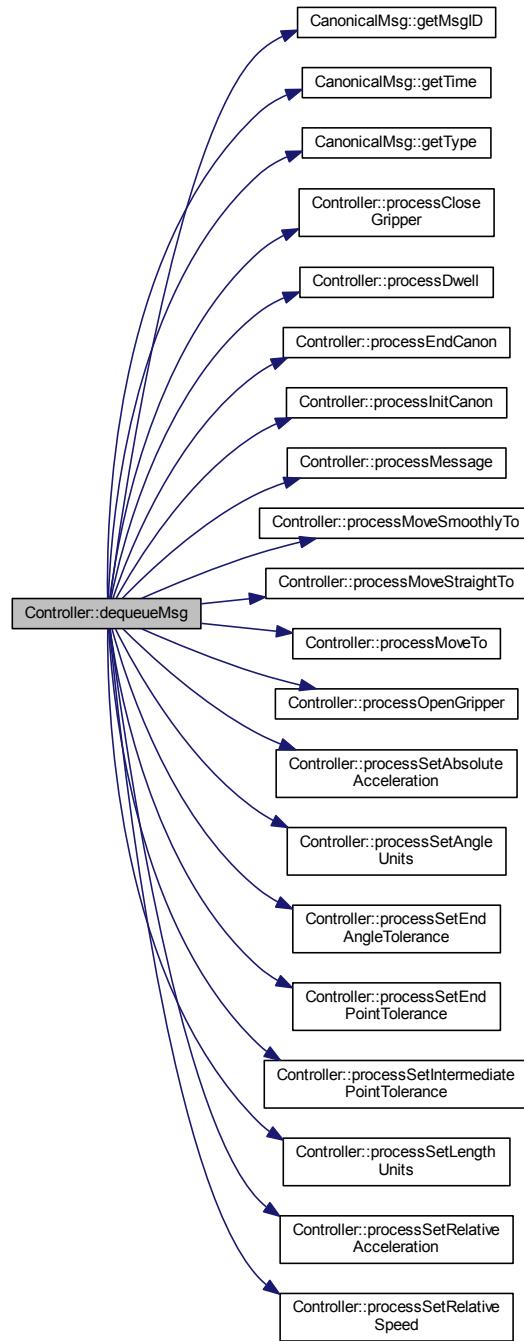
Definition at line 31 of file controller.cpp.

8.8.3 Member Function Documentation

8.8.3.1 int Controller::dequeueMsg()

Definition at line 166 of file controller.cpp.

Here is the call graph for this function:



8.8.3.2 virtual int Controller::processCloseGripper (`CloseGripperMsg * closeGripperMsg`) [pure virtual]

Implemented in [MyController](#).

Here is the caller graph for this function:



8.8.3.3 virtual int Controller::processDwell (*DwellMsg * dwellMsg*) [pure virtual]

Implemented in [MyController](#).

Here is the caller graph for this function:



8.8.3.4 virtual int Controller::processEndCanon (*EndCanonMsg * endCanonMsg*) [pure virtual]

Implemented in [MyController](#).

Here is the caller graph for this function:



8.8.3.5 virtual int Controller::processInitCanon (*InitCanonMsg * initCanonMsg*) [pure virtual]

Implemented in [MyController](#).

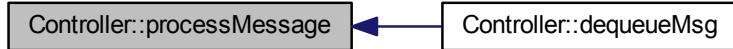
Here is the caller graph for this function:



8.8.3.6 virtual int Controller::processMessage (MessageMsg * *messageMsg*) [pure virtual]

Implemented in [MyController](#).

Here is the caller graph for this function:



8.8.3.7 virtual int Controller::processMoveSmoothlyTo (MoveSmoothlyToMsg * *moveSmoothlyToMsg*) [pure virtual]

Implemented in [MyController](#).

Here is the caller graph for this function:



8.8.3.8 virtual int Controller::processMoveStraightTo (MoveStraightToMsg * *moveStraightToMsg*) [pure virtual]

Implemented in [MyController](#).

Here is the caller graph for this function:



8.8.3.9 virtual int Controller::processMoveTo (*MoveToMsg* * *moveToMsg*) [pure virtual]

Implemented in [MyController](#).

Here is the caller graph for this function:



8.8.3.10 virtual int Controller::processOpenGripper (*OpenGripperMsg* * *openGripperMsg*) [pure virtual]

Implemented in [MyController](#).

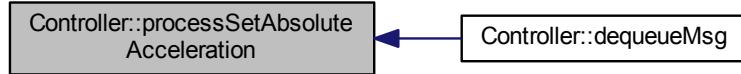
Here is the caller graph for this function:



8.8.3.11 virtual int Controller::processSetAbsoluteAcceleration (*SetAbsoluteAccelerationMsg* * *setAbsoluteAccelerationMsg*) [pure virtual]

Implemented in [MyController](#).

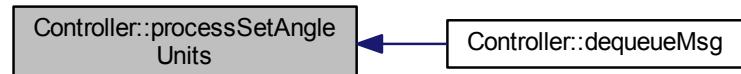
Here is the caller graph for this function:



8.8.3.12 virtual int Controller::processSetAngleUnits (SetAngleUnitsMsg * setAngleUnitsMsg) [pure virtual]

Implemented in [MyController](#).

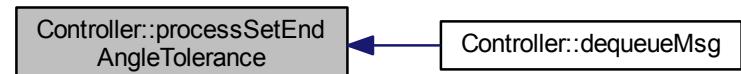
Here is the caller graph for this function:



8.8.3.13 virtual int Controller::processSetEndAngleTolerance (SetEndAngleToleranceMsg * setEndAngleToleranceMsg) [pure virtual]

Implemented in [MyController](#).

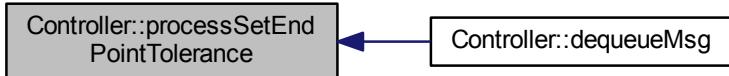
Here is the caller graph for this function:



8.8.3.14 virtual int Controller::processSetEndPointTolerance (SetEndPointToleranceMsg * setEndPointToleranceMsg) [pure virtual]

Implemented in [MyController](#).

Here is the caller graph for this function:



8.8.3.15 virtual int Controller::processSetIntermediatePointTolerance (SetIntermediatePointToleranceMsg * *setIntermediatePointToleranceMsg*) [pure virtual]

Implemented in [MyController](#).

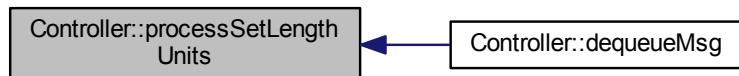
Here is the caller graph for this function:



8.8.3.16 virtual int Controller::processSetLengthUnits (SetLengthUnitsMsg * *setLengthUnitsMsg*) [pure virtual]

Implemented in [MyController](#).

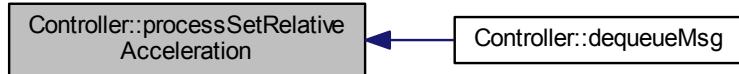
Here is the caller graph for this function:



8.8.3.17 virtual int Controller::processSetRelativeAcceleration (SetRelativeAccelerationMsg * *setRelativeAccelerationMsg*) [pure virtual]

Implemented in [MyController](#).

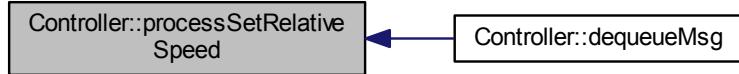
Here is the caller graph for this function:



8.8.3.18 virtual int Controller::processSetRelativeSpeed (SetRelativeSpeedMsg * setRelativeSpeedMsg) [pure virtual]

Implemented in [MyController](#).

Here is the caller graph for this function:



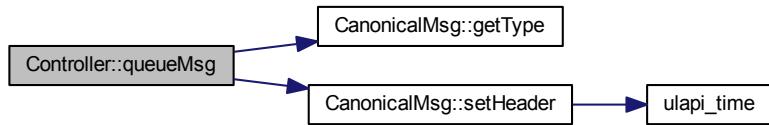
8.8.3.19 int Controller::queueCloseGripper (CloseGripperMsg closeGripperMsg)

8.8.3.20 int Controller::queueDwell (DwellMsg dwellMsg)

8.8.3.21 int Controller::queueMsg (CanonicalMsg * msgIn)

Definition at line 35 of file controller.cpp.

Here is the call graph for this function:



8.8.4 Member Data Documentation

8.8.4.1 `std::deque<void*> Controller::cmdQueue [private]`

Definition at line 52 of file controller.hh.

8.8.4.2 `int Controller::queue_length [private]`

Definition at line 53 of file controller.hh.

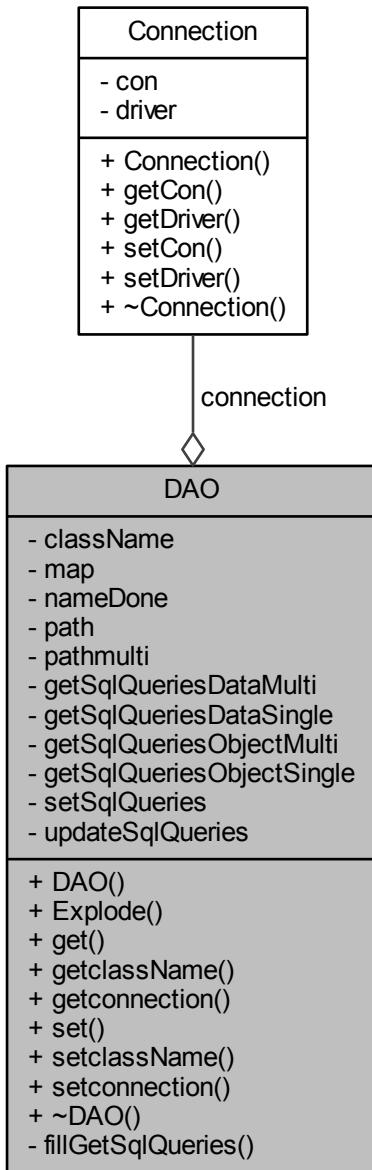
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[controller.hh](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[controller.cpp](#)

8.9 DAO Class Reference

```
#include <DAO.h>
```

Collaboration diagram for DAO:



Public Member Functions

- **DAO** (std::string name)
- std::vector< std::string > **Explode** (const std::string &str, char separator)
- std::map< std::string, std::string > **get** (std::string name)

- std::vector< std::string > [getclassName \(\)](#)
- [Connection * getConnection \(\)](#)
- void [set \(std::map< std::string, std::string > data\)](#)
- void [setclassName \(std::vector< std::string > _className\)](#)
- void [setconnection \(Connection *_connection\)](#)
- [~DAO \(\)](#)

Private Member Functions

- void [fillGetSqlQueries \(\)](#)

Private Attributes

- std::vector< std::string > [className](#)
- [Connection * connection](#)
- std::map< std::string, std::string > [map](#)
- std::vector< std::string > [nameDone](#)
- std::string [path](#)
- std::string [pathmulti](#)

Static Private Attributes

- static std::map< std::string, std::vector< std::string > > [getSqlQueriesDataMulti](#)
- static std::map< std::string, std::string > [getSqlQueriesDataSingle](#)
- static std::map< std::string, std::vector< std::string > > [getSqlQueriesObjectMulti](#)
- static std::map< std::string, std::vector< std::string > > [getSqlQueriesObjectSingle](#)
- static std::map< std::string, std::vector< std::string > > [setSqlQueries](#)
- static std::map< std::string, std::vector< std::string > > [updateSqlQueries](#)

8.9.1 Detailed Description

Definition at line 26 of file DAO.h.

8.9.2 Constructor & Destructor Documentation

8.9.2.1 DAO::DAO (std::string *name*)

Definition at line 16 of file DAO.cpp.

Here is the call graph for this function:



8.9.2.2 DAO::~DAO()

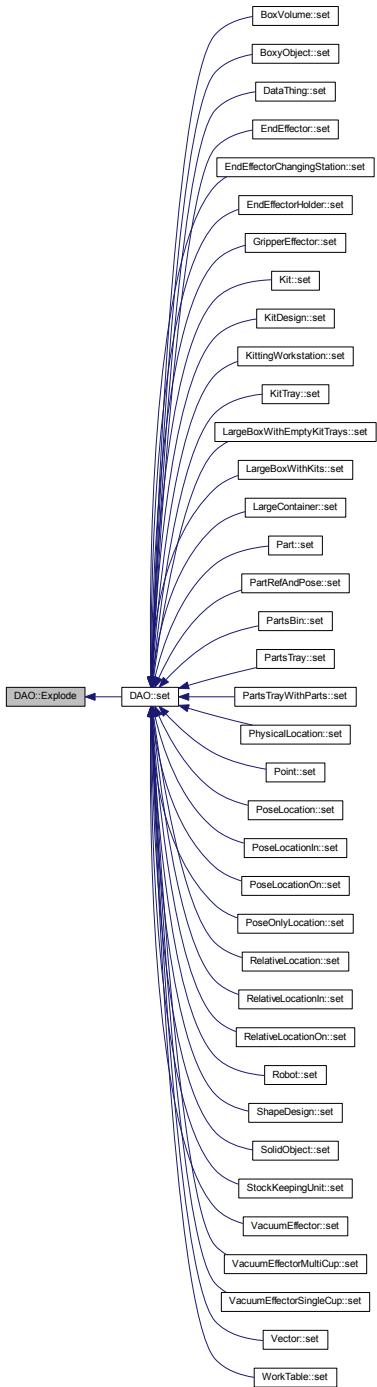
Definition at line 168 of file DAO.cpp.

8.9.3 Member Function Documentation

8.9.3.1 std::vector< std::string > DAO::Explode (const std::string & str, char separator)

Definition at line 914 of file DAO.cpp.

Here is the caller graph for this function:



8.9.3.2 void DAO::fillSqlQueries() [private]

Definition at line 22 of file DAO.cpp.

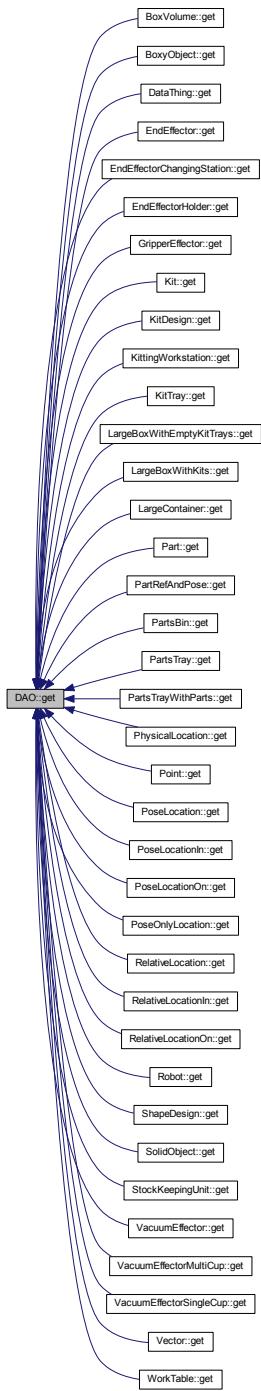
Here is the caller graph for this function:



8.9.3.3 std::map< std::string, std::string > DAO::get (std::string name)

Definition at line 170 of file DAO.cpp.

Here is the caller graph for this function:



8.9.3.4 std::vector< std::string > DAO::getClassName()

Definition at line 156 of file DAO.cpp.

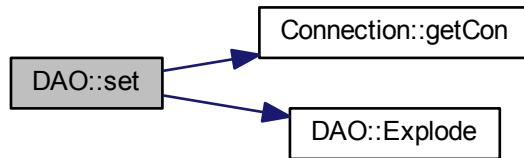
8.9.3.5 Connection * DAO::getconnection()

Definition at line 159 of file DAO.cpp.

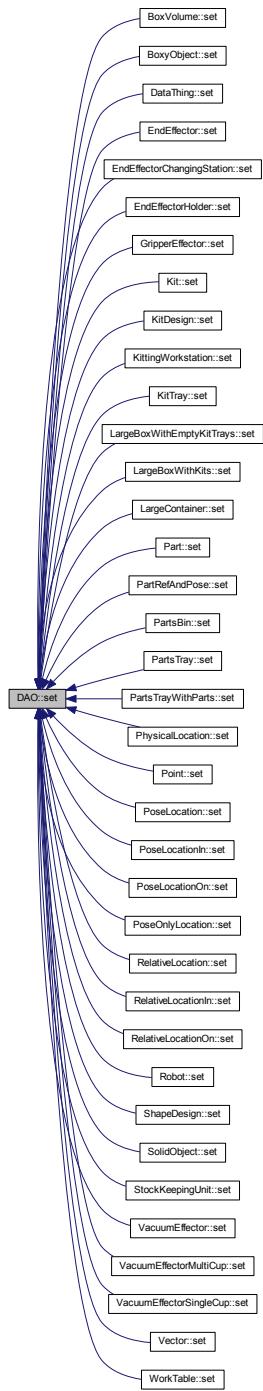
8.9.3.6 void DAO::set(std::map< std::string, std::string > data)

Definition at line 672 of file DAO.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



8.9.3.7 void DAO::setclassName (std::vector< std::string > _className)

Definition at line 162 of file DAO.cpp.

8.9.3.8 void DAO::setconnection (*Connection * _connection*)

Definition at line 165 of file DAO.cpp.

8.9.4 Member Data Documentation

8.9.4.1 std::vector<std::string> DAO::className [private]

Definition at line 28 of file DAO.h.

8.9.4.2 Connection* DAO::connection [private]

Definition at line 29 of file DAO.h.

8.9.4.3 std::map< std::string, std::vector< std::string > > DAO::getSqlQueriesDataMulti [static], [private]

Definition at line 36 of file DAO.h.

8.9.4.4 std::map< std::string, std::string > DAO::getSqlQueriesDataSingle [static], [private]

Definition at line 35 of file DAO.h.

8.9.4.5 std::map< std::string, std::vector< std::string > > DAO::getSqlQueriesObjectMulti [static], [private]

Definition at line 38 of file DAO.h.

8.9.4.6 std::map< std::string, std::vector< std::string > > DAO::getSqlQueriesObjectSingle [static], [private]

Definition at line 37 of file DAO.h.

8.9.4.7 std::map<std::string, std::string> DAO::map [private]

Definition at line 31 of file DAO.h.

8.9.4.8 std::vector<std::string> DAO::nameDone [private]

Definition at line 30 of file DAO.h.

8.9.4.9 std::string DAO::path [private]

Definition at line 32 of file DAO.h.

8.9.4.10 std::string DAO::pathmulti [private]

Definition at line 33 of file DAO.h.

8.9.4.11 std::map<std::string, std::vector<std::string>> DAO::setSqlQueries [static], [private]

Definition at line 39 of file DAO.h.

8.9.4.12 std::map<std::string, std::vector<std::string>> DAO::updateSqlQueries [static], [private]

Definition at line 40 of file DAO.h.

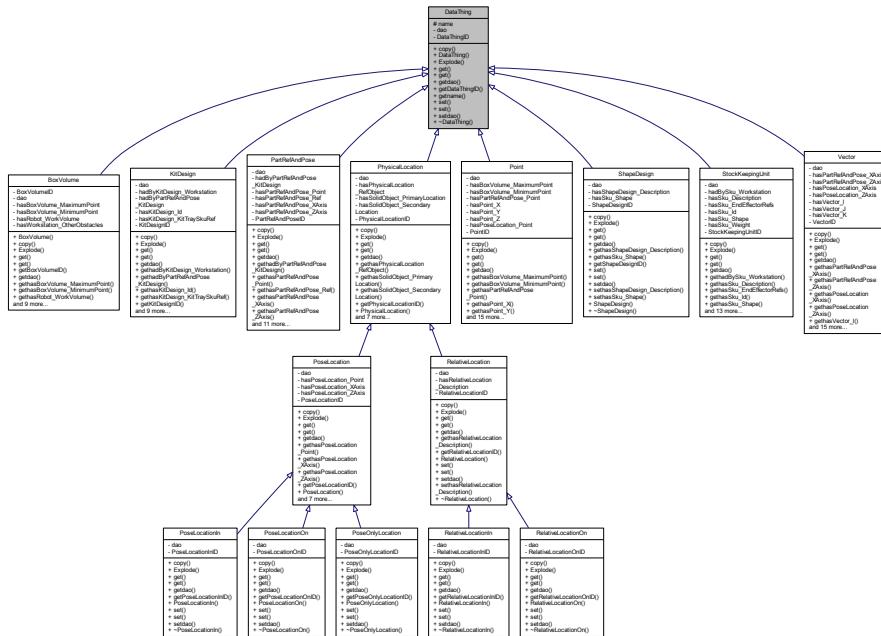
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/DAO.h
 - C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/DAO.cpp

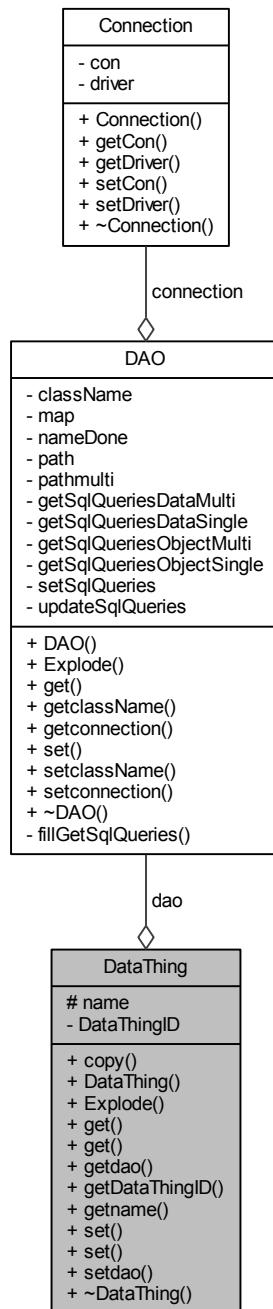
8.10 DataThing Class Reference

```
#include <DataThing.h>
```

Inheritance diagram for DataThing:



Collaboration diagram for DataThing:



Public Member Functions

- void [copy](#) (std::map< std::string, std::string > object)
- [DataThing](#) (std::string name)

- std::vector< std::string > [Explode](#) (const std::string &str, char separator)
- void [get](#) (int id)
- void [get](#) (std::string name)
- DAO * [getdao](#) ()
- int [getDataThingID](#) ()
- std::string [getname](#) ()
- void [set](#) (int id, DataThing *obj)
- void [set](#) (std::string name)
- void [setdao](#) (DAO *_dao)
- ~DataThing ()

Protected Attributes

- std::string [name](#)

Private Attributes

- DAO * [dao](#)
- int [DataThingID](#)

8.10.1 Detailed Description

Definition at line 26 of file DataThing.h.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 DataThing::DataThing (std::string name)

Definition at line 19 of file DataThing.cpp.

8.10.2.2 DataThing::~DataThing ()

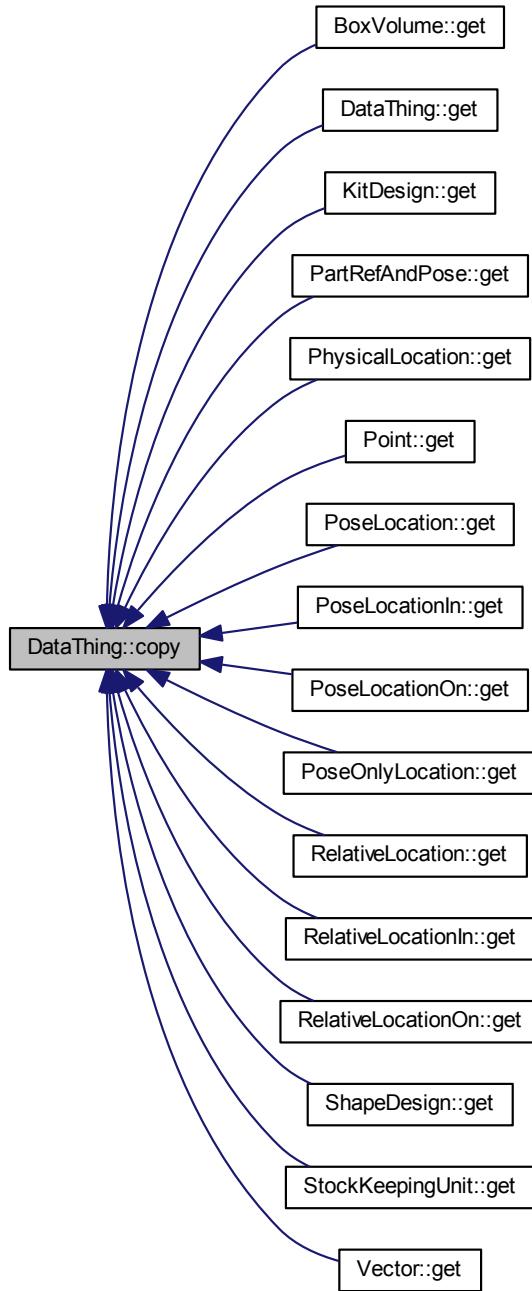
Definition at line 22 of file DataThing.cpp.

8.10.3 Member Function Documentation

8.10.3.1 void DataThing::copy (std::map< std::string, std::string > object)

Definition at line 56 of file DataThing.cpp.

Here is the caller graph for this function:



8.10.3.2 std::vector< std::string > DataThing::Explode (const std::string & str, char separator)

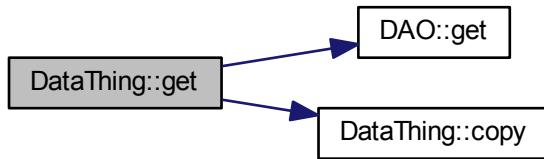
Definition at line 65 of file DataThing.cpp.

8.10.3.3 void DataThing::get(int *id*)

8.10.3.4 void DataThing::get(std::string *name*)

Definition at line 37 of file DataThing.cpp.

Here is the call graph for this function:



8.10.3.5 DAO * DataThing::getdao()

Definition at line 28 of file DataThing.cpp.

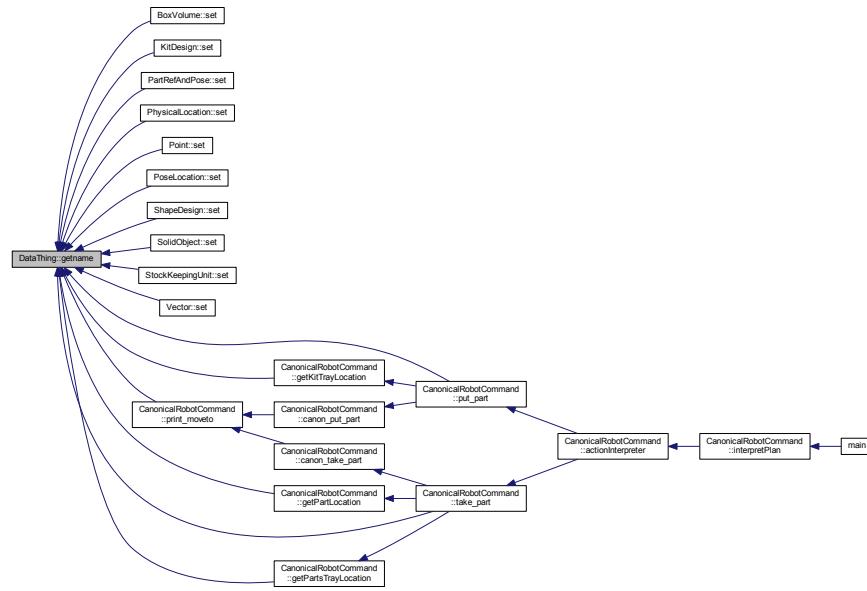
8.10.3.6 int DataThing::getDataThingID()

Definition at line 25 of file DataThing.cpp.

8.10.3.7 std::string DataThing::getname()

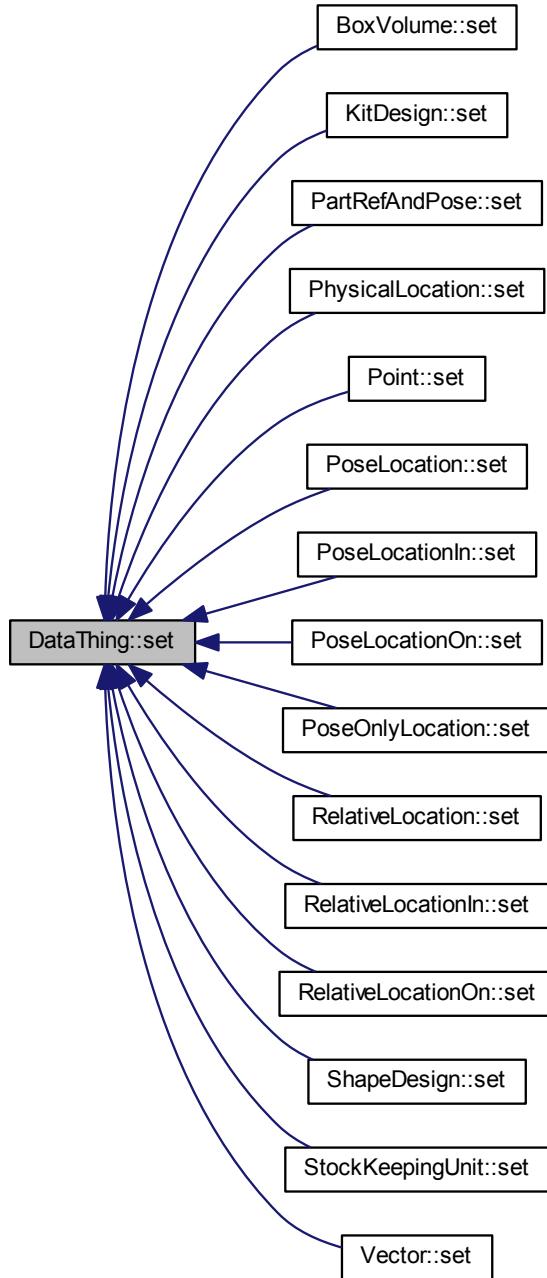
Definition at line 31 of file DataThing.cpp.

Here is the caller graph for this function:



8.10.3.8 void DataThing::set (int *id*, DataThing * *obj*)

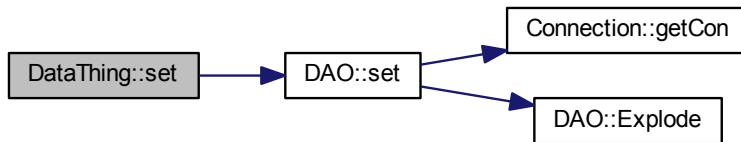
Here is the caller graph for this function:



8.10.3.9 void DataThing::set (std::string name)

Definition at line 44 of file DataThing.cpp.

Here is the call graph for this function:



8.10.3.10 void DataThing::setdao (DAO * _dao)

Definition at line 34 of file DataThing.cpp.

8.10.4 Member Data Documentation

8.10.4.1 DAO* DataThing::dao [private]

Definition at line 28 of file DataThing.h.

8.10.4.2 int DataThing::DataThingID [private]

Definition at line 27 of file DataThing.h.

8.10.4.3 std::string DataThing::name [protected]

Definition at line 30 of file DataThing.h.

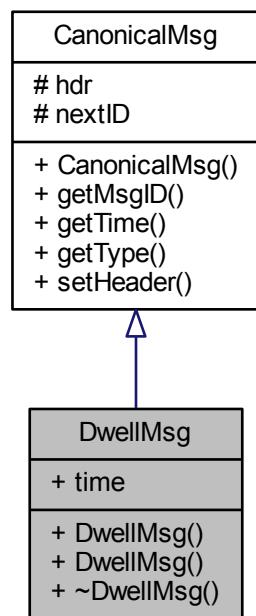
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[DataThing.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[DataThing.cpp](#)

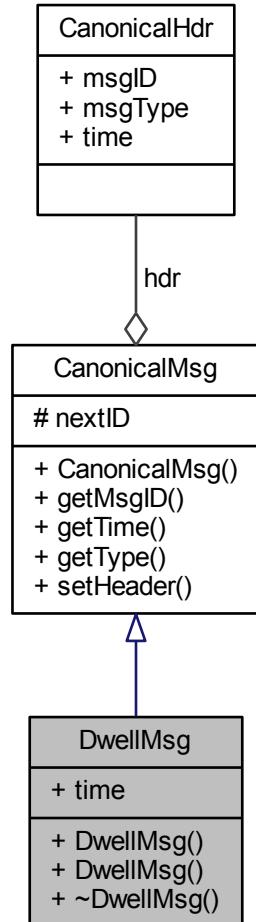
8.11 DwellMsg Class Reference

```
#include <canonicalMsg.hh>
```

Inheritance diagram for DwellMsg:



Collaboration diagram for DwellMsg:



Public Member Functions

- [DwellMsg \(\)](#)
- [DwellMsg \(double timeln\)](#)
- [~DwellMsg \(\)](#)

Public Attributes

- double [time](#)

Additional Inherited Members

8.11.1 Detailed Description

Definition at line 71 of file canonicalMsg.hh.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 DwellMsg::DwellMsg() [inline]

Definition at line 73 of file canonicalMsg.hh.

8.11.2.2 DwellMsg::DwellMsg(double *timeln*) [inline]

Definition at line 74 of file canonicalMsg.hh.

8.11.2.3 DwellMsg::~DwellMsg() [inline]

Definition at line 75 of file canonicalMsg.hh.

8.11.3 Member Data Documentation

8.11.3.1 double DwellMsg::time

Definition at line 75 of file canonicalMsg.hh.

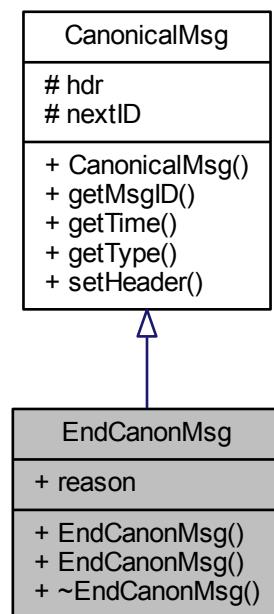
The documentation for this class was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[canonicalMsg.hh](#)

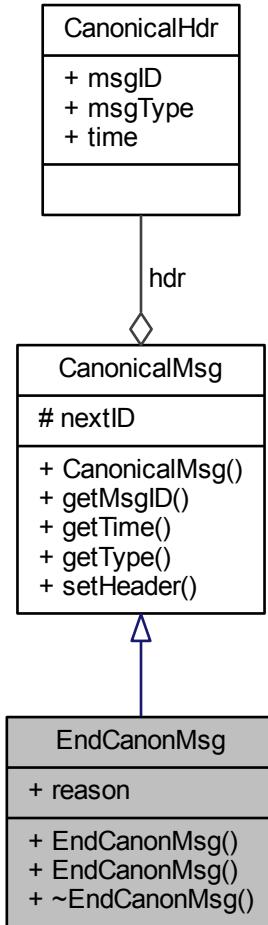
8.12 EndCanonMsg Class Reference

```
#include <canonicalMsg.hh>
```

Inheritance diagram for EndCanonMsg:



Collaboration diagram for EndCanonMsg:



Public Member Functions

- [EndCanonMsg \(\)](#)
- [EndCanonMsg \(int reasonIn\)](#)
- [~EndCanonMsg \(\)](#)

Public Attributes

- int [reason](#)

Additional Inherited Members

8.12.1 Detailed Description

Definition at line 79 of file canonicalMsg.hh.

8.12.2 Constructor & Destructor Documentation

8.12.2.1 EndCanonMsg::EndCanonMsg() [inline]

Definition at line 81 of file canonicalMsg.hh.

8.12.2.2 EndCanonMsg::EndCanonMsg(int *reasonIn*) [inline]

Definition at line 82 of file canonicalMsg.hh.

8.12.2.3 EndCanonMsg::~EndCanonMsg() [inline]

Definition at line 83 of file canonicalMsg.hh.

8.12.3 Member Data Documentation

8.12.3.1 int EndCanonMsg::reason

Definition at line 83 of file canonicalMsg.hh.

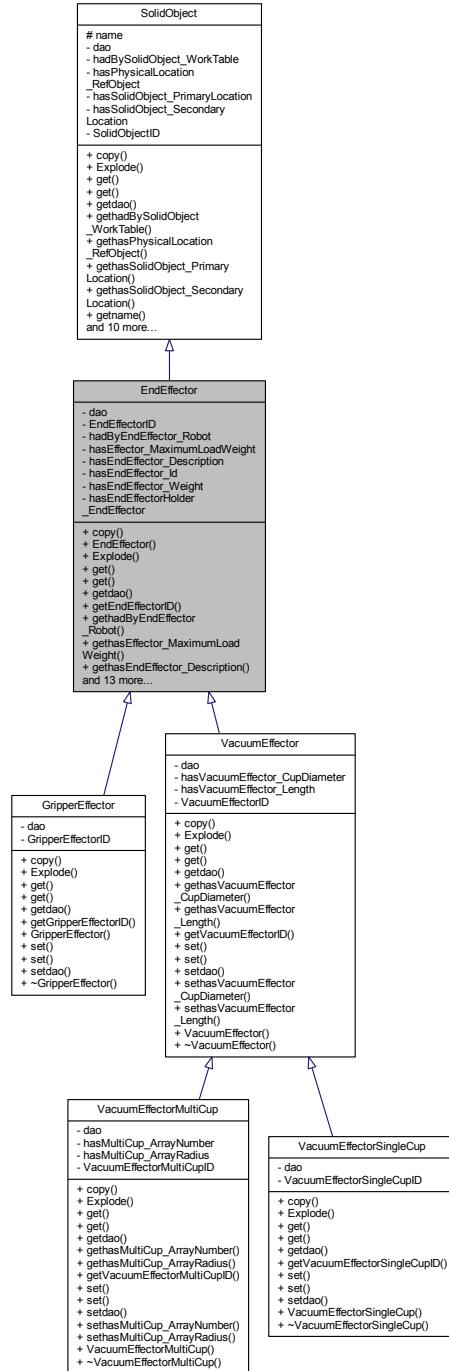
The documentation for this class was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[canonicalMsg.hh](#)

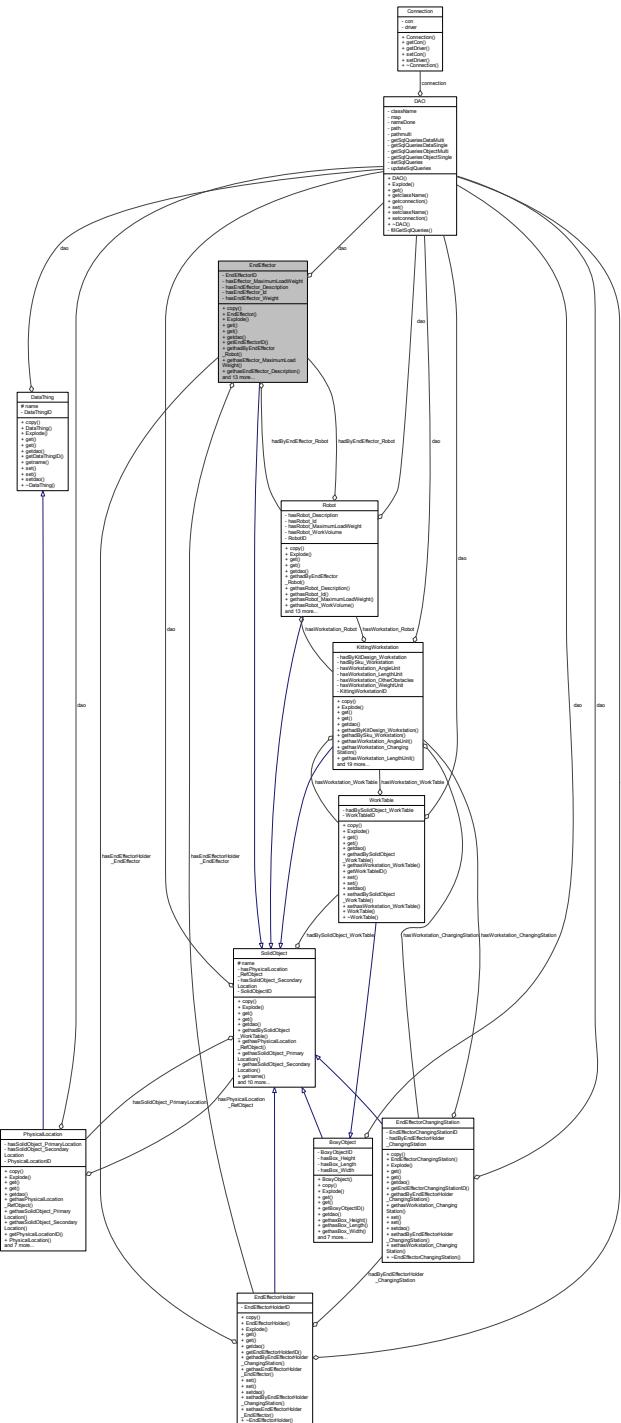
8.13 EndEffector Class Reference

```
#include <EndEffector.h>
```

Inheritance diagram for EndEffector:



Collaboration diagram for EndEffector:



Public Member Functions

- void `copy` (std::map< std::string, std::string > object)
 - `EndEffector` (std::string name)

- std::vector< std::string > [Explode](#) (const std::string &str, char separator)
- void [get](#) (int id)
- void [get](#) (std::string name)
- DAO * [getdao](#) ()
- int [getEndEffectorID](#) ()
- Robot * [gethadByEndEffector_Robot](#) ()
- double [gethasEffector_MaximumLoadWeight](#) ()
- std::string [gethasEndEffector_Description](#) ()
- std::string [gethasEndEffector_Id](#) ()
- double [gethasEndEffector_Weight](#) ()
- EndEffectorHolder * [gethasEndEffectorHolder_EndEffector](#) ()
- void [set](#) (int id, EndEffector *obj)
- void [set](#) (std::string name)
- void [setdao](#) (DAO *_dao)
- void [sethadByEndEffector_Robot](#) (Robot *_hadByEndEffector_Robot)
- void [sethasEffector_MaximumLoadWeight](#) (double _hasEffector_MaximumLoadWeight)
- void [sethasEndEffector_Description](#) (std::string _hasEndEffector_Description)
- void [sethasEndEffector_Id](#) (std::string _hasEndEffector_Id)
- void [sethasEndEffector_Weight](#) (double _hasEndEffector_Weight)
- void [sethasEndEffectorHolder_EndEffector](#) (EndEffectorHolder *_hasEndEffectorHolder_EndEffector)
- ~EndEffector ()

Private Attributes

- DAO * dao
- int EndEffectorID
- Robot * hadByEndEffector_Robot
- double hasEffector_MaximumLoadWeight
- std::string hasEndEffector_Description
- std::string hasEndEffector_Id
- double hasEndEffector_Weight
- EndEffectorHolder * hasEndEffectorHolder_EndEffector

Additional Inherited Members

8.13.1 Detailed Description

Definition at line 29 of file EndEffector.h.

8.13.2 Constructor & Destructor Documentation

8.13.2.1 EndEffector::EndEffector (std::string name)

Definition at line 21 of file EndEffector.cpp.

8.13.2.2 EndEffector::~EndEffector ()

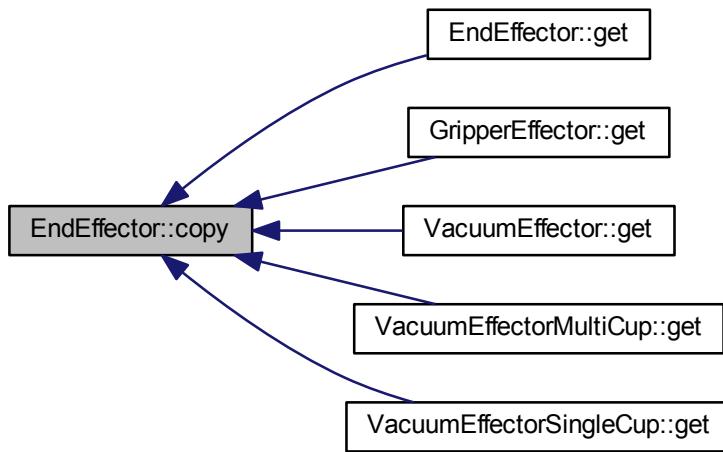
Definition at line 26 of file EndEffector.cpp.

8.13.3 Member Function Documentation

8.13.3.1 void EndEffector::copy (std::map< std::string, std::string > object)

Definition at line 112 of file EndEffector.cpp.

Here is the caller graph for this function:



8.13.3.2 std::vector< std::string > EndEffector::Explode (const std::string & str, char separator)

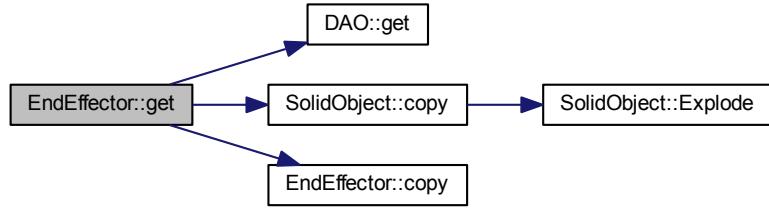
Definition at line 131 of file EndEffector.cpp.

8.13.3.3 void EndEffector::get (int id)

8.13.3.4 void EndEffector::get (std::string name)

Definition at line 76 of file EndEffector.cpp.

Here is the call graph for this function:



8.13.3.5 `DAO * EndEffector::getdao()`

Definition at line 46 of file `EndEffector.cpp`.

8.13.3.6 `int EndEffector::getEndEffectorID()`

Definition at line 43 of file `EndEffector.cpp`.

8.13.3.7 `Robot * EndEffector::gethadByEndEffector_Robot()`

Definition at line 49 of file `EndEffector.cpp`.

8.13.3.8 `double EndEffector::gethasEffecto_MaximumLoadWeight()`

Definition at line 37 of file `EndEffector.cpp`.

8.13.3.9 `std::string EndEffector::gethasEndEffector_Description()`

Definition at line 31 of file `EndEffector.cpp`.

8.13.3.10 `std::string EndEffector::gethasEndEffector_Id()`

Definition at line 40 of file `EndEffector.cpp`.

8.13.3.11 `double EndEffector::gethasEndEffector_Weight()`

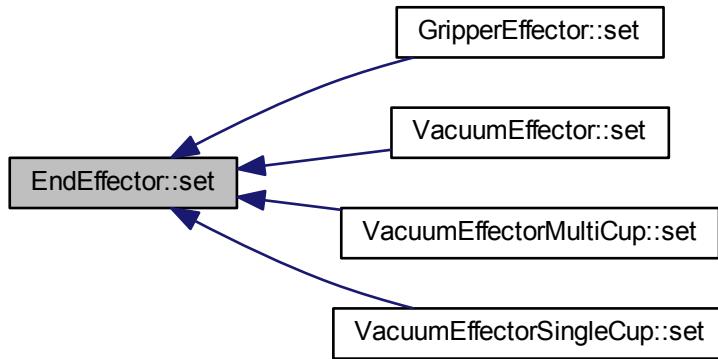
Definition at line 34 of file `EndEffector.cpp`.

8.13.3.12 `EndEffectorHolder * EndEffector::gethasEndEffectorHolder_EndEffector()`

Definition at line 52 of file `EndEffector.cpp`.

8.13.3.13 void EndEffector::set (int *id*, EndEffector * *obj*)

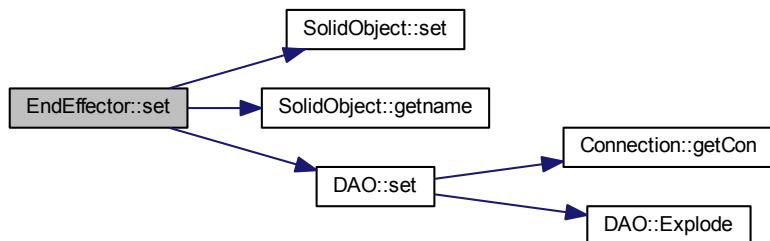
Here is the caller graph for this function:



8.13.3.14 void EndEffector::set (std::string *name*)

Definition at line 86 of file EndEffector.cpp.

Here is the call graph for this function:



8.13.3.15 void EndEffector::setdao (DAO * *_dao*)

Definition at line 67 of file EndEffector.cpp.

8.13.3.16 void EndEffector::sethadByEndEffector_Robot (**Robot** * *_hadByEndEffector_Robot*)

Definition at line 70 of file EndEffector.cpp.

8.13.3.17 void EndEffector::sethasEffecto_MaximumLoadWeight (double *_hasEffecto_MaximumLoadWeight*)

Definition at line 61 of file EndEffector.cpp.

8.13.3.18 void EndEffector::sethasEndEffector_Description (std::string *_hasEndEffector_Description*)

Definition at line 55 of file EndEffector.cpp.

8.13.3.19 void EndEffector::sethasEndEffector_Id (std::string *_hasEndEffector_Id*)

Definition at line 64 of file EndEffector.cpp.

8.13.3.20 void EndEffector::sethasEndEffector_Weight (double *_hasEndEffector_Weight*)

Definition at line 58 of file EndEffector.cpp.

8.13.3.21 void EndEffector::sethasEndEffectorHolder_EndEffector (**EndEffectorHolder** * *_hasEndEffectorHolder_EndEffector*)

Definition at line 73 of file EndEffector.cpp.

8.13.4 Member Data Documentation

8.13.4.1 **DAO*** EndEffector::*dao* [private]

Definition at line 35 of file EndEffector.h.

8.13.4.2 int EndEffector::*EndEffectorID* [private]

Definition at line 34 of file EndEffector.h.

8.13.4.3 **Robot*** EndEffector::*hadByEndEffector_Robot* [private]

Definition at line 36 of file EndEffector.h.

8.13.4.4 double EndEffector::*hasEffecto_MaximumLoadWeight* [private]

Definition at line 32 of file EndEffector.h.

8.13.4.5 std::string EndEffector::*hasEndEffector_Description* [private]

Definition at line 30 of file EndEffector.h.

8.13.4.6 `std::string EndEffector::hasEndEffector_Id` [private]

Definition at line 33 of file EndEffector.h.

8.13.4.7 `double EndEffector::hasEndEffector_Weight` [private]

Definition at line 31 of file EndEffector.h.

8.13.4.8 `EndEffectorHolder* EndEffector::hasEndEffectorHolder_EndEffector` [private]

Definition at line 37 of file EndEffector.h.

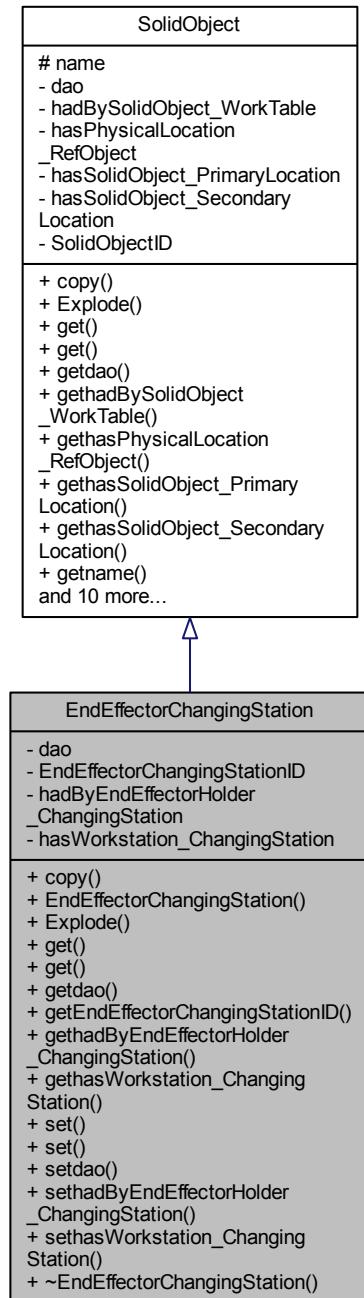
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[EndEffector.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[EndEffector.cpp](#)

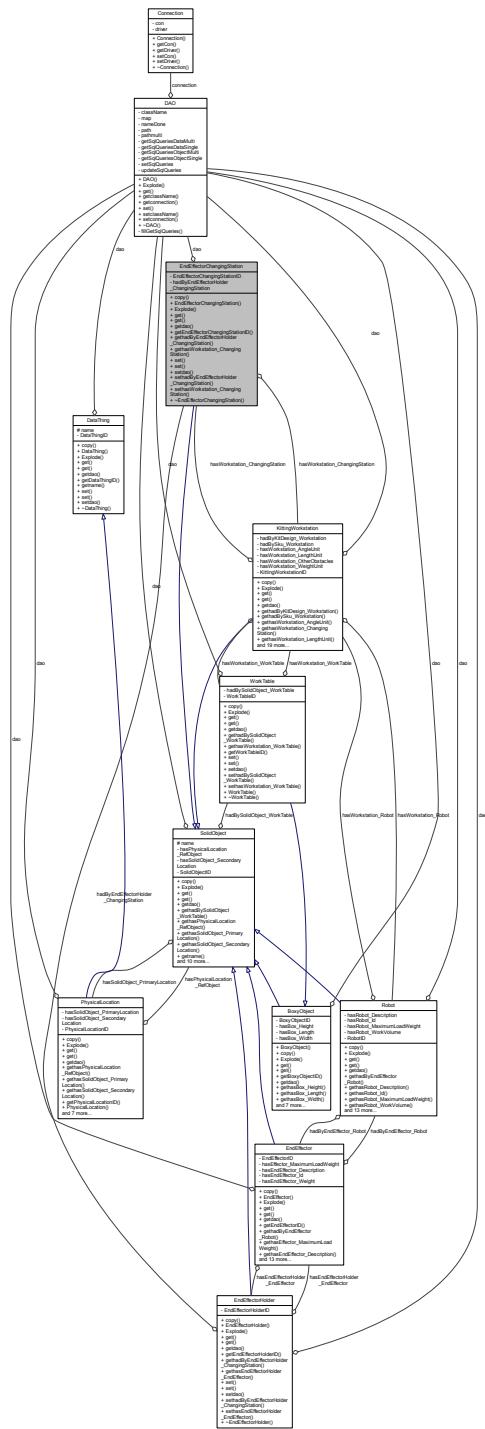
8.14 EndEffectorChangingStation Class Reference

```
#include <EndEffectorChangingStation.h>
```

Inheritance diagram for EndEffectorChangingStation:



Collaboration diagram for EndEffectorChangingStation:



Public Member Functions

- void `copy` (std::map< std::string, std::string > object)
 - `EndEffectorChangingStation` (std::string name)

- std::vector< std::string > [Explode](#) (const std::string &str, char separator)
- void [get](#) (int id)
- void [get](#) (std::string name)
- DAO * [getdao](#) ()
- int [getEndEffectorChangingStationID](#) ()
- std::vector< EndEffectorHolder * > [gethadByEndEffectorHolder_ChangingStation](#) ()
- KittingWorkstation * [gethasWorkstation_ChangingStation](#) ()
- void [set](#) (int id, EndEffectorChangingStation *obj)
- void [set](#) (std::string name)
- void [setdao](#) (DAO *_dao)
- void [sethadByEndEffectorHolder_ChangingStation](#) (std::vector< EndEffectorHolder * > _hadByEndEffectorHolder_ChangingStation)
- void [sethasWorkstation_ChangingStation](#) (KittingWorkstation *_hasWorkstation_ChangingStation)
- ~[EndEffectorChangingStation](#) ()

Private Attributes

- DAO * dao
- int [EndEffectorChangingStationID](#)
- std::vector< EndEffectorHolder * > [hadByEndEffectorHolder_ChangingStation](#)
- KittingWorkstation * [hasWorkstation_ChangingStation](#)

Additional Inherited Members

8.14.1 Detailed Description

Definition at line 29 of file EndEffectorChangingStation.h.

8.14.2 Constructor & Destructor Documentation

8.14.2.1 EndEffectorChangingStation::EndEffectorChangingStation (std::string name)

Definition at line 21 of file EndEffectorChangingStation.cpp.

8.14.2.2 EndEffectorChangingStation::~EndEffectorChangingStation ()

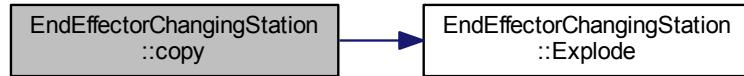
Definition at line 25 of file EndEffectorChangingStation.cpp.

8.14.3 Member Function Documentation

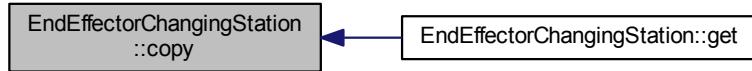
8.14.3.1 void EndEffectorChangingStation::copy (std::map< std::string, std::string > object)

Definition at line 84 of file EndEffectorChangingStation.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



8.14.3.2 std::vector< std::string > EndEffectorChangingStation::Explode (const std::string & str, char separator)

Definition at line 102 of file EndEffectorChangingStation.cpp.

Here is the caller graph for this function:

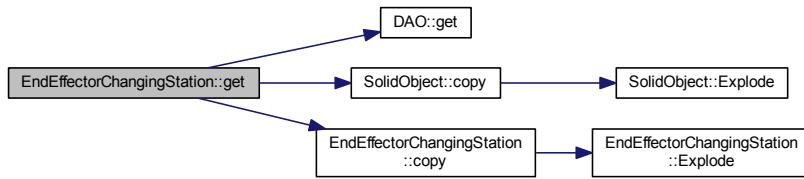


8.14.3.3 void EndEffectorChangingStation::get (int id)

8.14.3.4 void EndEffectorChangingStation::get (std::string name)

Definition at line 52 of file EndEffectorChangingStation.cpp.

Here is the call graph for this function:



8.14.3.5 `DAO * EndEffectorChangingStation::getdao ()`

Definition at line 34 of file `EndEffectorChangingStation.cpp`.

8.14.3.6 `int EndEffectorChangingStation::getEndEffectorChangingStationID ()`

Definition at line 31 of file `EndEffectorChangingStation.cpp`.

8.14.3.7 `std::vector< EndEffectorHolder * > EndEffectorChangingStation::gethadByEndEffectorHolder_ChangingStation ()`

Definition at line 37 of file `EndEffectorChangingStation.cpp`.

8.14.3.8 `KittingWorkstation * EndEffectorChangingStation::gethasWorkstation_ChangingStation ()`

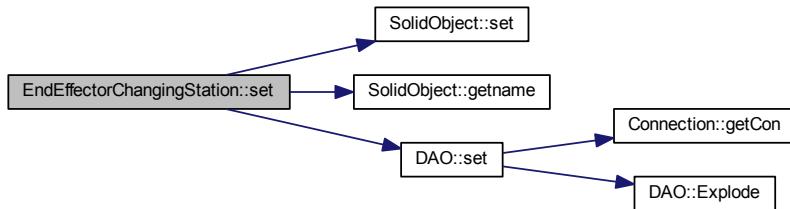
Definition at line 40 of file `EndEffectorChangingStation.cpp`.

8.14.3.9 `void EndEffectorChangingStation::set (int id, EndEffectorChangingStation * obj)`

8.14.3.10 `void EndEffectorChangingStation::set (std::string name)`

Definition at line 62 of file `EndEffectorChangingStation.cpp`.

Here is the call graph for this function:



8.14.3.11 void EndEffectorChangingStation::setdao (DAO * *_dao*)

Definition at line 43 of file EndEffectorChangingStation.cpp.

8.14.3.12 void EndEffectorChangingStation::sethadByEndEffectorHolder_ChangingStation (std::vector< EndEffectorHolder * > *_hadByEndEffectorHolder_ChangingStation*)

Definition at line 46 of file EndEffectorChangingStation.cpp.

8.14.3.13 void EndEffectorChangingStation::sethasWorkstation_ChangingStation (KittingWorkstation * *_hasWorkstation_ChangingStation*)

Definition at line 49 of file EndEffectorChangingStation.cpp.

8.14.4 Member Data Documentation

8.14.4.1 DAO* EndEffectorChangingStation::dao [private]

Definition at line 31 of file EndEffectorChangingStation.h.

8.14.4.2 int EndEffectorChangingStation::EndEffectorChangingStationID [private]

Definition at line 30 of file EndEffectorChangingStation.h.

8.14.4.3 std::vector<EndEffectorHolder*> EndEffectorChangingStation::hadByEndEffectorHolder_ChangingStation [private]

Definition at line 32 of file EndEffectorChangingStation.h.

8.14.4.4 KittingWorkstation* EndEffectorChangingStation::hasWorkstation_ChangingStation [private]

Definition at line 33 of file EndEffectorChangingStation.h.

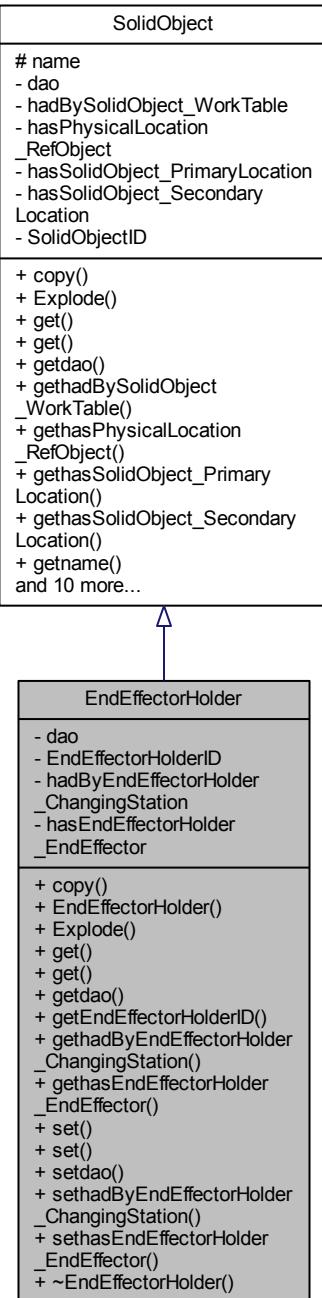
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[EndEffectorChangingStation.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[EndEffectorChangingStation.cpp](#)

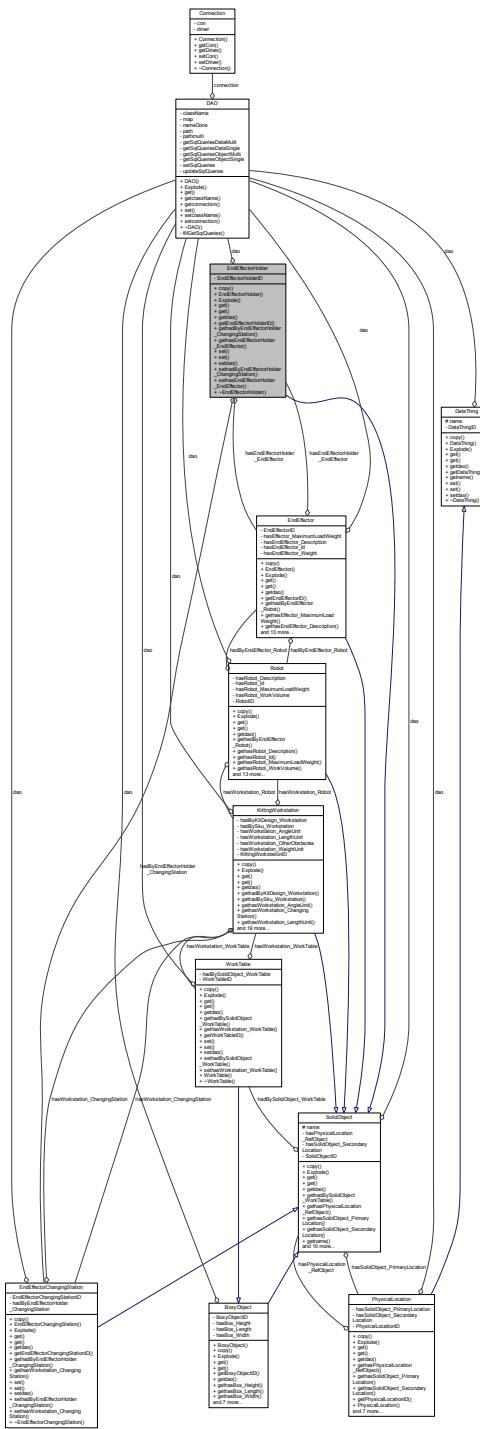
8.15 EndEffectorHolder Class Reference

```
#include <EndEffectorHolder.h>
```

Inheritance diagram for EndEffectorHolder:



Collaboration diagram for EndEffectorHolder:



Public Member Functions

- void **copy** (std::map< std::string, std::string > object)
 - **EndEffectorHolder** (std::string name)

- std::vector< std::string > [Explode](#) (const std::string &str, char separator)
- void [get](#) (int id)
- void [get](#) (std::string name)
- DAO * [getdao](#) ()
- int [getEndEffectorHolderID](#) ()
- EndEffectorChangingStation * [gethadByEndEffectorHolder_ChangingStation](#) ()
- EndEffector * [gethasEndEffectorHolder_EndEffector](#) ()
- void [set](#) (int id, EndEffectorHolder *obj)
- void [set](#) (std::string name)
- void [setdao](#) (DAO *_dao)
- void [sethadByEndEffectorHolder_ChangingStation](#) (EndEffectorChangingStation *_hadByEndEffectorHolder_ChangingStation)
- void [sethasEndEffectorHolder_EndEffector](#) (EndEffector *_hasEndEffectorHolder_EndEffector)
- ~[EndEffectorHolder](#) ()

Private Attributes

- DAO * dao
- int [EndEffectorHolderID](#)
- EndEffectorChangingStation * [hadByEndEffectorHolder_ChangingStation](#)
- EndEffector * [hasEndEffectorHolder_EndEffector](#)

Additional Inherited Members

8.15.1 Detailed Description

Definition at line 29 of file EndEffectorHolder.h.

8.15.2 Constructor & Destructor Documentation

8.15.2.1 EndEffectorHolder::EndEffectorHolder (std::string name)

Definition at line 21 of file EndEffectorHolder.cpp.

8.15.2.2 EndEffectorHolder::~EndEffectorHolder ()

Definition at line 26 of file EndEffectorHolder.cpp.

8.15.3 Member Function Documentation

8.15.3.1 void EndEffectorHolder::copy (std::map< std::string, std::string > object)

Definition at line 80 of file EndEffectorHolder.cpp.

Here is the caller graph for this function:



8.15.3.2 std::vector< std::string > EndEffectorHolder::Explode (const std::string & str, char separator)

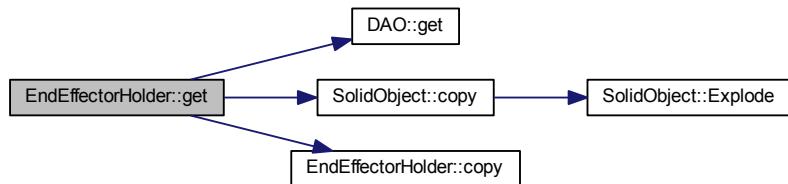
Definition at line 95 of file EndEffectorHolder.cpp.

8.15.3.3 void EndEffectorHolder::get (int id)

8.15.3.4 void EndEffectorHolder::get (std::string name)

Definition at line 52 of file EndEffectorHolder.cpp.

Here is the call graph for this function:



8.15.3.5 DAO * EndEffectorHolder::getdao ()

Definition at line 34 of file EndEffectorHolder.cpp.

8.15.3.6 int EndEffectorHolder::getEndEffectorHolderID ()

Definition at line 31 of file EndEffectorHolder.cpp.

8.15.3.7 EndEffectorChangingStation * EndEffectorHolder::gethadByEndEffectorHolder_ChangingStation ()

Definition at line 37 of file EndEffectorHolder.cpp.

8.15.3.8 `EndEffector * EndEffectorHolder::gethasEndEffectorHolder_EndEffector()`

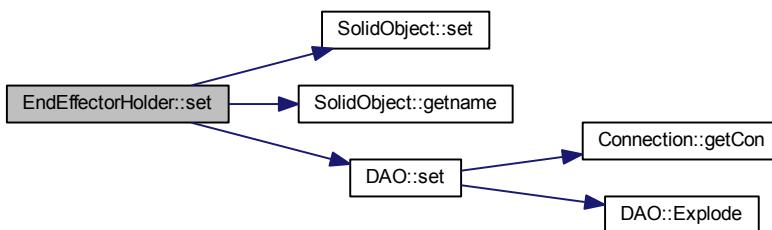
Definition at line 40 of file EndEffectorHolder.cpp.

8.15.3.9 `void EndEffectorHolder::set(int id, EndEffectorHolder * obj)`

8.15.3.10 `void EndEffectorHolder::set(std::string name)`

Definition at line 62 of file EndEffectorHolder.cpp.

Here is the call graph for this function:



8.15.3.11 `void EndEffectorHolder::setdao(DAO * _dao)`

Definition at line 43 of file EndEffectorHolder.cpp.

8.15.3.12 `void EndEffectorHolder::sethadByEndEffectorHolder_ChangingStation(EndEffectorChangingStation * _hadByEndEffectorHolder_ChangingStation)`

Definition at line 46 of file EndEffectorHolder.cpp.

8.15.3.13 `void EndEffectorHolder::sethasEndEffectorHolder_EndEffector(EndEffector * _hasEndEffectorHolder_EndEffector)`

Definition at line 49 of file EndEffectorHolder.cpp.

8.15.4 Member Data Documentation

8.15.4.1 `DAO* EndEffectorHolder::dao [private]`

Definition at line 31 of file EndEffectorHolder.h.

8.15.4.2 `int EndEffectorHolder::EndEffectorHolderID [private]`

Definition at line 30 of file EndEffectorHolder.h.

8.15.4.3 EndEffectorChangingStation* EndEffectorHolder::hadByEndEffectorHolder_ChangingStation [private]

Definition at line 32 of file EndEffectorHolder.h.

8.15.4.4 EndEffector* EndEffectorHolder::hasEndEffectorHolder_EndEffector [private]

Definition at line 33 of file EndEffectorHolder.h.

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

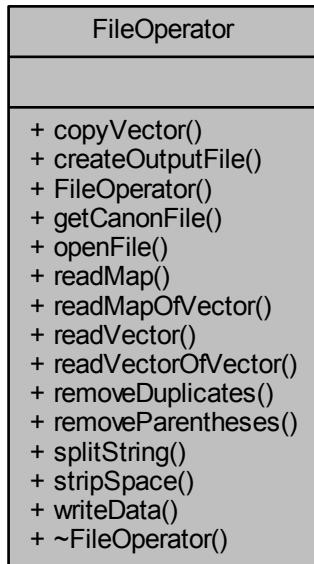
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[EndEffectorHolder.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[EndEffectorHolder.cpp](#)

8.16 FileOperator Class Reference

Class for operations on files.

```
#include <Operator.h>
```

Collaboration diagram for FileOperator:



Public Member Functions

- `vector< string > copyVector (vector< string > myVector)`
- `void createOutputFile ()`

Create the output file that will contain the canonical robot commands. The different steps are defined as follows:

- [FileOperator \(\)](#)
Auto-generated constructor stub.
- [char * getCanonFile \(\)](#)
- [ifstream openFile \(const char *filename\)](#)
- [void readMap \(map< string, string > myMap\)](#)
Read the elements of a map.
- [void readMapOfVector \(map< string, vector< string > > myMap\)](#)
Read the elements of a map that contains a vector as the second parameter.
- [void readVector \(vector< string > myVector\)](#)
Read and display elements of a vector.
- [void readVectorOfVector \(vector< vector< string > > myVector\)](#)
Read the elements of a vector that contains a vector of string.
- [void removeDuplicates \(vector< string > &myVector\)](#)
Remove duplicate values in a vector.
- [string removeParentheses \(std::string s\)](#)
Remove parentheses in a string.
- [vector< string > splitString \(string s\)](#)
Split string s and store each element in a vector<string>
- [void stripSpace \(string &str\)](#)
- [void writeData \(string message\)](#)
Find the type of the parameter myParameter by parsing [KittingPDDLProblem:m_ParamTypeList](#).
- [virtual ~FileOperator \(\)](#)
Auto-generated destructor stub.

8.16.1 Detailed Description

Class for operations on files.

This class is used to manipulate files (open, substract, concat, etc).

Author

[Zeid Kootbally](#) zeid.kootbally@nist.gov

Version

1.0

Date

May 17, 2012

Definition at line 29 of file Operator.h.

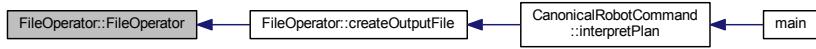
8.16.2 Constructor & Destructor Documentation

8.16.2.1 FileOperator::FileOperator()

Auto-generated constructor stub.

Definition at line 18 of file Operator.cc.

Here is the caller graph for this function:



8.16.2.2 FileOperator::~FileOperator() [virtual]

Auto-generated destructor stub.

Definition at line 24 of file Operator.cc.

8.16.3 Member Function Documentation

8.16.3.1 vector<string> FileOperator::copyVector(vector< string > myVector)

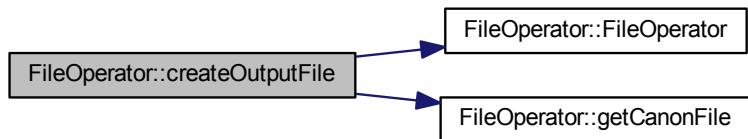
8.16.3.2 void FileOperator::create outputFile()

Create the output file that will contain the canonical robot commands. The different steps are defined as follows:

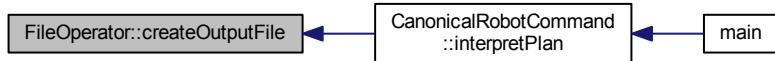
- Retrieve the name of the output file using [FileOperator::getCanonFile\(\)](#)
- Check if this file already exists
 - If it exists then delete it and recreate it
- Create the output file

Definition at line 209 of file Operator.cc.

Here is the call graph for this function:



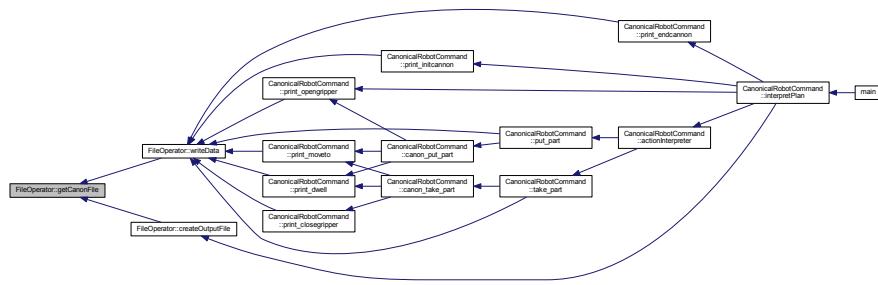
Here is the caller graph for this function:



8.16.3.3 char * FileOperator::getCanonFile ()

Definition at line 37 of file Operator.cc.

Here is the caller graph for this function:



8.16.3.4 ifstream FileOperator::openFile (const char * *filename*)

8.16.3.5 void FileOperator::readMap (map< string, string > myMap)

Read the elements of a map.

Parameters

myMap | The map to read

Definition at line 149 of file Operator.cc.

8.16.3.6 void FileOperator::readMapOfVector (map< string, vector< string > > myMap)

Read the elements of a map that contains a vector as the second parameter.

Parameters

myMap | The map to read

Definition at line 136 of file Operator.cc.

8.16.3.7 void FileOperator::readVector (`vector< string > myVector`)

Read and display elements of a vector.

Parameters

<i>myVector</i>	The vector to read
-----------------	--------------------

Definition at line 100 of file Operator.cc.

8.16.3.8 void FileOperator::readVectorOfVector (`vector< vector< string > > myVector`)

Read the elements of a vector that contains a vector of string.

Parameters

<i>myVector</i>	The vector to read
-----------------	--------------------

Definition at line 113 of file Operator.cc.

8.16.3.9 void FileOperator::removeDuplicates (`vector< string > & myVector`)

Remove duplicate values in a vector.

Parameters

<i>myVector</i>	The vector to read
-----------------	--------------------

Definition at line 126 of file Operator.cc.

8.16.3.10 string FileOperator::removeParentheses (`std::string s`)

Remove parentheses in a string.

Parameters

<i>s</i>	String that contains parentheses
----------	----------------------------------

Returns

The string *s* without parentheses

The different steps are:

- Find the position *pos2* of the closing parenthesis ")"
 - Use *pos2* and remove the closing parenthesis from *s*: use the `string::erase` function
- Find the position *pos2* of the opening parenthesis "("
 - Keep the string that starts after *pos2*: use the `string::substr` function

Definition at line 62 of file Operator.cc.

8.16.3.11 `vector< string > FileOperator::splitString (string s)`

Split string *s* and store each element in a `vector<string>`

Parameters

<code>s</code>	The string that needs to be split
----------------	-----------------------------------

Returns

`vector` A vector that contains all elements from the string *s*

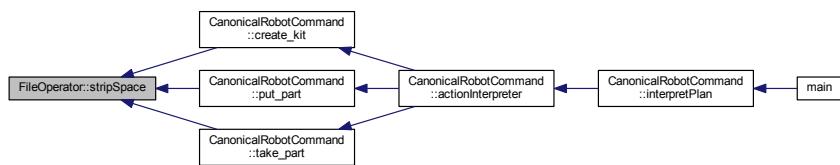
Example: Read the string "word1 word2 word3" and generates a `vector<word1,word2,word3>`

Definition at line 85 of file Operator.cc.

8.16.3.12 `void FileOperator::stripSpace (string & str)`

Definition at line 28 of file Operator.cc.

Here is the caller graph for this function:



8.16.3.13 `void FileOperator::writeData (string message)`

Find the type of the parameter *myParameter* by parsing [KittingPDDLProblem::m_ParamTypeList](#).

Parameters

<code>myParameter</code>	The parameter for which the type is needed
--------------------------	--

Write canonical commands in the output file The different steps are defined as follows:

- Retrieve the name of the output file using [FileOperator::getCanonFile\(\)](#)
- Open the output file and append *message* to it
- Close the output file

Parameters

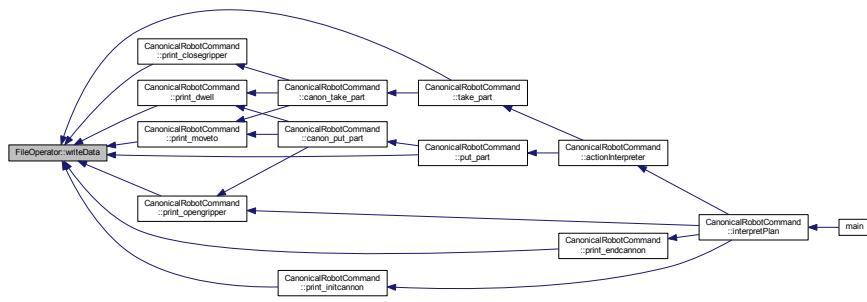
<code>message</code>	Message to write in the file
----------------------	------------------------------

Definition at line 188 of file Operator.cc.

Here is the call graph for this function:



Here is the caller graph for this function:



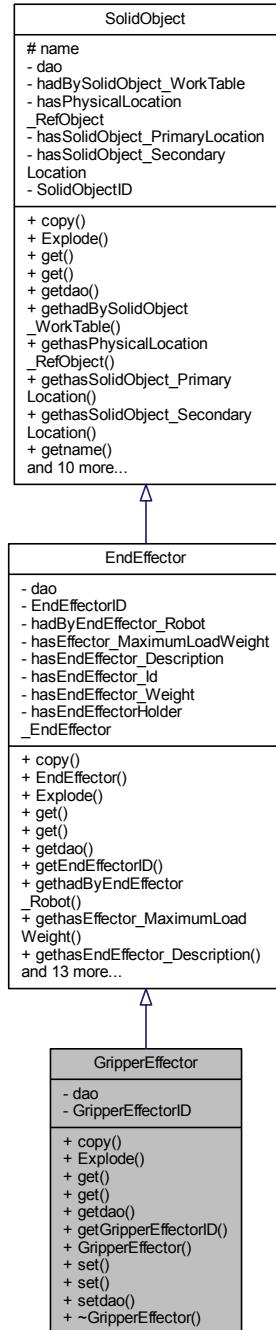
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/[Operator.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/[Operator.cc](#)

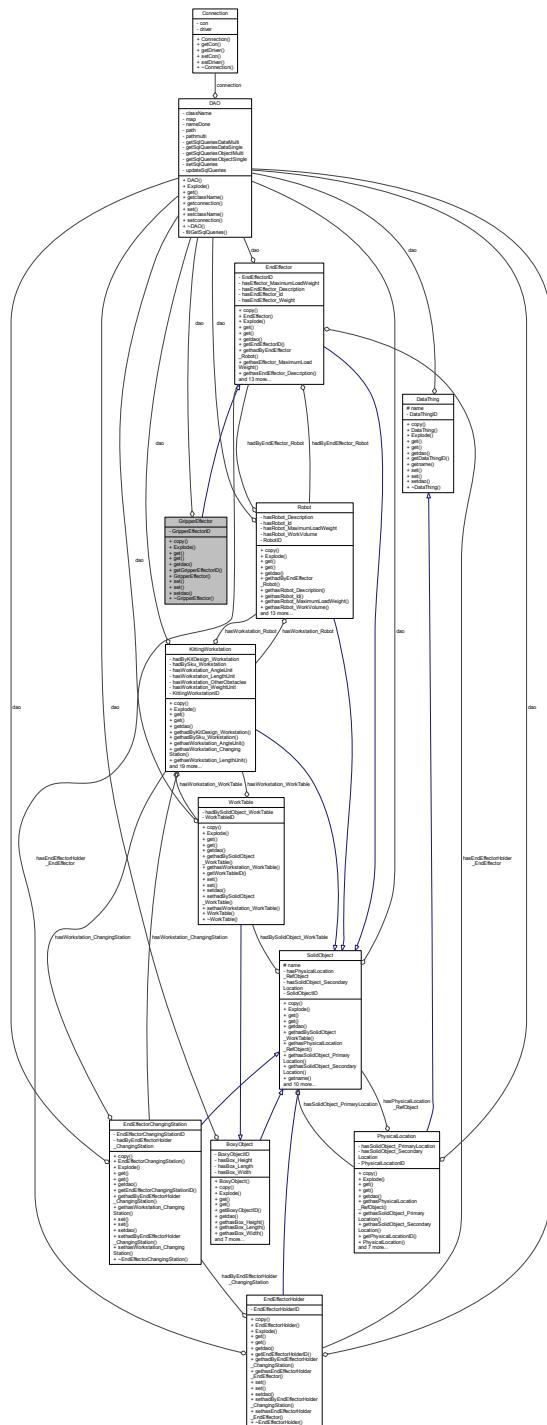
8.17 GripperEffector Class Reference

```
#include <GripperEffector.h>
```

Inheritance diagram for GripperEffector:



Collaboration diagram for GripperEffector:



Public Member Functions

- void `copy` (std::map< std::string, std::string > object)
 - std::vector< std::string > `Explode` (const std::string &str, char separator)

- void `get` (int *id*)
- void `get` (std::string *name*)
- DAO * `getdao` ()
- int `getGripperEffectorID` ()
- GripperEffector (std::string *name*)
- void `set` (int *id*, GripperEffector **obj*)
- void `set` (std::string *name*)
- void `setdao` (DAO *_*dao*)
- ~GripperEffector ()

Private Attributes

- DAO * *dao*
- int `GripperEffectorID`

Additional Inherited Members

8.17.1 Detailed Description

Definition at line 27 of file GripperEffector.h.

8.17.2 Constructor & Destructor Documentation

8.17.2.1 GripperEffector::GripperEffector (std::string *name*)

Definition at line 19 of file GripperEffector.cpp.

8.17.2.2 GripperEffector::~GripperEffector ()

Definition at line 22 of file GripperEffector.cpp.

8.17.3 Member Function Documentation

8.17.3.1 void GripperEffector::copy (std::map< std::string, std::string > *object*)

Definition at line 63 of file GripperEffector.cpp.

Here is the caller graph for this function:



8.17.3.2 `std::vector< std::string > GripperEffector::Explode (const std::string & str, char separator)`

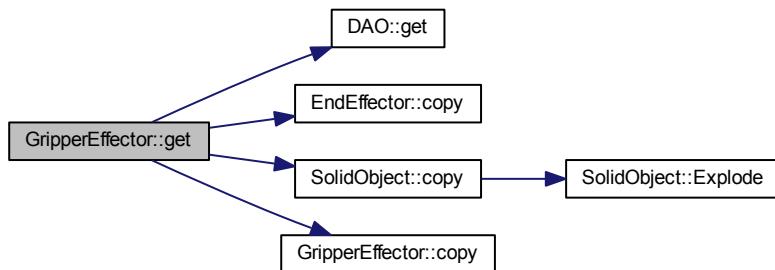
Definition at line 72 of file GripperEffector.cpp.

8.17.3.3 `void GripperEffector::get (int id)`

8.17.3.4 `void GripperEffector::get (std::string name)`

Definition at line 34 of file GripperEffector.cpp.

Here is the call graph for this function:



8.17.3.5 `DAO * GripperEffector::getdao ()`

Definition at line 28 of file GripperEffector.cpp.

8.17.3.6 `int GripperEffector::getGripperEffectorID ()`

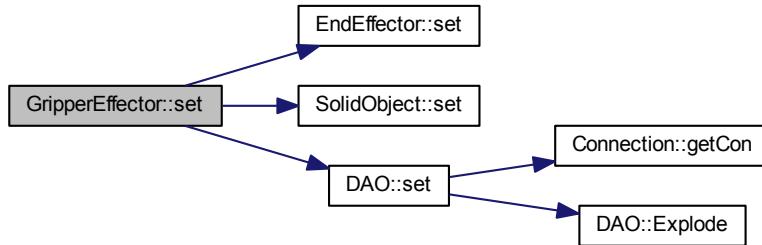
Definition at line 25 of file GripperEffector.cpp.

8.17.3.7 `void GripperEffector::set (int id, GripperEffector * obj)`

8.17.3.8 `void GripperEffector::set (std::string name)`

Definition at line 47 of file GripperEffector.cpp.

Here is the call graph for this function:



8.17.3.9 void GripperEffector::setdao (DAO * _dao)

Definition at line 31 of file `GripperEffector.cpp`.

8.17.4 Member Data Documentation

8.17.4.1 DAO* GripperEffector::dao [private]

Definition at line 29 of file `GripperEffector.h`.

8.17.4.2 int GripperEffector::GripperEffectorID [private]

Definition at line 28 of file `GripperEffector.h`.

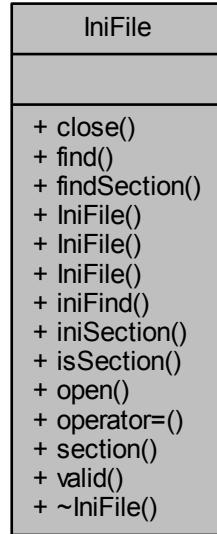
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[GripperEffector.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[GripperEffector.cpp](#)

8.18 IniFile Class Reference

```
#include <IniFile.h>
```

Collaboration diagram for IniFile:



Public Member Functions

- int `close ()`
- const char * `find` (const char *tag, const char *section_to_find=0)
- int `findSection` (void *fp, const char *section)
- `IniFile ()`
- `IniFile` (const char *path)
- `IniFile` (const `IniFile` &)
- const char * `iniFind` (void *fp, const char *tag, const char *section)
- int `iniSection` (void *fp, const char *section, `INIFILE_ENTRY` array[], int max)
- int `isSection` (const char *section)
- int `open` (const char *path)
- `IniFile &` `operator=` (const `IniFile` &)
- int `section` (const char *section_to_load, `INIFILE_ENTRY` array[], int max)
- int `valid ()`
- virtual `~IniFile ()`

8.18.1 Detailed Description

Definition at line 27 of file IniFile.h.

8.18.2 Constructor & Destructor Documentation

8.18.2.1 `IniFile::IniFile()`

Definition at line 10 of file IniFile.cpp.

8.18.2.2 `IniFile::IniFile(const char * path)`

8.18.2.3 `IniFile::~IniFile() [virtual]`

Definition at line 15 of file IniFile.cpp.

8.18.2.4 `IniFile::IniFile(const IniFile &)`

8.18.3 Member Function Documentation

8.18.3.1 `int IniFile::close()`

8.18.3.2 `const char* IniFile::find(const char * tag, const char * section_to_find = 0)`

8.18.3.3 `int IniFile::findSection(void * fp, const char * section)`

Definition at line 86 of file IniFile.cpp.

Here is the call graph for this function:



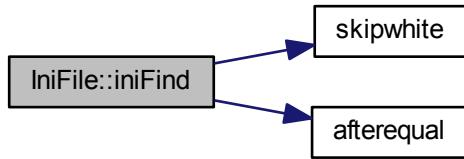
Here is the caller graph for this function:



8.18.3.4 const char * IniFile::iniFind (void * fp, const char * tag, const char * section)

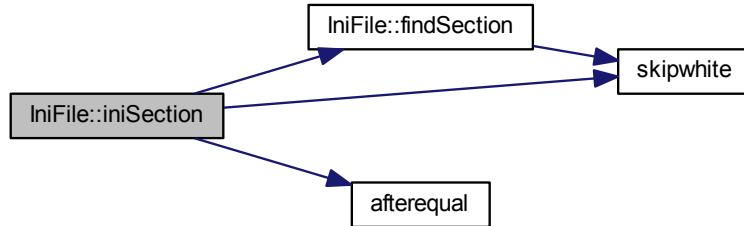
Definition at line 149 of file IniFile.cpp.

Here is the call graph for this function:

**8.18.3.5 int IniFile::iniSection (void * fp, const char * section, INIFILE_ENTRY array[], int max)**

Definition at line 311 of file IniFile.cpp.

Here is the call graph for this function:

**8.18.3.6 int IniFile::isSection (const char * section)****8.18.3.7 int IniFile::open (const char * path)****8.18.3.8 IniFile& IniFile::operator= (const IniFile &)****8.18.3.9 int IniFile::section (const char * section_to_load, INIFILE_ENTRY array[], int max)****8.18.3.10 int IniFile::valid ()**

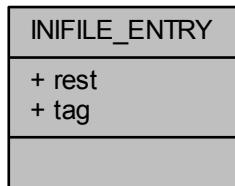
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/[IniFile.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/[IniFile.cpp](#)

8.19 INIFILE_ENTRY Struct Reference

```
#include <IniFile.h>
```

Collaboration diagram for INIFILE_ENTRY:



Public Attributes

- char [rest](#) [[INIFILE_MAX_LINELEN](#)]
- char [tag](#) [[INIFILE_MAX_LINELEN](#)]

8.19.1 Detailed Description

Definition at line 22 of file IniFile.h.

8.19.2 Member Data Documentation

8.19.2.1 char INIFILE_ENTRY::rest[INIFILE_MAX_LINELEN]

Definition at line 24 of file IniFile.h.

8.19.2.2 char INIFILE_ENTRY::tag[INIFILE_MAX_LINELEN]

Definition at line 23 of file IniFile.h.

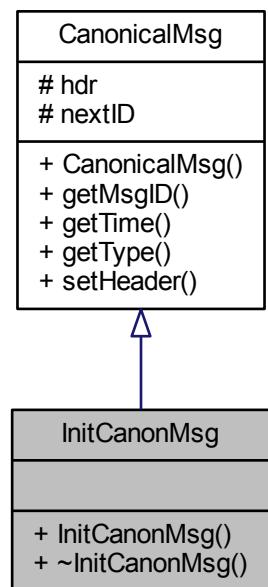
The documentation for this struct was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/[IniFile.h](#)

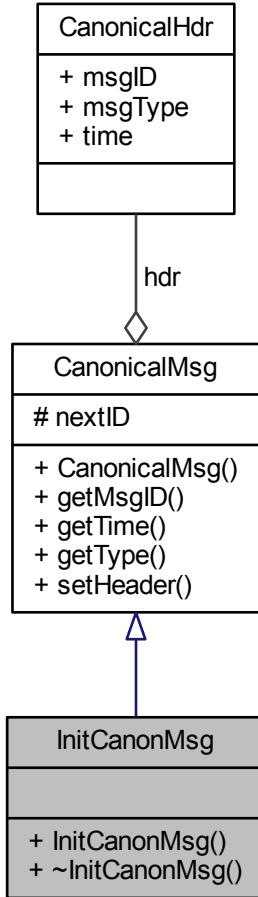
8.20 InitCanonMsg Class Reference

```
#include <canonicalMsg.hh>
```

Inheritance diagram for InitCanonMsg:



Collaboration diagram for InitCanonMsg:



Public Member Functions

- [InitCanonMsg \(\)](#)
- [~InitCanonMsg \(\)](#)

Additional Inherited Members

8.20.1 Detailed Description

Definition at line 87 of file canonicalMsg.hh.

8.20.2 Constructor & Destructor Documentation

8.20.2.1 InitCanonMsg::InitCanonMsg() [inline]

Definition at line 89 of file canonicalMsg.hh.

8.20.2.2 InitCanonMsg::~InitCanonMsg() [inline]

Definition at line 90 of file canonicalMsg.hh.

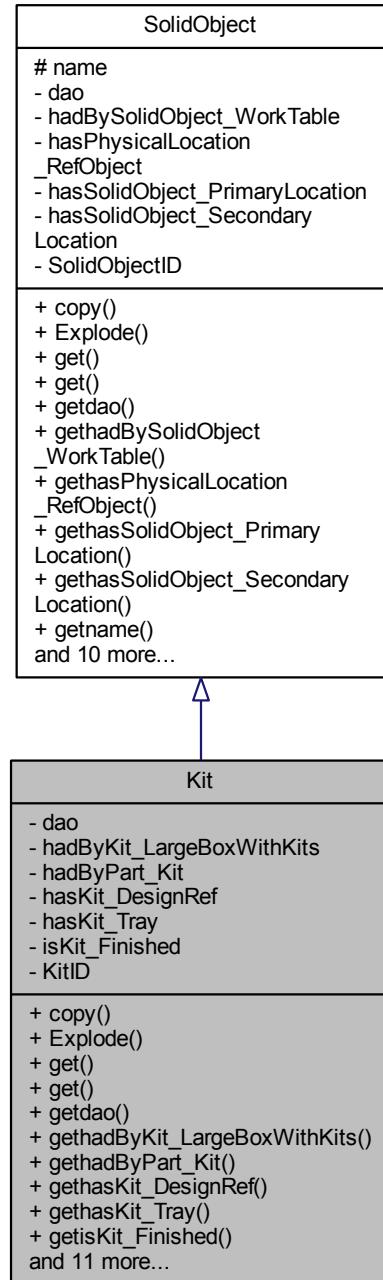
The documentation for this class was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[canonicalMsg.hh](#)

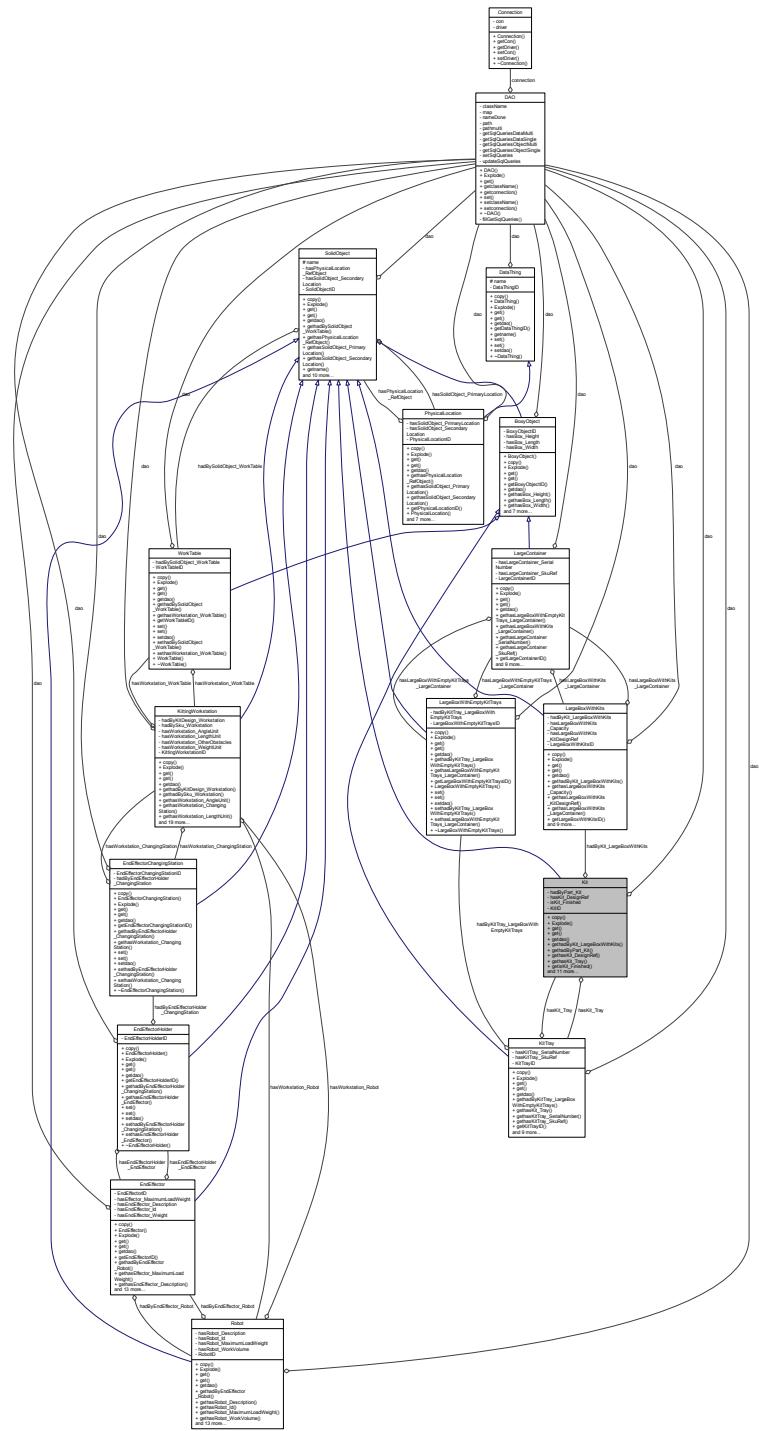
8.21 Kit Class Reference

```
#include <Kit.h>
```

Inheritance diagram for Kit:



Collaboration diagram for Kit:



Public Member Functions

- void **copy** (std::map< std::string, std::string > object)
 - std::vector< std::string > **Explode** (const std::string &str, char separator)

- void `get` (int id)
- void `get` (std::string name)
- DAO * `getdao` ()
- LargeBoxWithKits * `gethadByKit_LargeBoxWithKits` ()
- std::vector< Part * > `gethadByPart_Kit` ()
- std::string `gethasKit_DesignRef` ()
- KitTray * `gethasKit_Tray` ()
- bool `getisKit_Finished` ()
- int `getKitID` ()
- Kit (std::string name)
- void `set` (int id, Kit *obj)
- void `set` (std::string name)
- void `setdao` (DAO *_dao)
- void `sethadByKit_LargeBoxWithKits` (LargeBoxWithKits *_hadByKit_LargeBoxWithKits)
- void `sethadByPart_Kit` (std::vector< Part * > _hadByPart_Kit)
- void `sethasKit_DesignRef` (std::string _hasKit_DesignRef)
- void `sethasKit_Tray` (KitTray *_hasKit_Tray)
- void `setisKit_Finished` (bool _isKit_Finished)
- `~Kit` ()

Private Attributes

- DAO * dao
- LargeBoxWithKits * hadByKit_LargeBoxWithKits
- std::vector< Part * > hadByPart_Kit
- std::string hasKit_DesignRef
- KitTray * hasKit_Tray
- bool isKit_Finished
- int KitID

Additional Inherited Members

8.21.1 Detailed Description

Definition at line 30 of file Kit.h.

8.21.2 Constructor & Destructor Documentation

8.21.2.1 Kit::Kit (std::string name)

Definition at line 22 of file Kit.cpp.

8.21.2.2 Kit::~Kit ()

Definition at line 27 of file Kit.cpp.

8.21.3 Member Function Documentation

8.21.3.1 void Kit::copy (std::map< std::string, std::string > object)

Definition at line 111 of file Kit.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



8.21.3.2 std::vector< std::string > Kit::Explode (const std::string & str, char separator)

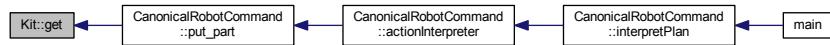
Definition at line 134 of file Kit.cpp.

Here is the caller graph for this function:



8.21.3.3 void Kit::get(int id)

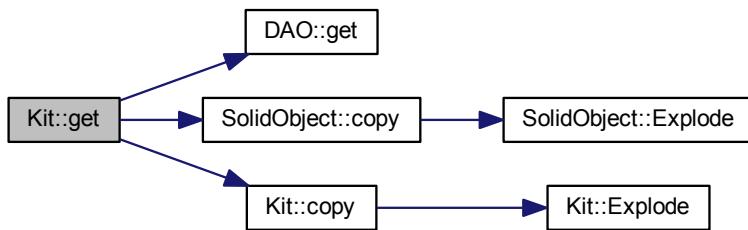
Here is the caller graph for this function:



8.21.3.4 void Kit::get(std::string name)

Definition at line 73 of file Kit.cpp.

Here is the call graph for this function:



8.21.3.5 DAO * Kit::getdao()

Definition at line 43 of file Kit.cpp.

8.21.3.6 LargeBoxWithKits * Kit::gethadByKit_LargeBoxWithKits()

Definition at line 46 of file Kit.cpp.

8.21.3.7 std::vector< Part * > Kit::gethadByPart_Kit()

Definition at line 52 of file Kit.cpp.

8.21.3.8 std::string Kit::gethasKit_DesignRef()

Definition at line 34 of file Kit.cpp.

Here is the caller graph for this function:



8.21.3.9 KitTray * Kit::gethasKit_Tray()

Definition at line 49 of file Kit.cpp.

8.21.3.10 bool Kit::getisKit_Finished()

Definition at line 37 of file Kit.cpp.

8.21.3.11 int Kit::getKitID()

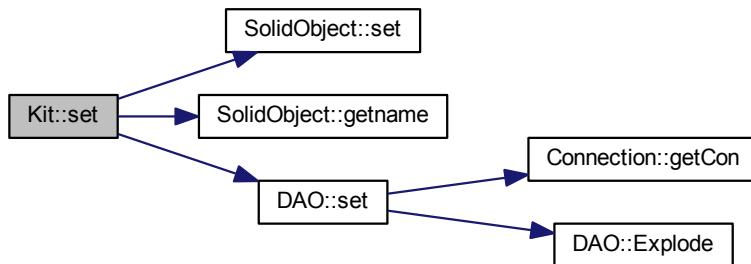
Definition at line 40 of file Kit.cpp.

8.21.3.12 void Kit::set(int id, Kit * obj)

8.21.3.13 void Kit::set(std::string name)

Definition at line 83 of file Kit.cpp.

Here is the call graph for this function:



8.21.3.14 void Kit::setdao(DAO * _dao)

Definition at line 61 of file Kit.cpp.

8.21.3.15 void Kit::sethadByKit_LargeBoxWithKits (LargeBoxWithKits * *_hadByKit_LargeBoxWithKits*)

Definition at line 64 of file Kit.cpp.

8.21.3.16 void Kit::sethadByPart_Kit (std::vector< Part * > *_hadByPart_Kit*)

Definition at line 70 of file Kit.cpp.

8.21.3.17 void Kit::sethasKit_DesignRef (std::string *_hasKit_DesignRef*)

Definition at line 55 of file Kit.cpp.

8.21.3.18 void Kit::sethasKit_Tray (KitTray * *_hasKit_Tray*)

Definition at line 67 of file Kit.cpp.

8.21.3.19 void Kit::setisKit_Finished (bool *_isKit_Finished*)

Definition at line 58 of file Kit.cpp.

8.21.4 Member Data Documentation

8.21.4.1 DAO* Kit::dao [private]

Definition at line 34 of file Kit.h.

8.21.4.2 LargeBoxWithKits* Kit::hadByKit_LargeBoxWithKits [private]

Definition at line 35 of file Kit.h.

8.21.4.3 std::vector<Part*> Kit::hadByPart_Kit [private]

Definition at line 37 of file Kit.h.

8.21.4.4 std::string Kit::hasKit_DesignRef [private]

Definition at line 31 of file Kit.h.

8.21.4.5 KitTray* Kit::hasKit_Tray [private]

Definition at line 36 of file Kit.h.

8.21.4.6 bool Kit::isKit_Finished [private]

Definition at line 32 of file Kit.h.

8.21.4.7 int Kit::KitID [private]

Definition at line 33 of file Kit.h.

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

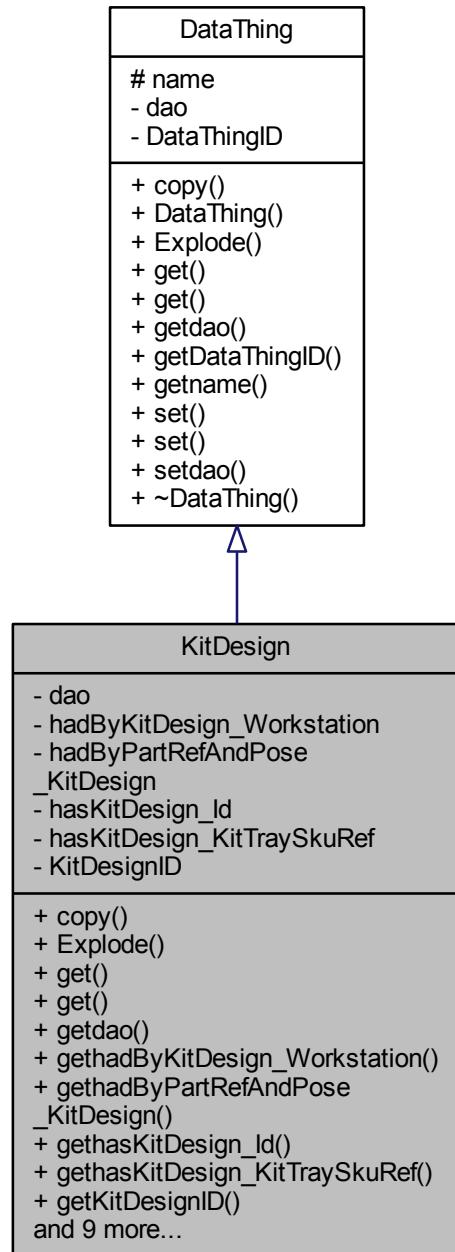
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[Kit.h](#)

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[Kit.cpp](#)

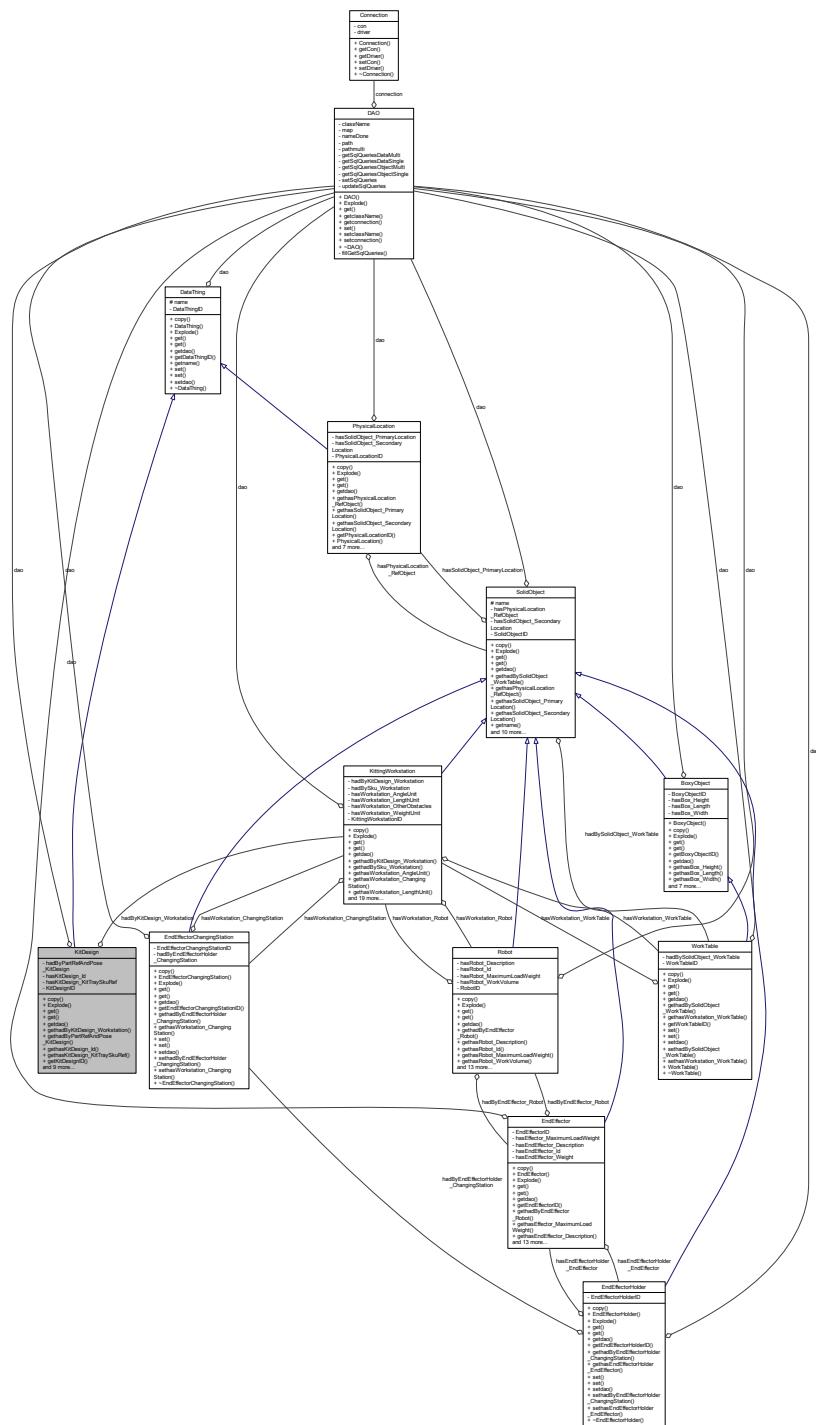
8.22 KitDesign Class Reference

```
#include <KitDesign.h>
```

Inheritance diagram for KitDesign:



Collaboration diagram for KitDesign:



Public Member Functions

- void `copy` (std::map< std::string, std::string > object)
 - std::vector< std::string > `Explode` (const std::string &str, char separator)

- void `get` (int id)
- void `get` (std::string name)
- DAO * `getdao` ()
- KittingWorkstation * `gethadByKitDesign_Workstation` ()
- std::vector< PartRefAndPose * > `gethadByPartRefAndPose_KitDesign` ()
- std::string `gethasKitDesign_Id` ()
- std::string `gethasKitDesign_KitTraySkuRef` ()
- int `getKitDesignID` ()
- KitDesign (std::string name)
- void `set` (int id, KitDesign *obj)
- void `set` (std::string name)
- void `setdao` (DAO *_dao)
- void `sethadByKitDesign_Workstation` (KittingWorkstation *_hadByKitDesign_Workstation)
- void `sethadByPartRefAndPose_KitDesign` (std::vector< PartRefAndPose * > _hadByPartRefAndPose_KitDesign)
- void `sethasKitDesign_Id` (std::string _hasKitDesign_Id)
- void `sethasKitDesign_KitTraySkuRef` (std::string _hasKitDesign_KitTraySkuRef)
- ~KitDesign ()

Private Attributes

- DAO * dao
- KittingWorkstation * hadByKitDesign_Workstation
- std::vector< PartRefAndPose * > hadByPartRefAndPose_KitDesign
- std::string hasKitDesign_Id
- std::string hasKitDesign_KitTraySkuRef
- int KitDesignID

Additional Inherited Members

8.22.1 Detailed Description

Definition at line 29 of file KitDesign.h.

8.22.2 Constructor & Destructor Documentation

8.22.2.1 KitDesign::KitDesign (std::string name)

Definition at line 21 of file KitDesign.cpp.

8.22.2.2 KitDesign::~KitDesign ()

Definition at line 25 of file KitDesign.cpp.

8.22.3 Member Function Documentation

8.22.3.1 void KitDesign::copy (std::map< std::string, std::string > object)

Definition at line 98 of file KitDesign.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



8.22.3.2 std::vector< std::string > KitDesign::Explode (const std::string & str, char separator)

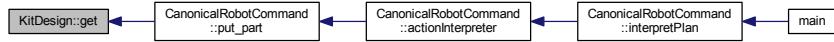
Definition at line 118 of file KitDesign.cpp.

Here is the caller graph for this function:



8.22.3.3 void KitDesign::get(int id)

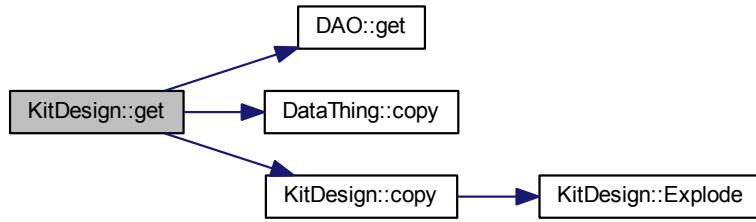
Here is the caller graph for this function:



8.22.3.4 void KitDesign::get(std::string name)

Definition at line 64 of file KitDesign.cpp.

Here is the call graph for this function:



8.22.3.5 DAO * KitDesign::getdao()

Definition at line 40 of file KitDesign.cpp.

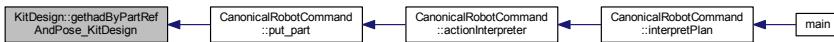
8.22.3.6 KittingWorkstation * KitDesign::gethadByKitDesign_Workstation()

Definition at line 43 of file KitDesign.cpp.

8.22.3.7 std::vector< PartRefAndPose * > KitDesign::gethadByPartRefAndPose_KitDesign()

Definition at line 46 of file KitDesign.cpp.

Here is the caller graph for this function:



8.22.3.8 `std::string KitDesign::gethasKitDesign_Id()`

Definition at line 31 of file KitDesign.cpp.

8.22.3.9 `std::string KitDesign::gethasKitDesign_KitTraySkuRef()`

Definition at line 34 of file KitDesign.cpp.

8.22.3.10 `int KitDesign::getKitDesignID()`

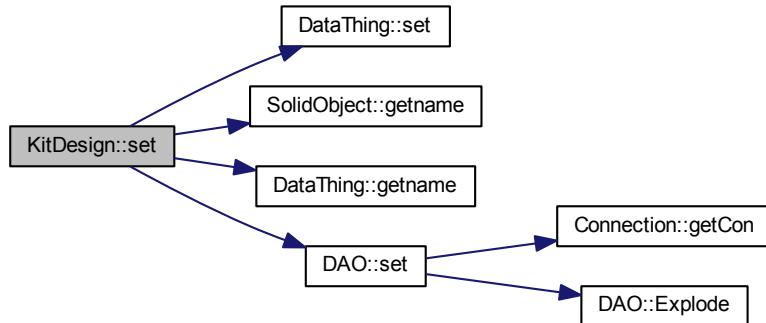
Definition at line 37 of file KitDesign.cpp.

8.22.3.11 `void KitDesign::set(int id, KitDesign * obj)`

8.22.3.12 `void KitDesign::set(std::string name)`

Definition at line 74 of file KitDesign.cpp.

Here is the call graph for this function:



8.22.3.13 `void KitDesign::setdao (DAO * _dao)`

Definition at line 55 of file KitDesign.cpp.

8.22.3.14 `void KitDesign::sethadByKitDesign_Workstation (KittingWorkstation * _hadByKitDesign_Workstation)`

Definition at line 58 of file KitDesign.cpp.

8.22.3.15 `void KitDesign::sethadByPartRefAndPose_KitDesign (std::vector< PartRefAndPose * > _hadByPartRefAndPose_KitDesign)`

Definition at line 61 of file KitDesign.cpp.

8.22.3.16 void KitDesign::sethasKitDesign_Id (std::string *_hasKitDesign_Id*)

Definition at line 49 of file KitDesign.cpp.

8.22.3.17 void KitDesign::sethasKitDesign_KitTraySkuRef (std::string *_hasKitDesign_KitTraySkuRef*)

Definition at line 52 of file KitDesign.cpp.

8.22.4 Member Data Documentation

8.22.4.1 DAO* KitDesign::dao [private]

Definition at line 33 of file KitDesign.h.

8.22.4.2 KittingWorkstation* KitDesign::hadByKitDesign_Workstation [private]

Definition at line 34 of file KitDesign.h.

8.22.4.3 std::vector<PartRefAndPose*> KitDesign::hadByPartRefAndPose_KitDesign [private]

Definition at line 35 of file KitDesign.h.

8.22.4.4 std::string KitDesign::hasKitDesign_Id [private]

Definition at line 30 of file KitDesign.h.

8.22.4.5 std::string KitDesign::hasKitDesign_KitTraySkuRef [private]

Definition at line 31 of file KitDesign.h.

8.22.4.6 int KitDesign::KitDesignID [private]

Definition at line 32 of file KitDesign.h.

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

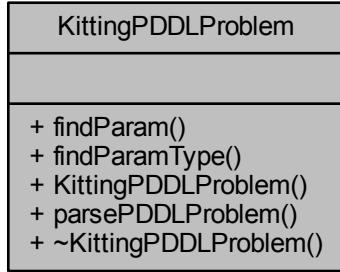
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[KitDesign.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[KitDesign.cpp](#)

8.23 KittingPDDLProblem Class Reference

Class for the Kitting PDDL Problem.

```
#include <KittingPDDLProblem.h>
```

Collaboration diagram for KittingPDDLProblem:



Public Member Functions

- `map< string, int > findParam (string myString, vector< string > myVector)`
Find each string of myVector in myString.
- `void findParamType (std::ifstream &inputfile, map< string, int > myMap, KittingPlan *kittingplan)`
Find the type of each parameter stored in myMap.
- `KittingPDDLProblem ()`
Auto-generated constructor stub.
- `void parsePDDLProblem (const char *filename, KittingPlan *kittingplan)`
Match the type of each parameter in #KittingPlan::m_paramVector by looking into the PDDL Problem File.
- `virtual ~KittingPDDLProblem ()`
Auto-generated destructor stub.

8.23.1 Detailed Description

Class for the Kitting PDDL Problem.

This class is used to manipulate the Kitting PDDL Problem File

Author

Zeid Kootbally zeid.kootbally@nist.gov

Version

1.0

Date

May 17, 2012

Precondition

The kitting PDDL Problem file must be in the etc directory.
 Only one object per line in the Problem PDDL file: r1 - robot kt1 - kitray

Definition at line 35 of file KittingPDDLProblem.h.

8.23.2 Constructor & Destructor Documentation

8.23.2.1 KittingPDDLProblem::KittingPDDLProblem ()

Auto-generated constructor stub.

Definition at line 17 of file KittingPDDLProblem.cc.

8.23.2.2 KittingPDDLProblem::~KittingPDDLProblem () [virtual]

Auto-generated destructor stub.

Definition at line 23 of file KittingPDDLProblem.cc.

8.23.3 Member Function Documentation

8.23.3.1 map< string, int > KittingPDDLProblem::findParam (string *myString*, vector< string > *myVector*)

Find each string of *myVector* in *myString*.

Parameters

<i>myString</i>	String to be searched
<i>myVector</i>	Vector of strings

Returns

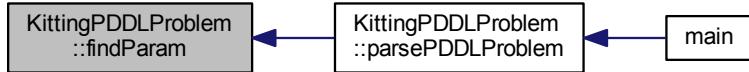
A map that contains each string from *myVector* and the line where it is found

The different steps are:

- For each string in *myVector*
 - Find *s* in *myString*
 - Return the *line* where *s* was found
 - Store *s* and the corresponding *line* in a map

Definition at line 93 of file KittingPDDLProblem.cc.

Here is the caller graph for this function:



8.23.3.2 void KittingPDDLProblem::findParamType (std::ifstream & *infile*, map< string, int > *myMap*, KittingPlan * *kittingplan*)

Find the type of each parameter stored in *myMap*.

Parameters

<i>infile</i>	An ifstream
<i>myMap</i>	A map that stores a string and the corresponding line. <i>myMap</i> is the result from the function KittingPDDLProblem::findParam(string,vector<string>)

Returns

A map that contains each string from *myVector* and the line where it is found

See Also

[KittingPDDLProblem::findParam\(string,vector<string>\)](#)

The different steps are:

- For each *line* in *infile*
 - For each *element* of *myMap*
 - * If *line* equals the second *element* (line number) of *myMap*
 - Retrieves the last element of *line*, which is the *type* of the parameter
 - Store the first *element* of *myMap* (parameter) and its *type* in a KittingPDDLProblem::m_ParamType

Definition at line 50 of file KittingPDDLProblem.cc.

Here is the caller graph for this function:



8.23.3.3 void KittingPDDLProblem::parsePDDLProblem (const char * *filename*, KittingPlan * *kittingplan*)

Match the type of each parameter in #KittingPlan::m_paramVector by looking into the PDDL Problem File.

Parameters

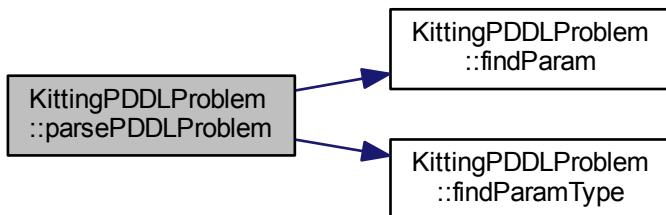
<i>filename</i>	Location of the PDDL Problem File (PDDL_PROBLEM)
-----------------	--

The different steps are:

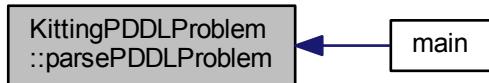
- Copy the content of KittingPlan::m_paramVector in vector *vectorTemp*
- Open the PDDL Problem File
- Copy the content of the file in a string *s*
- For each element of *vectorTemp*
 - Find *vectorTemp[i]* in *s* and the corresponding line *line*
 - Store *vectorTemp[i]* and *line* in a map<string, int> paramLine which looks like: paramLine=map<vectorTemp[i].line>
- Parse the PDDL Problem File
 - For each element *line* in *paramLine*
 - * When the parser reaches *line*, retrieve the whole line
 - * Get the last element of this line, which is basically the type of the parameter *vectorTemp[i]*
 - * Store each parameter *vectorTemp[i]* and its type in KittingPlan::m_ParamTypeList

Definition at line 137 of file KittingPDDLProblem.cc.

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:

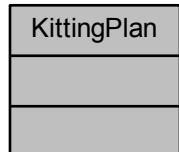
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/[KittingPDDLProblem.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/[KittingPDDLProblem.cc](#)

8.24 KittingPlan Class Reference

Class for the Kitting Plan.

```
#include <KittingPlan.h>
```

Collaboration diagram for KittingPlan:



8.24.1 Detailed Description

Class for the Kitting Plan.

This class is used to manipulate the Kitting Plan File

Author

[Zeid Kootbally](#) zeid.kootbally@nist.gov

Version

1.0

Date

May 17, 2012

Precondition

Make sure the kitting Plan Instance file is in etc.
Make sure the kitting Plan Instance file is not empty.

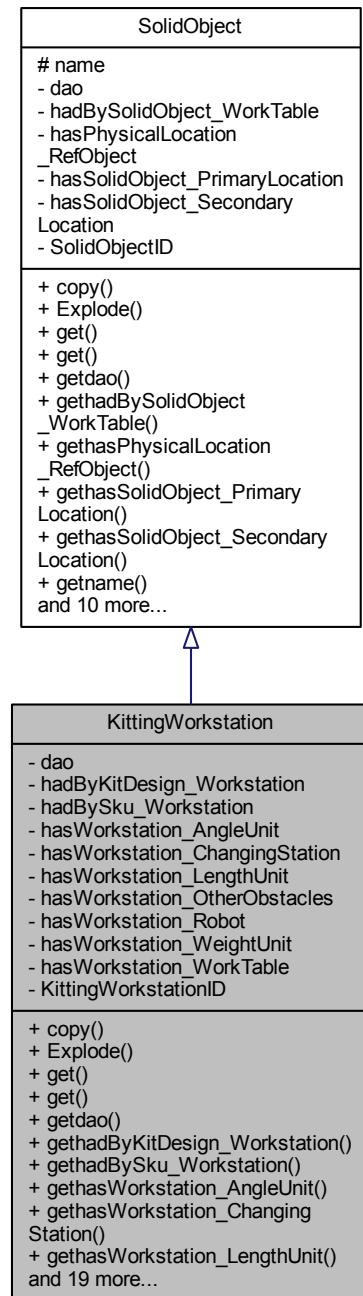
The documentation for this class was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/[KittingPlan.h](#)

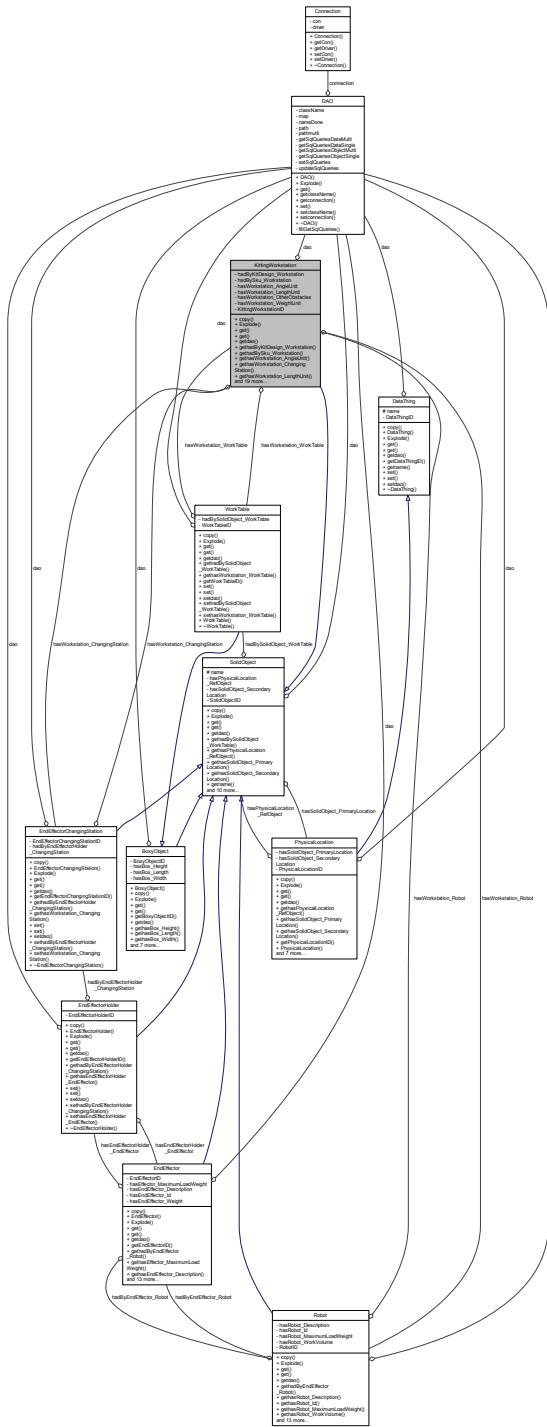
8.25 KittingWorkstation Class Reference

```
#include <KittingWorkstation.h>
```

Inheritance diagram for KittingWorkstation:



Collaboration diagram for KittingWorkstation:



Public Member Functions

- void **copy** (std::map< std::string, std::string > object)
 - std::vector< std::string > **Explode** (const std::string &str, char separator)

- void `get` (int id)
- void `get` (std::string name)
- DAO * `getdao` ()
- std::vector< KitDesign * > `gethadByKitDesign_Workstation` ()
- std::vector< StockKeepingUnit * > `gethadBySku_Workstation` ()
- std::string `gethasWorkstation_AngleUnit` ()
- EndEffectorChangingStation * `gethasWorkstation_ChangingStation` ()
- std::string `gethasWorkstation_LengthUnit` ()
- std::vector< BoxVolume * > `gethasWorkstation_OtherObstacles` ()
- Robot * `gethasWorkstation_Robot` ()
- std::string `gethasWorkstation_WeightUnit` ()
- WorkTable * `gethasWorkstation_WorkTable` ()
- int `getKittingWorkstationID` ()
- KittingWorkstation (std::string name)
- void `set` (int id, KittingWorkstation *obj)
- void `set` (std::string name)
- void `setdao` (DAO *_dao)
- void `sethadByKitDesign_Workstation` (std::vector< KitDesign * > _hadByKitDesign_Workstation)
- void `sethadBySku_Workstation` (std::vector< StockKeepingUnit * > _hadBySku_Workstation)
- void `sethasWorkstation_AngleUnit` (std::string _hasWorkstation_AngleUnit)
- void `sethasWorkstation_ChangingStation` (EndEffectorChangingStation * _hasWorkstation_ChangingStation)
- void `sethasWorkstation_LengthUnit` (std::string _hasWorkstation_LengthUnit)
- void `sethasWorkstation_OtherObstacles` (std::vector< BoxVolume * > _hasWorkstation_OtherObstacles)
- void `sethasWorkstation_Robot` (Robot * _hasWorkstation_Robot)
- void `sethasWorkstation_WeightUnit` (std::string _hasWorkstation_WeightUnit)
- void `sethasWorkstation_WorkTable` (WorkTable * _hasWorkstation_WorkTable)
- ~KittingWorkstation ()

Private Attributes

- DAO * `dao`
- std::vector< KitDesign * > `hadByKitDesign_Workstation`
- std::vector< StockKeepingUnit * > `hadBySku_Workstation`
- std::string `hasWorkstation_AngleUnit`
- EndEffectorChangingStation * `hasWorkstation_ChangingStation`
- std::string `hasWorkstation_LengthUnit`
- std::vector< BoxVolume * > `hasWorkstation_OtherObstacles`
- Robot * `hasWorkstation_Robot`
- std::string `hasWorkstation_WeightUnit`
- WorkTable * `hasWorkstation_WorkTable`
- int `KittingWorkstationID`

Additional Inherited Members

8.25.1 Detailed Description

Definition at line 33 of file KittingWorkstation.h.

8.25.2 Constructor & Destructor Documentation

8.25.2.1 KittingWorkstation::KittingWorkstation (std::string name)

Definition at line 25 of file KittingWorkstation.cpp.

8.25.2.2 KittingWorkstation::~KittingWorkstation ()

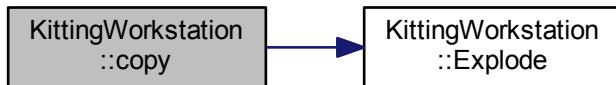
Definition at line 31 of file KittingWorkstation.cpp.

8.25.3 Member Function Documentation

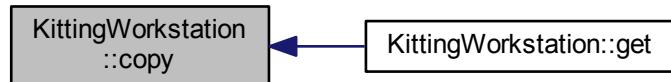
8.25.3.1 void KittingWorkstation::copy (std::map< std::string, std::string > object)

Definition at line 157 of file KittingWorkstation.cpp.

Here is the call graph for this function:



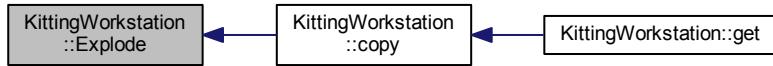
Here is the caller graph for this function:



8.25.3.2 std::vector< std::string > KittingWorkstation::Explode (const std::string & str, char separator)

Definition at line 196 of file KittingWorkstation.cpp.

Here is the caller graph for this function:

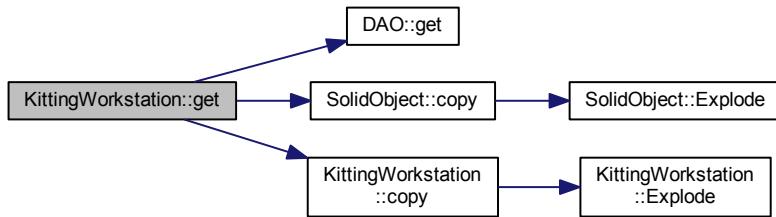


8.25.3.3 void KittingWorkstation::get (int id)

8.25.3.4 void KittingWorkstation::get (std::string name)

Definition at line 106 of file KittingWorkstation.cpp.

Here is the call graph for this function:



8.25.3.5 DAO * KittingWorkstation::getdao ()

Definition at line 55 of file KittingWorkstation.cpp.

8.25.3.6 std::vector< KitDesign * > KittingWorkstation::gethadByKitDesign_Workstation ()

Definition at line 58 of file KittingWorkstation.cpp.

8.25.3.7 std::vector< StockKeepingUnit * > KittingWorkstation::gethadBySku_Workstation ()

Definition at line 70 of file KittingWorkstation.cpp.

8.25.3.8 std::string KittingWorkstation::gethasWorkstation_AngleUnit ()

Definition at line 46 of file KittingWorkstation.cpp.

8.25.3.9 EndEffectorChangingStation * KittingWorkstation::gethasWorkstation_ChangingStation()

Definition at line 61 of file KittingWorkstation.cpp.

8.25.3.10 std::string KittingWorkstation::gethasWorkstation_LengthUnit()

Definition at line 43 of file KittingWorkstation.cpp.

8.25.3.11 std::vector< BoxVolume * > KittingWorkstation::gethasWorkstation_OtherObstacles()

Definition at line 67 of file KittingWorkstation.cpp.

8.25.3.12 Robot * KittingWorkstation::gethasWorkstation_Robot()

Definition at line 73 of file KittingWorkstation.cpp.

8.25.3.13 std::string KittingWorkstation::gethasWorkstation_WeightUnit()

Definition at line 49 of file KittingWorkstation.cpp.

8.25.3.14 WorkTable * KittingWorkstation::gethasWorkstation_WorkTable()

Definition at line 64 of file KittingWorkstation.cpp.

8.25.3.15 int KittingWorkstation::getKittingWorkstationID()

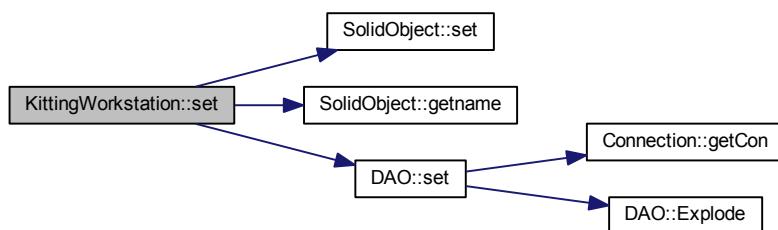
Definition at line 52 of file KittingWorkstation.cpp.

8.25.3.16 void KittingWorkstation::set(int *id*, KittingWorkstation * *obj*)

8.25.3.17 void KittingWorkstation::set(std::string *name*)

Definition at line 116 of file KittingWorkstation.cpp.

Here is the call graph for this function:



8.25.3.18 void KittingWorkstation::setdao (DAO * *_dao*)

Definition at line 85 of file KittingWorkstation.cpp.

8.25.3.19 void KittingWorkstation::sethadByKitDesign_Workstation (std::vector< KitDesign * > *_hadByKitDesign_Workstation*)

Definition at line 88 of file KittingWorkstation.cpp.

8.25.3.20 void KittingWorkstation::sethadBySku_Workstation (std::vector< StockKeepingUnit * > *_hadBySku_Workstation*)

Definition at line 100 of file KittingWorkstation.cpp.

8.25.3.21 void KittingWorkstation::sethasWorkstation_AngleUnit (std::string *_hasWorkstation_AngleUnit*)

Definition at line 79 of file KittingWorkstation.cpp.

8.25.3.22 void KittingWorkstation::sethasWorkstation_ChangingStation (EndEffectorChangingStation * *_hasWorkstation_ChangingStation*)

Definition at line 91 of file KittingWorkstation.cpp.

8.25.3.23 void KittingWorkstation::sethasWorkstation_LengthUnit (std::string *_hasWorkstation_LengthUnit*)

Definition at line 76 of file KittingWorkstation.cpp.

8.25.3.24 void KittingWorkstation::sethasWorkstation_OtherObstacles (std::vector< BoxVolume * > *_hasWorkstation_OtherObstacles*)

Definition at line 97 of file KittingWorkstation.cpp.

8.25.3.25 void KittingWorkstation::sethasWorkstation_Robot (Robot * *_hasWorkstation_Robot*)

Definition at line 103 of file KittingWorkstation.cpp.

8.25.3.26 void KittingWorkstation::sethasWorkstation_WeightUnit (std::string *_hasWorkstation_WeightUnit*)

Definition at line 82 of file KittingWorkstation.cpp.

8.25.3.27 void KittingWorkstation::sethasWorkstation_WorkTable (WorkTable * *_hasWorkstation_WorkTable*)

Definition at line 94 of file KittingWorkstation.cpp.

8.25.4 Member Data Documentation

8.25.4.1 **DAO* KittingWorkstation::dao** [private]

Definition at line 38 of file KittingWorkstation.h.

8.25.4.2 **std::vector<KitDesign*> KittingWorkstation::hadByKitDesign_Workstation** [private]

Definition at line 39 of file KittingWorkstation.h.

8.25.4.3 **std::vector<StockKeepingUnit*> KittingWorkstation::hadBySku_Workstation** [private]

Definition at line 43 of file KittingWorkstation.h.

8.25.4.4 **std::string KittingWorkstation::hasWorkstation_AngleUnit** [private]

Definition at line 35 of file KittingWorkstation.h.

8.25.4.5 **EndEffectorChangingStation* KittingWorkstation::hasWorkstation_ChangingStation** [private]

Definition at line 40 of file KittingWorkstation.h.

8.25.4.6 **std::string KittingWorkstation::hasWorkstation_LengthUnit** [private]

Definition at line 34 of file KittingWorkstation.h.

8.25.4.7 **std::vector<BoxVolume*> KittingWorkstation::hasWorkstation_OtherObstacles** [private]

Definition at line 42 of file KittingWorkstation.h.

8.25.4.8 **Robot* KittingWorkstation::hasWorkstation_Robot** [private]

Definition at line 44 of file KittingWorkstation.h.

8.25.4.9 **std::string KittingWorkstation::hasWorkstation_WeightUnit** [private]

Definition at line 36 of file KittingWorkstation.h.

8.25.4.10 **WorkTable* KittingWorkstation::hasWorkstation_WorkTable** [private]

Definition at line 41 of file KittingWorkstation.h.

8.25.4.11 **int KittingWorkstation::KittingWorkstationID** [private]

Definition at line 37 of file KittingWorkstation.h.

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

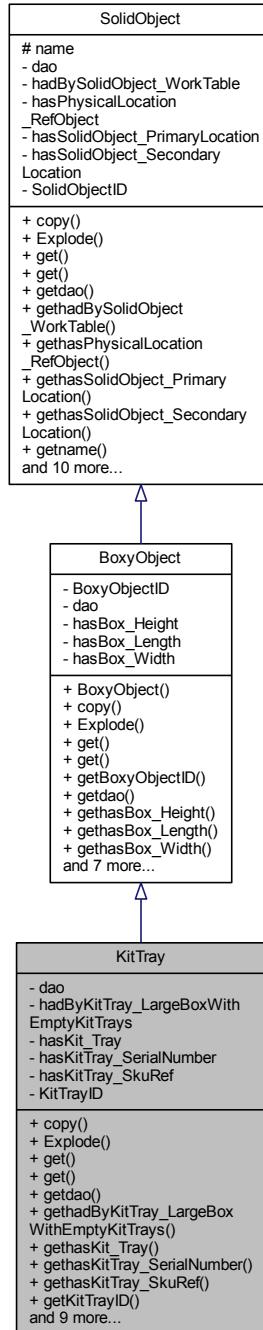
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[KittingWorkstation.h](#)

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[KittingWorkstation.cpp](#)

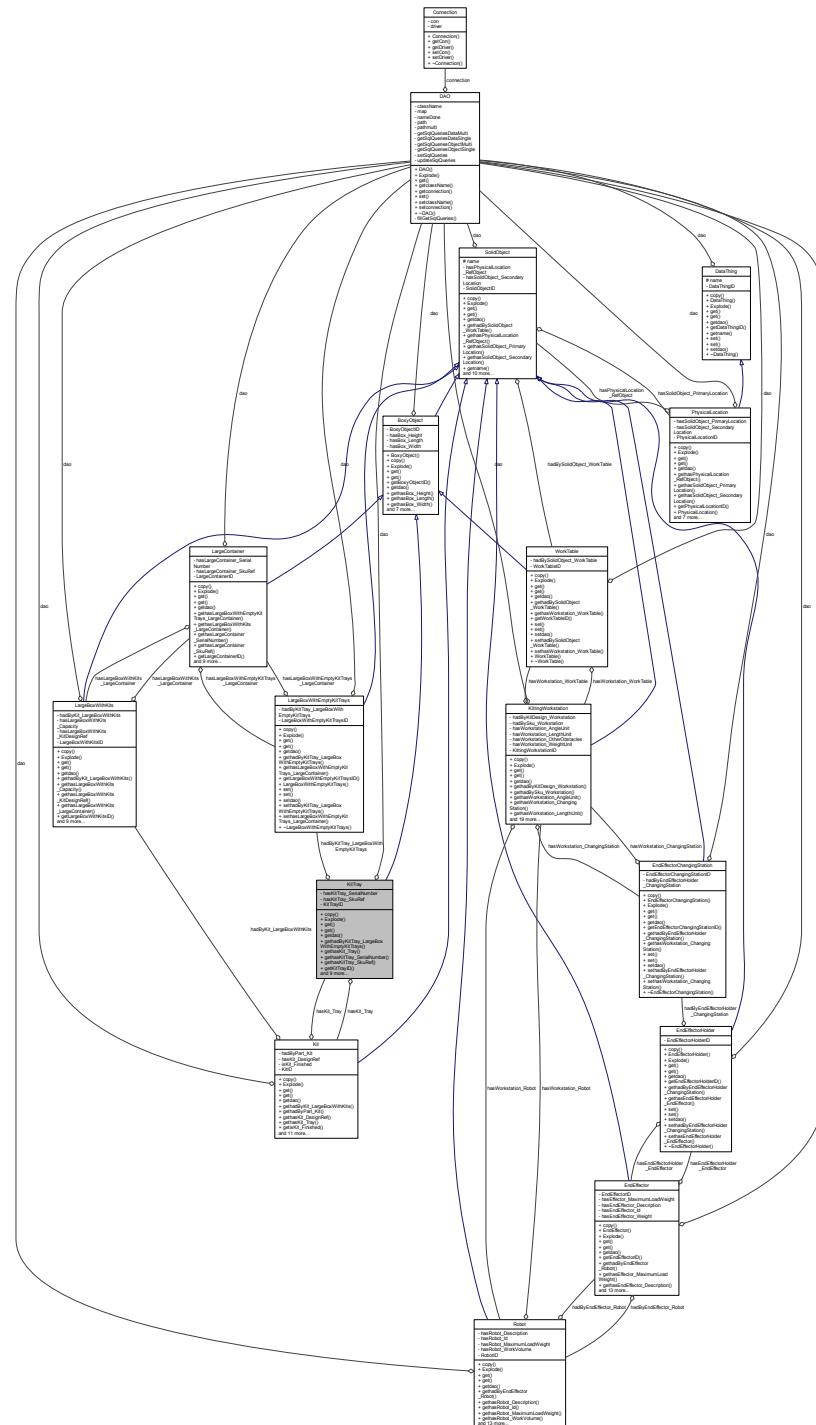
8.26 KitTray Class Reference

```
#include <KitTray.h>
```

Inheritance diagram for KitTray:



Collaboration diagram for KitTray:



Public Member Functions

- void **copy** (std::map< std::string, std::string > object)
 - std::vector< std::string > **Explode** (const std::string &str, char separator)

- void `get` (int id)
- void `get` (std::string name)
- DAO * `getdao` ()
- LargeBoxWithEmptyKitTrays * `gethadByKitTray_LargeBoxWithEmptyKitTrays` ()
- Kit * `gethasKit_Tray` ()
- std::string `gethasKitTray_SerialNumber` ()
- std::string `gethasKitTray_SkuRef` ()
- int `getKitTrayID` ()
- KitTray (std::string name)
- void `set` (int id, KitTray *obj)
- void `set` (std::string name)
- void `setdao` (DAO *_dao)
- void `sethadByKitTray_LargeBoxWithEmptyKitTrays` (LargeBoxWithEmptyKitTrays *_hadByKitTray_LargeBoxWithEmptyKitTrays)
- void `sethasKit_Tray` (Kit *_hasKit_Tray)
- void `sethasKitTray_SerialNumber` (std::string _hasKitTray_SerialNumber)
- void `sethasKitTray_SkuRef` (std::string _hasKitTray_SkuRef)
- ~KitTray ()

Private Attributes

- DAO * dao
- LargeBoxWithEmptyKitTrays * hadByKitTray_LargeBoxWithEmptyKitTrays
- Kit * hasKit_Tray
- std::string hasKitTray_SerialNumber
- std::string hasKitTray_SkuRef
- int KitTrayID

Additional Inherited Members

8.26.1 Detailed Description

Definition at line 29 of file KitTray.h.

8.26.2 Constructor & Destructor Documentation

8.26.2.1 KitTray::KitTray (std::string name)

Definition at line 21 of file KitTray.cpp.

8.26.2.2 KitTray::~KitTray ()

Definition at line 26 of file KitTray.cpp.

8.26.3 Member Function Documentation

8.26.3.1 void KitTray::copy (std::map< std::string, std::string > object)

Definition at line 99 of file KitTray.cpp.

Here is the caller graph for this function:



8.26.3.2 std::vector< std::string > KitTray::Explode (const std::string & str, char separator)

Definition at line 116 of file KitTray.cpp.

8.26.3.3 void KitTray::get (int id)

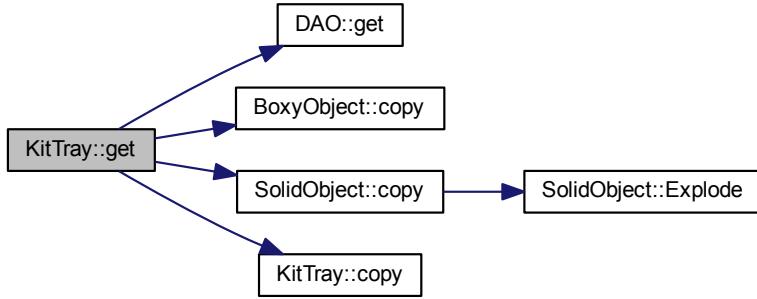
Here is the caller graph for this function:



8.26.3.4 void KitTray::get (std::string name)

Definition at line 64 of file KitTray.cpp.

Here is the call graph for this function:



8.26.3.5 DAO * KitTray::getdao()

Definition at line 40 of file KitTray.cpp.

8.26.3.6 LargeBoxWithEmptyKitTrays * KitTray::gethadByKitTray_LargeBoxWithEmptyKitTrays()

Definition at line 46 of file KitTray.cpp.

8.26.3.7 Kit * KitTray::gethasKit_Tray()

Definition at line 43 of file KitTray.cpp.

8.26.3.8 std::string KitTray::gethasKitTray_SerialNumber()

Definition at line 34 of file KitTray.cpp.

8.26.3.9 std::string KitTray::gethasKitTray_SkuRef()

Definition at line 31 of file KitTray.cpp.

8.26.3.10 int KitTray::getKitTrayID()

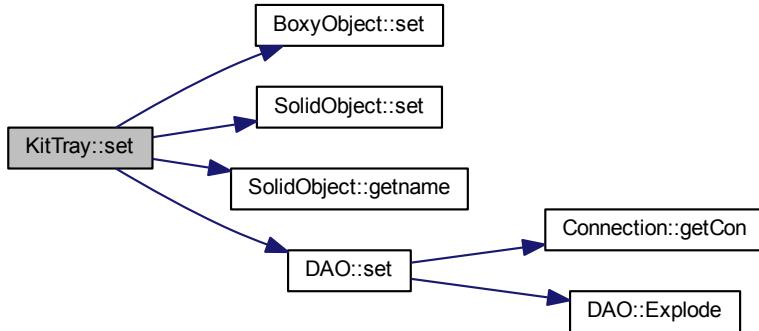
Definition at line 37 of file KitTray.cpp.

8.26.3.11 void KitTray::set(int id, KitTray * obj)

8.26.3.12 void KitTray::set(std::string name)

Definition at line 77 of file KitTray.cpp.

Here is the call graph for this function:



8.26.3.13 void KitTray::setdao (DAO * *_dao*)

Definition at line 55 of file KitTray.cpp.

8.26.3.14 void KitTray::sethadByKitTray_LargeBoxWithEmptyKitTrays (LargeBoxWithEmptyKitTrays * *_hadByKitTray_LargeBoxWithEmptyKitTrays*)

Definition at line 61 of file KitTray.cpp.

8.26.3.15 void KitTray::sethasKit_Tray (Kit * *_hasKit_Tray*)

Definition at line 58 of file KitTray.cpp.

8.26.3.16 void KitTray::sethasKitTray_SerialNumber (std::string *_hasKitTray_SerialNumber*)

Definition at line 52 of file KitTray.cpp.

8.26.3.17 void KitTray::sethasKitTray_SkuRef (std::string *_hasKitTray_SkuRef*)

Definition at line 49 of file KitTray.cpp.

8.26.4 Member Data Documentation

8.26.4.1 DAO* KitTray::*dao* [private]

Definition at line 33 of file KitTray.h.

8.26.4.2 LargeBoxWithEmptyKitTrays* KitTray::hadByKitTray_LargeBoxWithEmptyKitTrays [private]

Definition at line 35 of file KitTray.h.

8.26.4.3 Kit* KitTray::hasKit_Tray [private]

Definition at line 34 of file KitTray.h.

8.26.4.4 std::string KitTray::hasKitTray_SerialNumber [private]

Definition at line 31 of file KitTray.h.

8.26.4.5 std::string KitTray::hasKitTray_SkuRef [private]

Definition at line 30 of file KitTray.h.

8.26.4.6 int KitTray::KitTrayID [private]

Definition at line 32 of file KitTray.h.

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

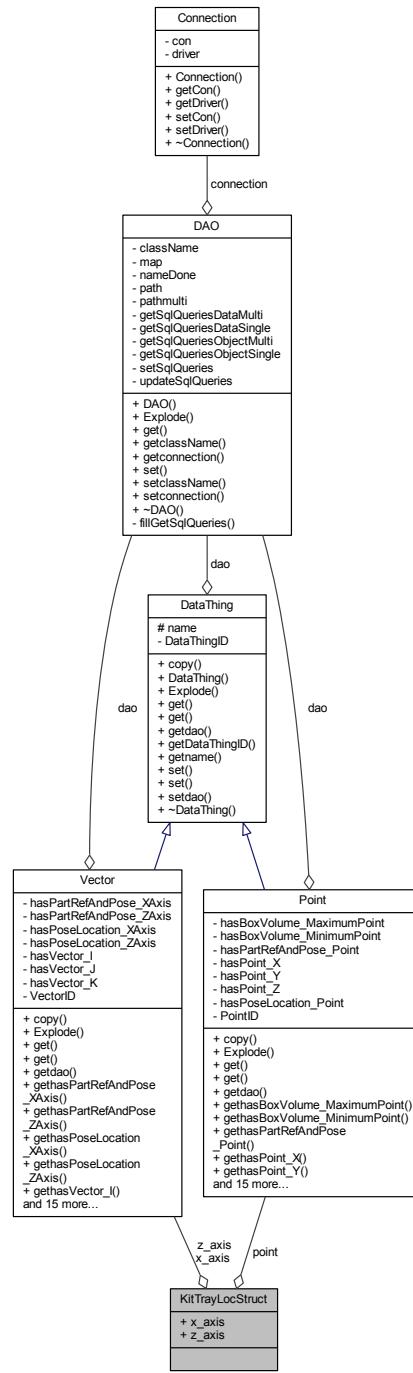
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[KitTray.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[KitTray.cpp](#)

8.27 KitTrayLocStruct Struct Reference

This structure represents a kit tray location.

```
#include <Structdef.h>
```

Collaboration diagram for KitTrayLocStruct:



Public Attributes

- [Point * point](#)
- [Vector * x_axis](#)

- [Vector * z_axis](#)

8.27.1 Detailed Description

This structure represents a kit tray location.

Definition at line 67 of file Structdef.h.

8.27.2 Member Data Documentation

8.27.2.1 KitTrayLocStruct::point

Member *point* contains the coordinates of the kit tray

Definition at line 69 of file Structdef.h.

8.27.2.2 KitTrayLocStruct::x_axis

Member *x_axis* represents the X axis for the kit tray

Definition at line 70 of file Structdef.h.

8.27.2.3 KitTrayLocStruct::z_axis

Member *z_axis* represents the Z axis for the kit tray

Definition at line 71 of file Structdef.h.

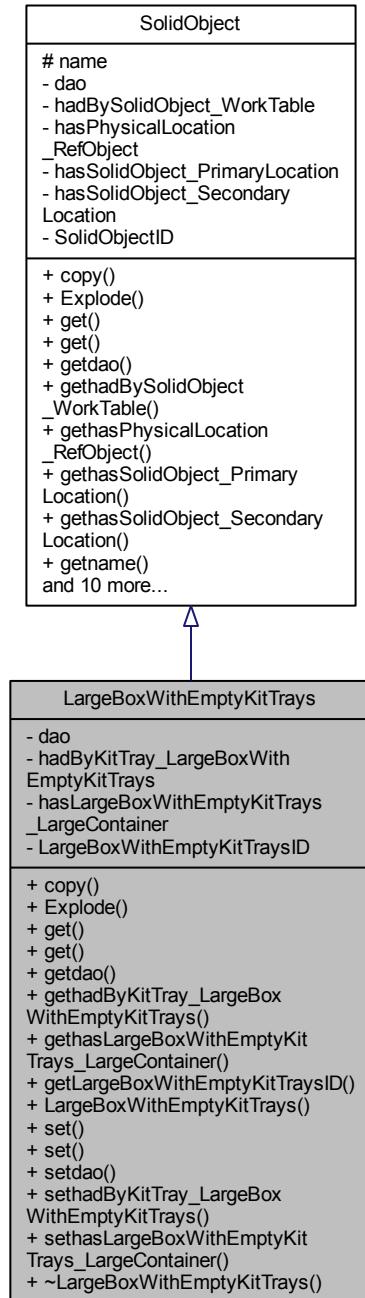
The documentation for this struct was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/[Structdef.h](#)

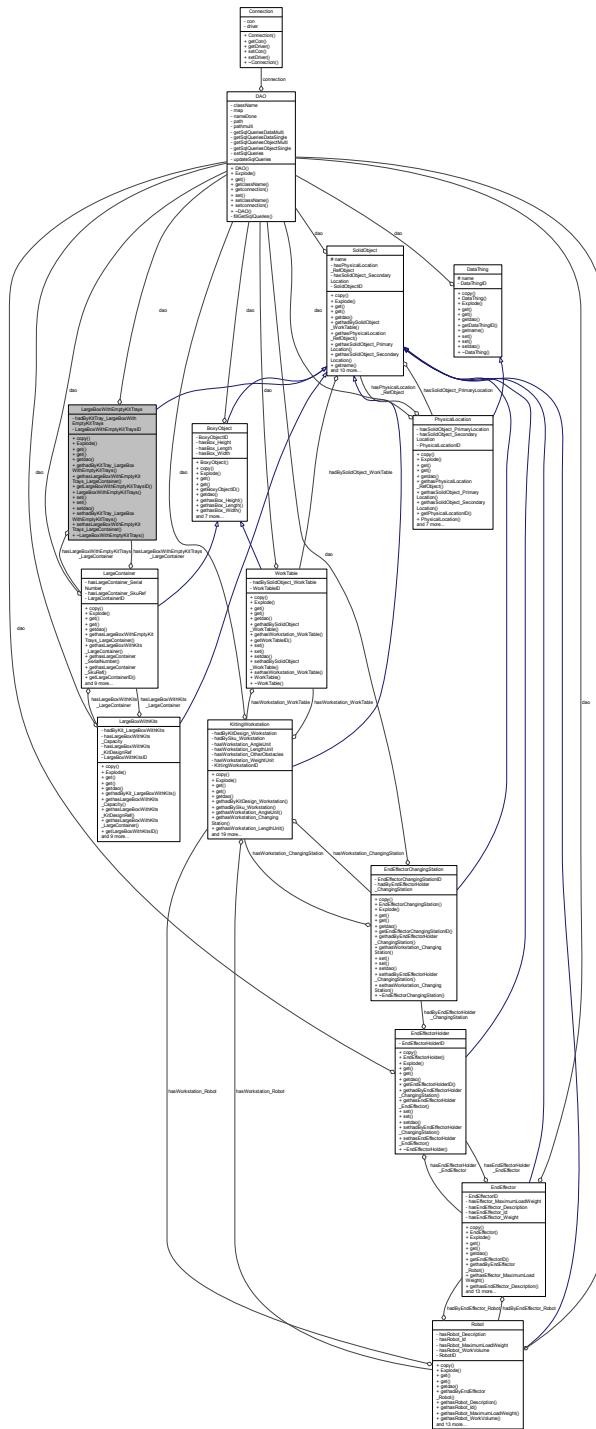
8.28 LargeBoxWithEmptyKitTrays Class Reference

```
#include <LargeBoxWithEmptyKitTrays.h>
```

Inheritance diagram for LargeBoxWithEmptyKitTrays:



Collaboration diagram for LargeBoxWithEmptyKitTrays:



Public Member Functions

- void **copy** (std::map< std::string, std::string > object)
 - std::vector< std::string > **Explode** (const std::string &str, char separator)

- void [get](#) (int id)
- void [get](#) (std::string name)
- DAO * [getdao](#) ()
- std::vector< KitTray * > [gethadByKitTray_LargeBoxWithEmptyKitTrays](#) ()
- LargeContainer * [gethasLargeBoxWithEmptyKitTrays_LargeContainer](#) ()
- int [getLargeBoxWithEmptyKitTraysID](#) ()
- LargeBoxWithEmptyKitTrays (std::string name)
- void [set](#) (int id, LargeBoxWithEmptyKitTrays *obj)
- void [set](#) (std::string name)
- void [setdao](#) (DAO *_dao)
- void [sethadByKitTray_LargeBoxWithEmptyKitTrays](#) (std::vector< KitTray * > _hadByKitTray_LargeBoxWithEmptyKitTrays)
- void [sethasLargeBoxWithEmptyKitTrays_LargeContainer](#) (LargeContainer * _hasLargeBoxWithEmptyKitTrays_LargeContainer)
- ~LargeBoxWithEmptyKitTrays ()

Private Attributes

- DAO * dao
- std::vector< KitTray * > hadByKitTray_LargeBoxWithEmptyKitTrays
- LargeContainer * hasLargeBoxWithEmptyKitTrays_LargeContainer
- int LargeBoxWithEmptyKitTraysID

Additional Inherited Members

8.28.1 Detailed Description

Definition at line 29 of file LargeBoxWithEmptyKitTrays.h.

8.28.2 Constructor & Destructor Documentation

8.28.2.1 LargeBoxWithEmptyKitTrays::LargeBoxWithEmptyKitTrays (std::string name)

Definition at line 21 of file LargeBoxWithEmptyKitTrays.cpp.

8.28.2.2 LargeBoxWithEmptyKitTrays::~LargeBoxWithEmptyKitTrays ()

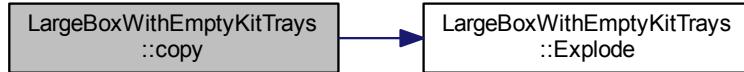
Definition at line 25 of file LargeBoxWithEmptyKitTrays.cpp.

8.28.3 Member Function Documentation

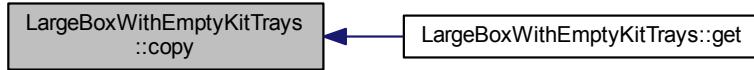
8.28.3.1 void LargeBoxWithEmptyKitTrays::copy (std::map< std::string, std::string > object)

Definition at line 84 of file LargeBoxWithEmptyKitTrays.cpp.

Here is the call graph for this function:



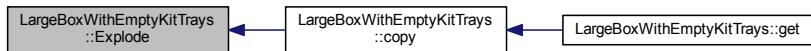
Here is the caller graph for this function:



8.28.3.2 `std::vector< std::string > LargeBoxWithEmptyKitTrays::Explode (const std::string & str, char separator)`

Definition at line 102 of file LargeBoxWithEmptyKitTrays.cpp.

Here is the caller graph for this function:

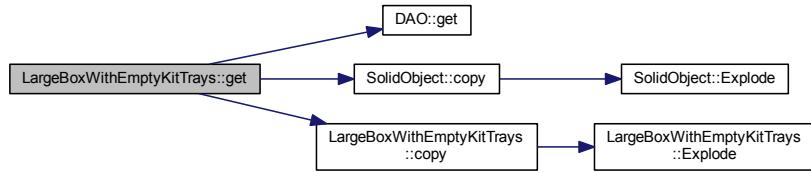


8.28.3.3 `void LargeBoxWithEmptyKitTrays::get (int id)`

8.28.3.4 `void LargeBoxWithEmptyKitTrays::get (std::string name)`

Definition at line 52 of file LargeBoxWithEmptyKitTrays.cpp.

Here is the call graph for this function:



8.28.3.5 DAO * LargeBoxWithEmptyKitTrays::getdao ()

Definition at line 34 of file LargeBoxWithEmptyKitTrays.cpp.

8.28.3.6 std::vector< KitTray * > LargeBoxWithEmptyKitTrays::gethadByKitTray_LargeBoxWithEmptyKitTrays ()

Definition at line 40 of file LargeBoxWithEmptyKitTrays.cpp.

8.28.3.7 LargeContainer * LargeBoxWithEmptyKitTrays::gethasLargeBoxWithEmptyKitTrays_LargeContainer ()

Definition at line 37 of file LargeBoxWithEmptyKitTrays.cpp.

8.28.3.8 int LargeBoxWithEmptyKitTrays::getLargeBoxWithEmptyKitTraysID ()

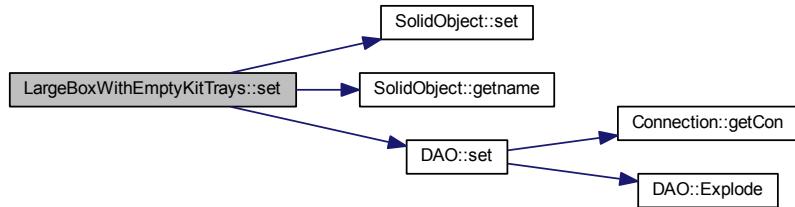
Definition at line 31 of file LargeBoxWithEmptyKitTrays.cpp.

8.28.3.9 void LargeBoxWithEmptyKitTrays::set (int id, LargeBoxWithEmptyKitTrays * obj)

8.28.3.10 void LargeBoxWithEmptyKitTrays::set (std::string name)

Definition at line 62 of file LargeBoxWithEmptyKitTrays.cpp.

Here is the call graph for this function:



8.28.3.11 void LargeBoxWithEmptyKitTrays::setdao (DAO * *_dao*)

Definition at line 43 of file LargeBoxWithEmptyKitTrays.cpp.

8.28.3.12 void LargeBoxWithEmptyKitTrays::sethadByKitTray_LargeBoxWithEmptyKitTrays (std::vector< KitTray * > *_hadByKitTray_LargeBoxWithEmptyKitTrays*)

Definition at line 49 of file LargeBoxWithEmptyKitTrays.cpp.

8.28.3.13 void LargeBoxWithEmptyKitTrays::sethasLargeBoxWithEmptyKitTrays_LargeContainer (LargeContainer * *_hasLargeBoxWithEmptyKitTrays_LargeContainer*)

Definition at line 46 of file LargeBoxWithEmptyKitTrays.cpp.

8.28.4 Member Data Documentation

8.28.4.1 DAO* LargeBoxWithEmptyKitTrays::*dao* [private]

Definition at line 31 of file LargeBoxWithEmptyKitTrays.h.

8.28.4.2 std::vector<KitTray*> LargeBoxWithEmptyKitTrays::*hadByKitTray_LargeBoxWithEmptyKitTrays* [private]

Definition at line 33 of file LargeBoxWithEmptyKitTrays.h.

8.28.4.3 LargeContainer* LargeBoxWithEmptyKitTrays::*hasLargeBoxWithEmptyKitTrays_LargeContainer* [private]

Definition at line 32 of file LargeBoxWithEmptyKitTrays.h.

8.28.4.4 int LargeBoxWithEmptyKitTrays::*LargeBoxWithEmptyKitTraysID* [private]

Definition at line 30 of file LargeBoxWithEmptyKitTrays.h.

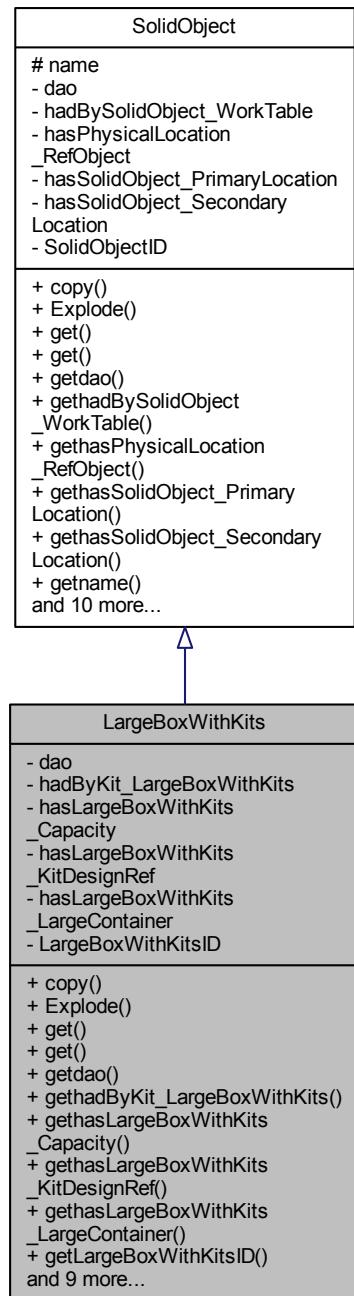
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[LargeBoxWithEmptyKitTrays.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[LargeBoxWithEmptyKitTrays.cpp](#)

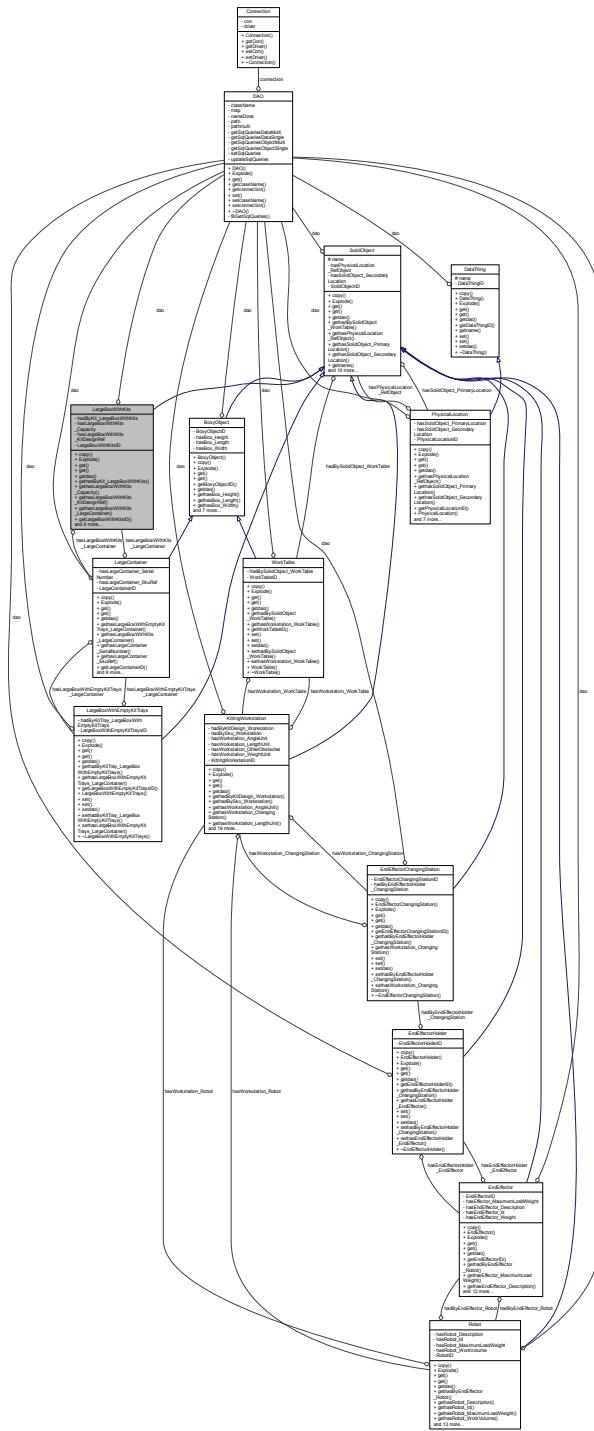
8.29 LargeBoxWithKits Class Reference

```
#include <LargeBoxWithKits.h>
```

Inheritance diagram for LargeBoxWithKits:



Collaboration diagram for LargeBoxWithKits:



Public Member Functions

- void **copy** (std::map< std::string, std::string > object)
 - std::vector< std::string > **Explode** (const std::string &str, char separator)

- void [get](#) (int id)
- void [get](#) (std::string name)
- DAO * [getdao](#) ()
- std::vector< Kit * > [gethadByKit_LargeBoxWithKits](#) ()
- std::string [gethasLargeBoxWithKits_Capacity](#) ()
- std::string [gethasLargeBoxWithKits_KitDesignRef](#) ()
- LargeContainer * [gethasLargeBoxWithKits_LargeContainer](#) ()
- int [getLargeBoxWithKitsID](#) ()
- LargeBoxWithKits ([std::string name](#))
- void [set](#) (int id, LargeBoxWithKits *obj)
- void [set](#) (std::string name)
- void [setdao](#) (DAO *_dao)
- void [sethadByKit_LargeBoxWithKits](#) (std::vector< Kit * > _hadByKit_LargeBoxWithKits)
- void [sethasLargeBoxWithKits_Capacity](#) (std::string _hasLargeBoxWithKits_Capacity)
- void [sethasLargeBoxWithKits_KitDesignRef](#) (std::string _hasLargeBoxWithKits_KitDesignRef)
- void [sethasLargeBoxWithKits_LargeContainer](#) (LargeContainer * _hasLargeBoxWithKits_LargeContainer)
- ~LargeBoxWithKits ()

Private Attributes

- DAO * [dao](#)
- std::vector< Kit * > [hadByKit_LargeBoxWithKits](#)
- std::string [hasLargeBoxWithKits_Capacity](#)
- std::string [hasLargeBoxWithKits_KitDesignRef](#)
- LargeContainer * [hasLargeBoxWithKits_LargeContainer](#)
- int [LargeBoxWithKitsID](#)

Additional Inherited Members

8.29.1 Detailed Description

Definition at line 29 of file LargeBoxWithKits.h.

8.29.2 Constructor & Destructor Documentation

8.29.2.1 LargeBoxWithKits::LargeBoxWithKits (std::string name)

Definition at line 21 of file LargeBoxWithKits.cpp.

8.29.2.2 LargeBoxWithKits::~LargeBoxWithKits ()

Definition at line 25 of file LargeBoxWithKits.cpp.

8.29.3 Member Function Documentation

8.29.3.1 void LargeBoxWithKits::copy (std::map< std::string, std::string > object)

Definition at line 98 of file LargeBoxWithKits.cpp.

Here is the call graph for this function:



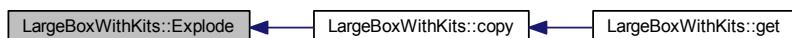
Here is the caller graph for this function:



8.29.3.2 std::vector< std::string > LargeBoxWithKits::Explode (const std::string & str, char separator)

Definition at line 118 of file LargeBoxWithKits.cpp.

Here is the caller graph for this function:

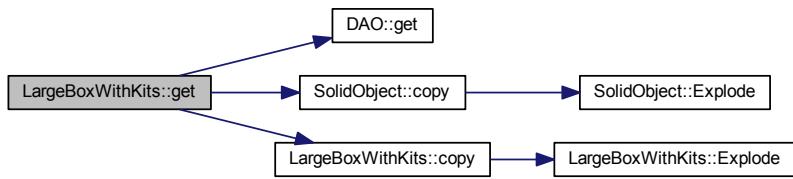


8.29.3.3 void LargeBoxWithKits::get (int id)

8.29.3.4 void LargeBoxWithKits::get (std::string name)

Definition at line 64 of file LargeBoxWithKits.cpp.

Here is the call graph for this function:



8.29.3.5 `DAO * LargeBoxWithKits::getdao()`

Definition at line 40 of file LargeBoxWithKits.cpp.

8.29.3.6 `std::vector< Kit * > LargeBoxWithKits::gethadByKit_LargeBoxWithKits()`

Definition at line 46 of file LargeBoxWithKits.cpp.

8.29.3.7 `std::string LargeBoxWithKits::gethasLargeBoxWithKits_Capacity()`

Definition at line 31 of file LargeBoxWithKits.cpp.

8.29.3.8 `std::string LargeBoxWithKits::gethasLargeBoxWithKits_KitDesignRef()`

Definition at line 34 of file LargeBoxWithKits.cpp.

8.29.3.9 `LargeContainer * LargeBoxWithKits::gethasLargeBoxWithKits_LargeContainer()`

Definition at line 43 of file LargeBoxWithKits.cpp.

8.29.3.10 `int LargeBoxWithKits::getLargeBoxWithKitsID()`

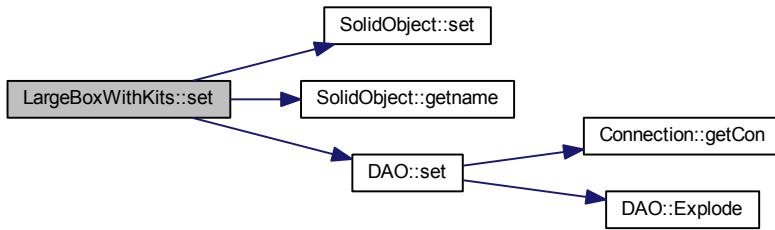
Definition at line 37 of file LargeBoxWithKits.cpp.

8.29.3.11 `void LargeBoxWithKits::set(int id, LargeBoxWithKits * obj)`

8.29.3.12 `void LargeBoxWithKits::set(std::string name)`

Definition at line 74 of file LargeBoxWithKits.cpp.

Here is the call graph for this function:



8.29.3.13 void LargeBoxWithKits::setdao (DAO * _dao)

Definition at line 55 of file LargeBoxWithKits.cpp.

8.29.3.14 void LargeBoxWithKits::sethadByKit_LargeBoxWithKits (std::vector< Kit * > _hadByKit_LargeBoxWithKits)

Definition at line 61 of file LargeBoxWithKits.cpp.

8.29.3.15 void LargeBoxWithKits::sethasLargeBoxWithKits_Capacity (std::string _hasLargeBoxWithKits_Capacity)

Definition at line 49 of file LargeBoxWithKits.cpp.

8.29.3.16 void LargeBoxWithKits::sethasLargeBoxWithKits_KitDesignRef (std::string _hasLargeBoxWithKits_KitDesignRef)

Definition at line 52 of file LargeBoxWithKits.cpp.

8.29.3.17 void LargeBoxWithKits::sethasLargeBoxWithKits_LargeContainer (LargeContainer * _hasLargeBoxWithKits_LargeContainer)

Definition at line 58 of file LargeBoxWithKits.cpp.

8.29.4 Member Data Documentation

8.29.4.1 DAO* LargeBoxWithKits::dao [private]

Definition at line 33 of file LargeBoxWithKits.h.

8.29.4.2 std::vector<Kit*> LargeBoxWithKits::hadByKit_LargeBoxWithKits [private]

Definition at line 35 of file LargeBoxWithKits.h.

8.29.4.3 std::string LargeBoxWithKits::hasLargeBoxWithKits_Capacity [private]

Definition at line 30 of file LargeBoxWithKits.h.

8.29.4.4 std::string LargeBoxWithKits::hasLargeBoxWithKits_KitDesignRef [private]

Definition at line 31 of file LargeBoxWithKits.h.

8.29.4.5 LargeContainer* LargeBoxWithKits::hasLargeBoxWithKits_LargeContainer [private]

Definition at line 34 of file LargeBoxWithKits.h.

8.29.4.6 int LargeBoxWithKits::LargeBoxWithKitsID [private]

Definition at line 32 of file LargeBoxWithKits.h.

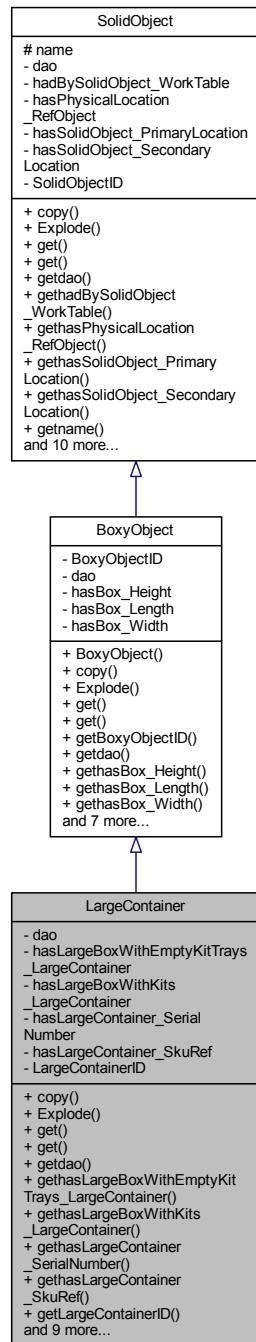
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[LargeBoxWithKits.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[LargeBoxWithKits.cpp](#)

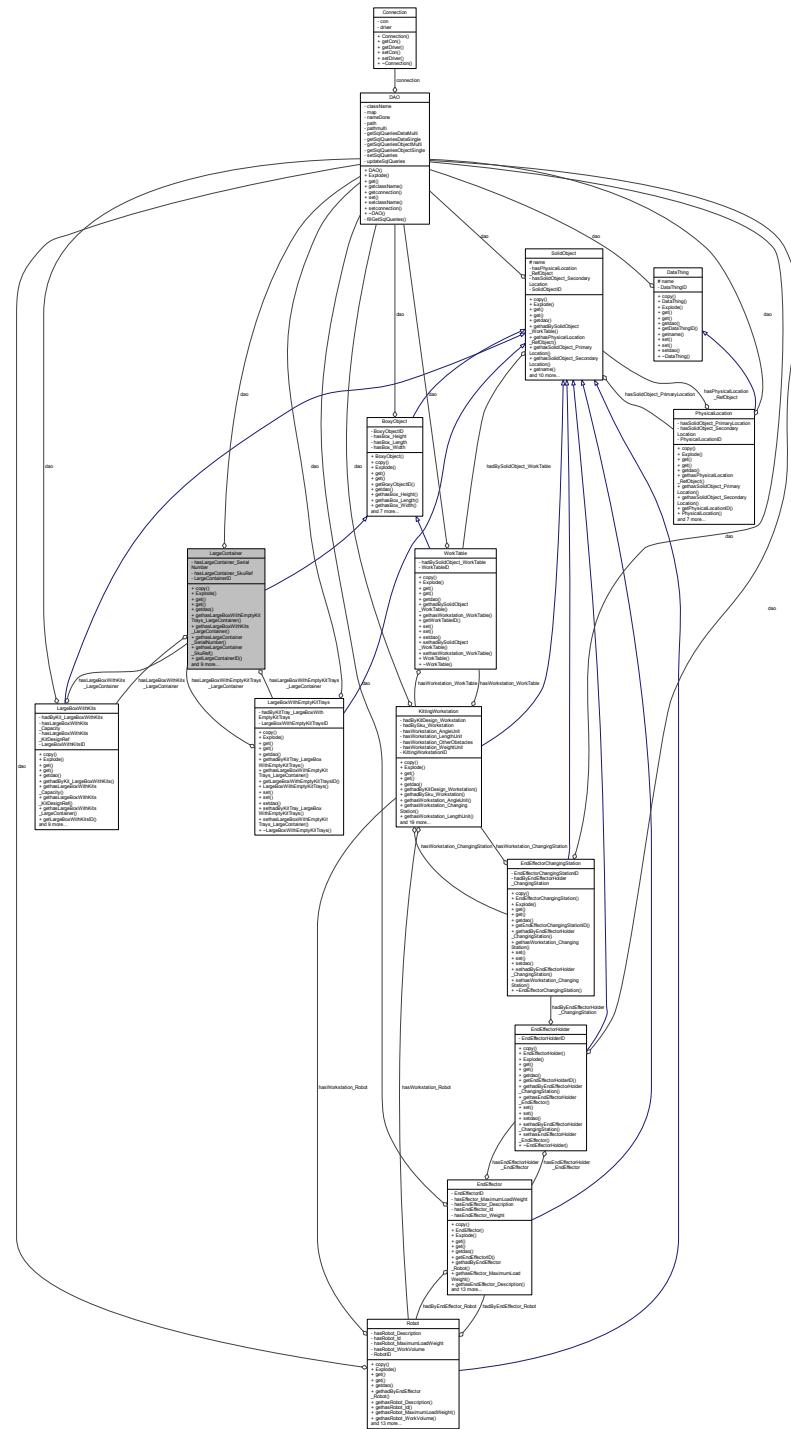
8.30 LargeContainer Class Reference

```
#include <LargeContainer.h>
```

Inheritance diagram for LargeContainer:



Collaboration diagram for LargeContainer:



Public Member Functions

- void `copy` (std::map< std::string, std::string > object)
 - std::vector< std::string > `Explode` (const std::string &str, char separator)

- void `get` (int id)
- void `get` (std::string name)
- DAO * `getdao` ()
- LargeBoxWithEmptyKitTrays * `gethasLargeBoxWithEmptyKitTrays_LargeContainer` ()
- LargeBoxWithKits * `gethasLargeBoxWithKits_LargeContainer` ()
- std::string `gethasLargeContainer_SerialNumber` ()
- std::string `gethasLargeContainer_SkuRef` ()
- int `getLargeContainerID` ()
- LargeContainer (std::string name)
- void `set` (int id, LargeContainer *obj)
- void `set` (std::string name)
- void `setdao` (DAO *_dao)
- void `sethasLargeBoxWithEmptyKitTrays_LargeContainer` (LargeBoxWithEmptyKitTrays *_hasLargeBoxWithEmptyKitTrays_LargeContainer)
- void `sethasLargeBoxWithKits_LargeContainer` (LargeBoxWithKits *_hasLargeBoxWithKits_LargeContainer)
- void `sethasLargeContainer_SerialNumber` (std::string _hasLargeContainer_SerialNumber)
- void `sethasLargeContainer_SkuRef` (std::string _hasLargeContainer_SkuRef)
- ~LargeContainer ()

Private Attributes

- DAO * dao
- LargeBoxWithEmptyKitTrays * hasLargeBoxWithEmptyKitTrays_LargeContainer
- LargeBoxWithKits * hasLargeBoxWithKits_LargeContainer
- std::string hasLargeContainer_SerialNumber
- std::string hasLargeContainer_SkuRef
- int LargeContainerID

Additional Inherited Members

8.30.1 Detailed Description

Definition at line 29 of file LargeContainer.h.

8.30.2 Constructor & Destructor Documentation

8.30.2.1 LargeContainer::LargeContainer (std::string name)

Definition at line 21 of file LargeContainer.cpp.

8.30.2.2 LargeContainer::~LargeContainer ()

Definition at line 26 of file LargeContainer.cpp.

8.30.3 Member Function Documentation

8.30.3.1 void LargeContainer::copy (std::map< std::string, std::string > object)

Definition at line 99 of file LargeContainer.cpp.

Here is the caller graph for this function:



8.30.3.2 std::vector< std::string > LargeContainer::Explode (const std::string & str, char separator)

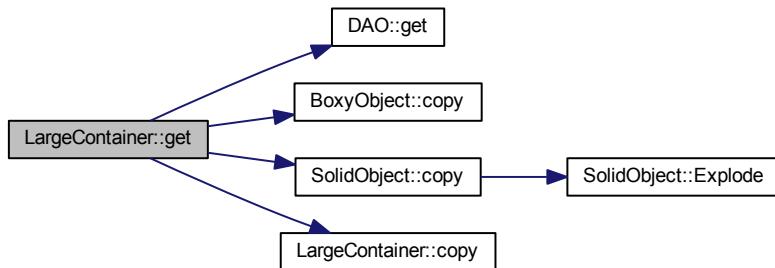
Definition at line 116 of file LargeContainer.cpp.

8.30.3.3 void LargeContainer::get (int id)

8.30.3.4 void LargeContainer::get (std::string name)

Definition at line 64 of file LargeContainer.cpp.

Here is the call graph for this function:



8.30.3.5 DAO * LargeContainer::getdao ()

Definition at line 40 of file LargeContainer.cpp.

8.30.3.6 `LargeBoxWithEmptyKitTrays * LargeContainer::gethasLargeBoxWithEmptyKitTrays_LargeContainer()`

Definition at line 46 of file LargeContainer.cpp.

8.30.3.7 `LargeBoxWithKits * LargeContainer::gethasLargeBoxWithKits_LargeContainer()`

Definition at line 43 of file LargeContainer.cpp.

8.30.3.8 `std::string LargeContainer::gethasLargeContainer_SerialNumber()`

Definition at line 34 of file LargeContainer.cpp.

8.30.3.9 `std::string LargeContainer::gethasLargeContainer_SkuRef()`

Definition at line 31 of file LargeContainer.cpp.

8.30.3.10 `int LargeContainer::getLargeContainerID()`

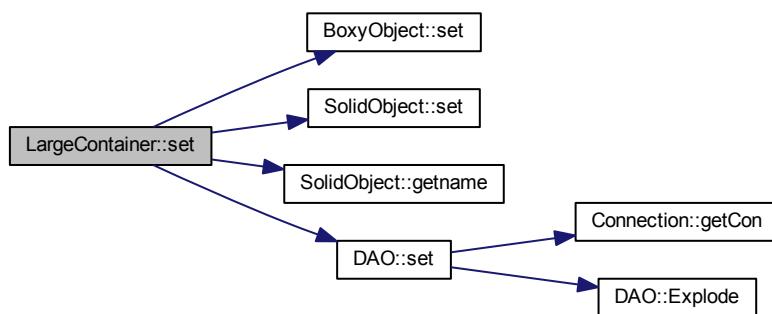
Definition at line 37 of file LargeContainer.cpp.

8.30.3.11 `void LargeContainer::set(int id, LargeContainer * obj)`

8.30.3.12 `void LargeContainer::set(std::string name)`

Definition at line 77 of file LargeContainer.cpp.

Here is the call graph for this function:



8.30.3.13 `void LargeContainer::setdao(DAO * _dao)`

Definition at line 55 of file LargeContainer.cpp.

8.30.3.14 void LargeContainer::sethasLargeBoxWithEmptyKitTrays_LargeContainer (LargeBoxWithEmptyKitTrays *
_hasLargeBoxWithEmptyKitTrays_LargeContainer)

Definition at line 61 of file LargeContainer.cpp.

8.30.3.15 void LargeContainer::sethasLargeBoxWithKits_LargeContainer (LargeBoxWithKits *
_hasLargeBoxWithKits_LargeContainer)

Definition at line 58 of file LargeContainer.cpp.

8.30.3.16 void LargeContainer::sethasLargeContainer_SerialNumber (std::string _hasLargeContainer_SerialNumber)

Definition at line 52 of file LargeContainer.cpp.

8.30.3.17 void LargeContainer::sethasLargeContainer_SkuRef (std::string _hasLargeContainer_SkuRef)

Definition at line 49 of file LargeContainer.cpp.

8.30.4 Member Data Documentation

8.30.4.1 DAO* LargeContainer::dao [private]

Definition at line 33 of file LargeContainer.h.

8.30.4.2 LargeBoxWithEmptyKitTrays* LargeContainer::hasLargeBoxWithEmptyKitTrays_LargeContainer [private]

Definition at line 35 of file LargeContainer.h.

8.30.4.3 LargeBoxWithKits* LargeContainer::hasLargeBoxWithKits_LargeContainer [private]

Definition at line 34 of file LargeContainer.h.

8.30.4.4 std::string LargeContainer::hasLargeContainer_SerialNumber [private]

Definition at line 31 of file LargeContainer.h.

8.30.4.5 std::string LargeContainer::hasLargeContainer_SkuRef [private]

Definition at line 30 of file LargeContainer.h.

8.30.4.6 int LargeContainer::LargeContainerID [private]

Definition at line 32 of file LargeContainer.h.

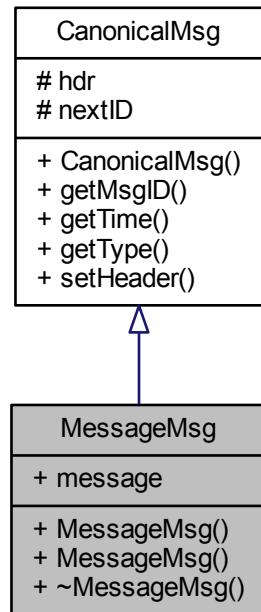
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[LargeContainer.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[LargeContainer.cpp](#)

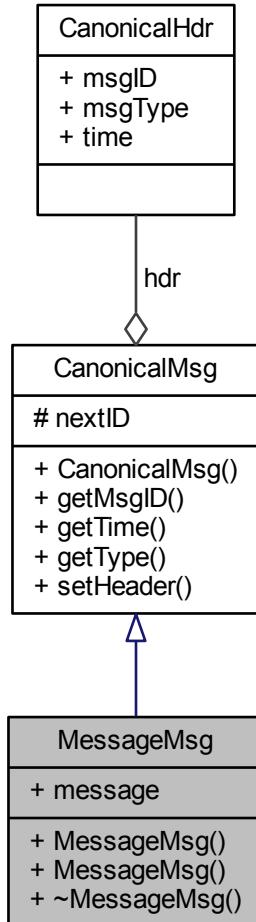
8.31 MessageMsg Class Reference

```
#include <canonicalMsg.hh>
```

Inheritance diagram for MessageMsg:



Collaboration diagram for MessageMsg:



Public Member Functions

- `MessageMsg ()`
- `MessageMsg (std::string messageIn)`
- `~MessageMsg ()`

Public Attributes

- `std::string message`

Additional Inherited Members

8.31.1 Detailed Description

Definition at line 93 of file canonicalMsg.hh.

8.31.2 Constructor & Destructor Documentation

8.31.2.1 MessageMsg::MessageMsg () [inline]

Definition at line 95 of file canonicalMsg.hh.

8.31.2.2 MessageMsg::MessageMsg (std::string *messageIn*) [inline]

Definition at line 96 of file canonicalMsg.hh.

8.31.2.3 MessageMsg::~MessageMsg () [inline]

Definition at line 97 of file canonicalMsg.hh.

8.31.3 Member Data Documentation

8.31.3.1 std::string MessageMsg::message

Definition at line 97 of file canonicalMsg.hh.

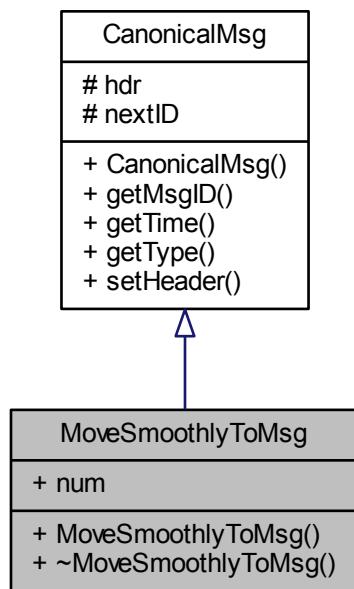
The documentation for this class was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[canonicalMsg.hh](#)

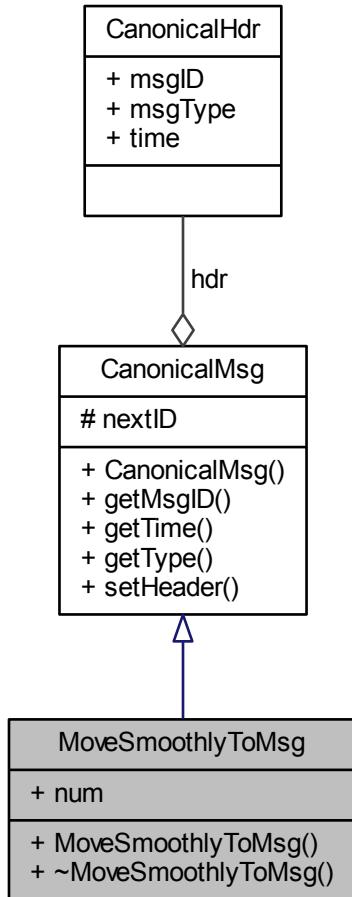
8.32 MoveSmoothlyToMsg Class Reference

```
#include <canonicalMsg.hh>
```

Inheritance diagram for MoveSmoothlyToMsg:



Collaboration diagram for MoveSmoothlyToMsg:



Public Member Functions

- `MoveSmoothlyToMsg ()`
- `~MoveSmoothlyToMsg ()`

Public Attributes

- int `num`

Additional Inherited Members

8.32.1 Detailed Description

Definition at line 101 of file canonicalMsg.hh.

8.32.2 Constructor & Destructor Documentation

8.32.2.1 MoveSmoothlyToMsg::MoveSmoothlyToMsg() [inline]

Definition at line 103 of file canonicalMsg.hh.

8.32.2.2 MoveSmoothlyToMsg::~MoveSmoothlyToMsg() [inline]

Definition at line 105 of file canonicalMsg.hh.

8.32.3 Member Data Documentation

8.32.3.1 int MoveSmoothlyToMsg::num

Definition at line 105 of file canonicalMsg.hh.

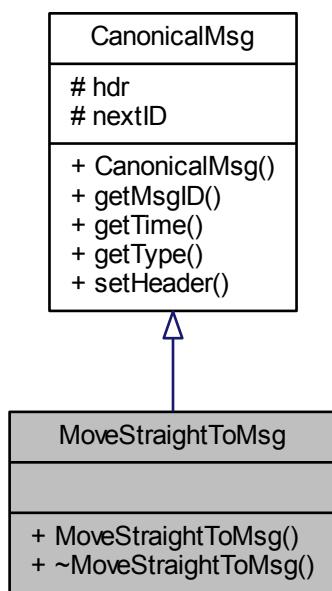
The documentation for this class was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[canonicalMsg.hh](#)

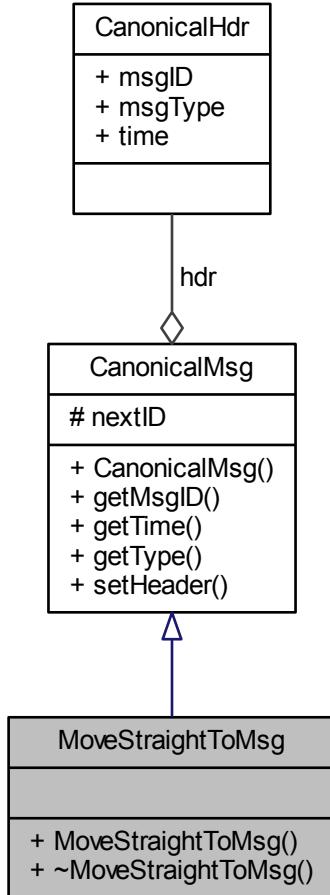
8.33 MoveStraightToMsg Class Reference

```
#include <canonicalMsg.hh>
```

Inheritance diagram for MoveStraightToMsg:



Collaboration diagram for MoveStraightToMsg:



Public Member Functions

- [MoveStraightToMsg \(\)](#)
- [~MoveStraightToMsg \(\)](#)

Additional Inherited Members

8.33.1 Detailed Description

Definition at line 110 of file canonicalMsg.hh.

8.33.2 Constructor & Destructor Documentation

8.33.2.1 MoveStraightToMsg::MoveStraightToMsg() [inline]

Definition at line 112 of file canonicalMsg.hh.

8.33.2.2 MoveStraightToMsg::~MoveStraightToMsg() [inline]

Definition at line 114 of file canonicalMsg.hh.

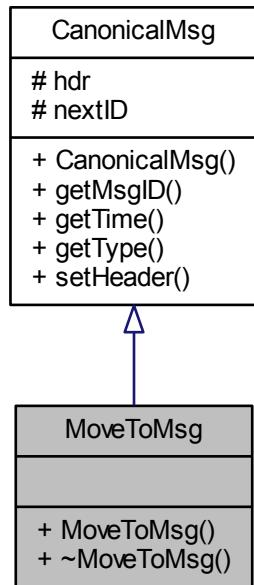
The documentation for this class was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[canonicalMsg.hh](#)

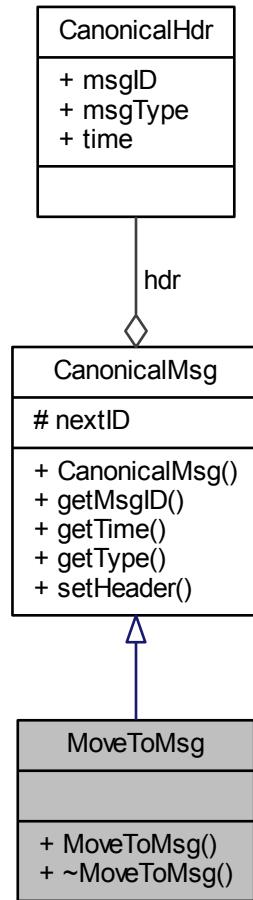
8.34 MoveToMsg Class Reference

```
#include <canonicalMsg.hh>
```

Inheritance diagram for MoveToMsg:



Collaboration diagram for MoveToMsg:



Public Member Functions

- [MoveToMsg \(\)](#)
- [~MoveToMsg \(\)](#)

Additional Inherited Members

8.34.1 Detailed Description

Definition at line 118 of file canonicalMsg.hh.

8.34.2 Constructor & Destructor Documentation

8.34.2.1 MoveToMsg::MoveToMsg () [inline]

Definition at line 120 of file canonicalMsg.hh.

8.34.2.2 MoveToMsg::~MoveToMsg () [inline]

Definition at line 122 of file canonicalMsg.hh.

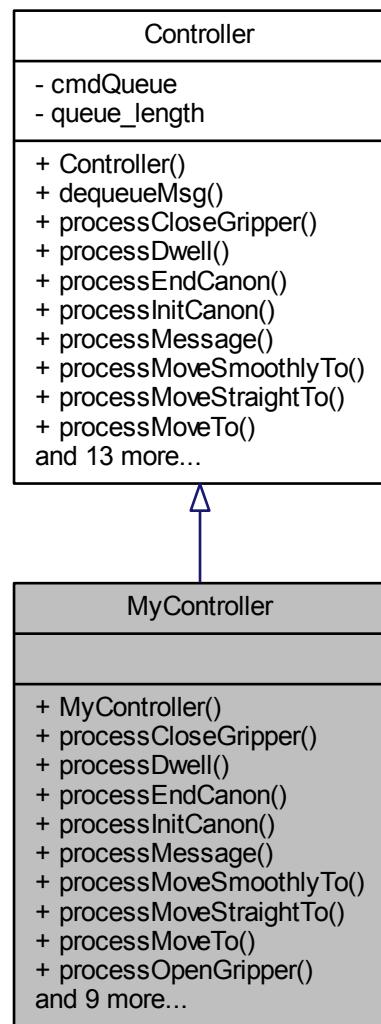
The documentation for this class was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[canonicalMsg.hh](#)

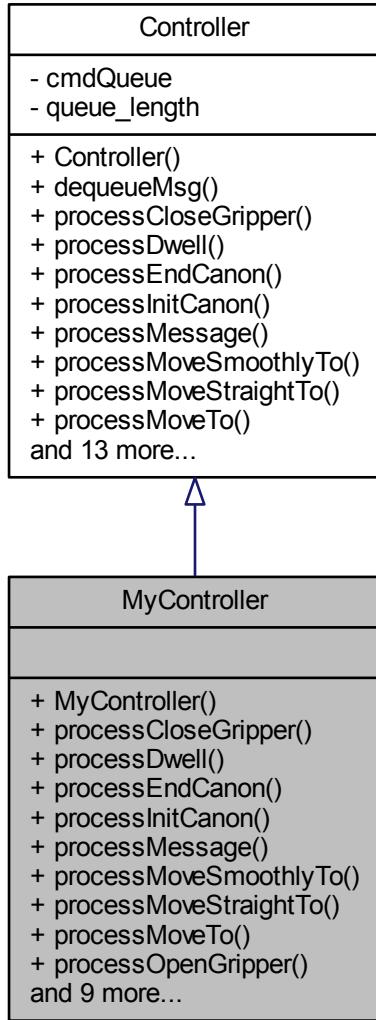
8.35 MyController Class Reference

```
#include <myController.hh>
```

Inheritance diagram for MyController:



Collaboration diagram for MyController:



Public Member Functions

- [MyController \(\)](#)
- int [processCloseGripper \(CloseGripperMsg *closeGripperMsg\)](#)
- int [processDwell \(DwellMsg *dwellMsg\)](#)
- int [processEndCanon \(EndCanonMsg *endCanonMsg\)](#)
- int [processInitCanon \(InitCanonMsg *initCanonMsg\)](#)
- int [processMessage \(MessageMsg *messageMsg\)](#)
- int [processMoveSmoothlyTo \(MoveSmoothlyToMsg *moveSmoothlyToMsg\)](#)
- int [processMoveStraightTo \(MoveStraightToMsg *moveStraightToMsg\)](#)

- int `processMoveTo (MoveToMsg *moveToMsg)`
- int `processOpenGripper (OpenGripperMsg *openGripperMsg)`
- int `processSetAbsoluteAcceleration (SetAbsoluteAccelerationMsg *setAbsoluteAccelerationMsg)`
- int `processSetAngleUnits (SetAngleUnitsMsg *setAngleUnitsMsg)`
- int `processSetEndAngleTolerance (SetEndAngleToleranceMsg *setEndAngleToleranceMsg)`
- int `processSetEndPointTolerance (SetEndPointToleranceMsg *setEndPointToleranceMsg)`
- int `processSetIntermediatePointTolerance (SetIntermediatePointToleranceMsg *setIntermediatePointToleranceMsg)`
- int `processSetLengthUnits (SetLengthUnitsMsg *setLengthUnitsMsg)`
- int `processSetRelativeAcceleration (SetRelativeAccelerationMsg *setRelativeAccelerationMsg)`
- int `processSetRelativeSpeed (SetRelativeSpeedMsg *setRelativeSpeedMsg)`
- `~MyController ()`

8.35.1 Detailed Description

Definition at line 23 of file myController.hh.

8.35.2 Constructor & Destructor Documentation

8.35.2.1 MyController::MyController () [inline]

Definition at line 25 of file myController.hh.

8.35.2.2 MyController::~MyController () [inline]

Definition at line 26 of file myController.hh.

8.35.3 Member Function Documentation

8.35.3.1 int MyController::processCloseGripper (CloseGripperMsg * closeGripperMsg) [virtual]

Implements [Controller](#).

Definition at line 23 of file myController.cpp.

8.35.3.2 int MyController::processDwell (DwellMsg * dwellMsg) [virtual]

Implements [Controller](#).

Definition at line 28 of file myController.cpp.

8.35.3.3 int MyController::processEndCanon (EndCanonMsg * endCanonMsg) [virtual]

Implements [Controller](#).

Definition at line 33 of file myController.cpp.

8.35.3.4 int MyController::processInitCanon (*InitCanonMsg * initCanonMsg*) [virtual]

Implements [Controller](#).

Definition at line 38 of file myController.cpp.

8.35.3.5 int MyController::processMessage (*MessageMsg * messageMsg*) [virtual]

Implements [Controller](#).

Definition at line 43 of file myController.cpp.

8.35.3.6 int MyController::processMoveSmoothlyTo (*MoveSmoothlyToMsg * moveSmoothlyToMsg*) [virtual]

Implements [Controller](#).

Definition at line 47 of file myController.cpp.

8.35.3.7 int MyController::processMoveStraightTo (*MoveStraightToMsg * moveStraightToMsg*) [virtual]

Implements [Controller](#).

Definition at line 51 of file myController.cpp.

8.35.3.8 int MyController::processMoveTo (*MoveToMsg * moveToMsg*) [virtual]

Implements [Controller](#).

Definition at line 55 of file myController.cpp.

8.35.3.9 int MyController::processOpenGripper (*OpenGripperMsg * openGripperMsg*) [virtual]

Implements [Controller](#).

Definition at line 59 of file myController.cpp.

8.35.3.10 int MyController::processSetAbsoluteAcceleration (*SetAbsoluteAccelerationMsg * setAbsoluteAccelerationMsg*) [virtual]

Implements [Controller](#).

Definition at line 64 of file myController.cpp.

8.35.3.11 int MyController::processSetAngleUnits (*SetAngleUnitsMsg * setAngleUnitsMsg*) [virtual]

Implements [Controller](#).

Definition at line 68 of file myController.cpp.

```
8.35.3.12 int MyController::processSetEndAngleTolerance ( SetEndAngleToleranceMsg * setEndAngleToleranceMsg )  
[virtual]
```

Implements [Controller](#).

Definition at line 72 of file myController.cpp.

```
8.35.3.13 int MyController::processSetEndPointTolerance ( SetEndPointToleranceMsg * setEndPointToleranceMsg )  
[virtual]
```

Implements [Controller](#).

Definition at line 76 of file myController.cpp.

```
8.35.3.14 int MyController::processSetIntermediatePointTolerance ( SetIntermediatePointToleranceMsg *  
setIntermediatePointToleranceMsg ) [virtual]
```

Implements [Controller](#).

Definition at line 80 of file myController.cpp.

```
8.35.3.15 int MyController::processSetLengthUnits ( SetLengthUnitsMsg * setLengthUnitsMsg ) [virtual]
```

Implements [Controller](#).

Definition at line 84 of file myController.cpp.

```
8.35.3.16 int MyController::processSetRelativeAcceleration ( SetRelativeAccelerationMsg * setRelativeAccelerationMsg )  
[virtual]
```

Implements [Controller](#).

Definition at line 88 of file myController.cpp.

```
8.35.3.17 int MyController::processSetRelativeSpeed ( SetRelativeSpeedMsg * setRelativeSpeedMsg ) [virtual]
```

Implements [Controller](#).

Definition at line 92 of file myController.cpp.

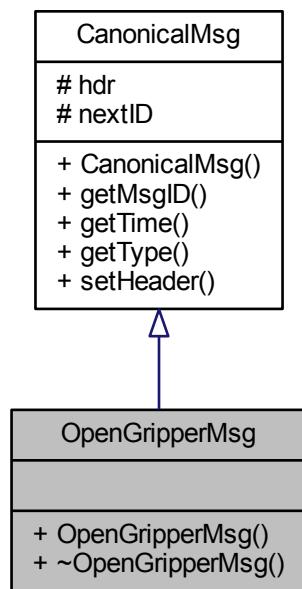
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[myController.hh](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[myController.cpp](#)

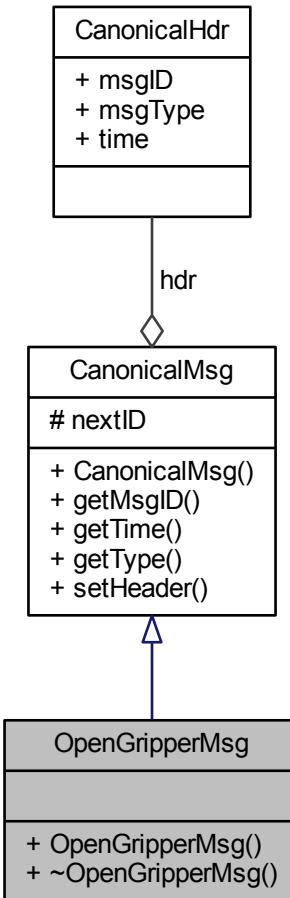
8.36 OpenGripperMsg Class Reference

```
#include <canonicalMsg.hh>
```

Inheritance diagram for OpenGripperMsg:



Collaboration diagram for OpenGripperMsg:



Public Member Functions

- [OpenGripperMsg \(\)](#)
- [~OpenGripperMsg \(\)](#)

Additional Inherited Members

8.36.1 Detailed Description

Definition at line 126 of file canonicalMsg.hh.

8.36.2 Constructor & Destructor Documentation

8.36.2.1 OpenGripperMsg::OpenGripperMsg() [inline]

Definition at line 128 of file canonicalMsg.hh.

8.36.2.2 OpenGripperMsg::~OpenGripperMsg() [inline]

Definition at line 129 of file canonicalMsg.hh.

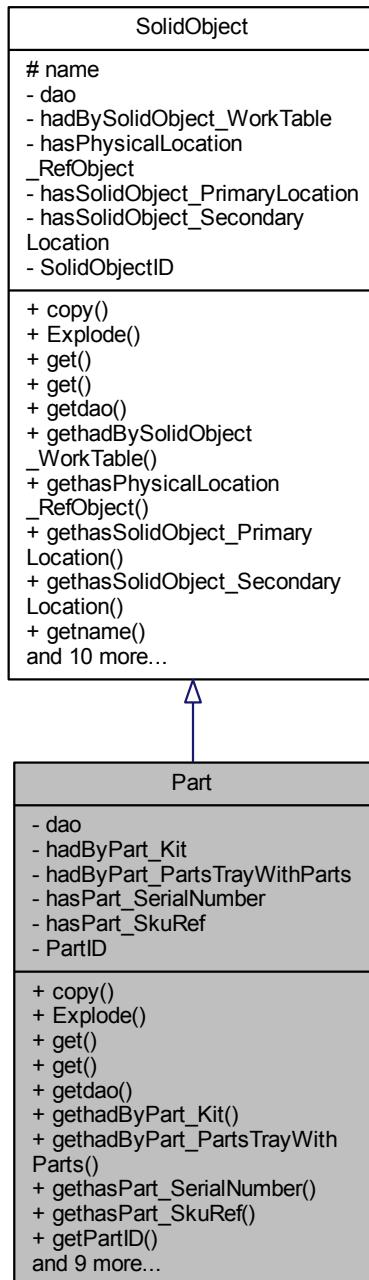
The documentation for this class was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[canonicalMsg.hh](#)

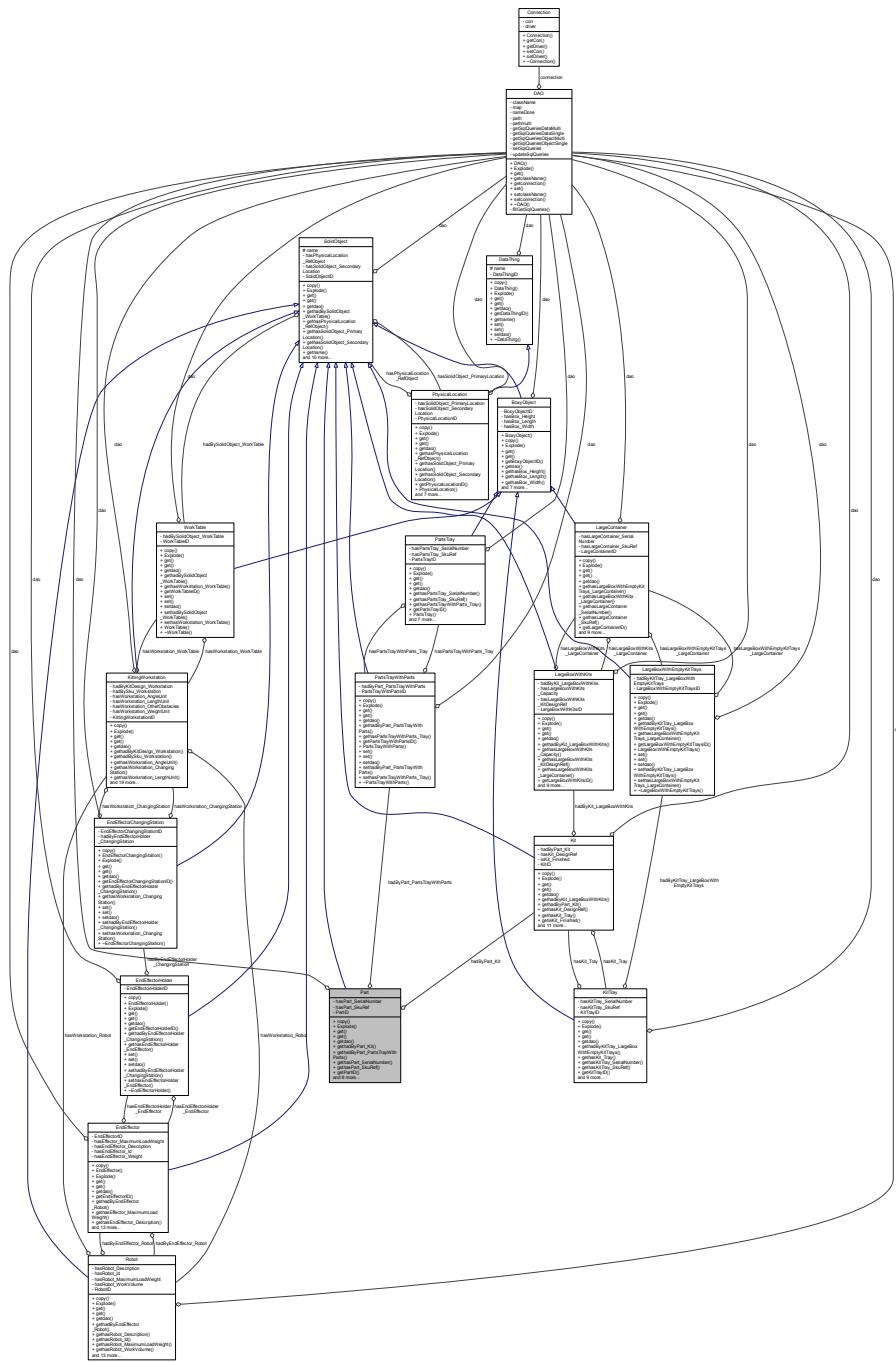
8.37 Part Class Reference

```
#include <Part.h>
```

Inheritance diagram for Part:



Collaboration diagram for Part:



Public Member Functions

- `void copy (std::map< std::string, std::string > object)`
- `std::vector< std::string > Explode (const std::string &str, char separator)`
- `void get (int id)`
- `void get (std::string name)`

- `DAO * getdao ()`
- `Kit * gethadByPart_Kit ()`
- `PartsTrayWithParts * gethadByPart_PartsTrayWithParts ()`
- `std::string gethasPart_SerialNumber ()`
- `std::string gethasPart_SkuRef ()`
- `int getPartID ()`
- `Part (std::string name)`
- `void set (int id, Part *obj)`
- `void set (std::string name)`
- `void setdao (DAO *_dao)`
- `void sethadByPart_Kit (Kit *_hadByPart_Kit)`
- `void sethadByPart_PartsTrayWithParts (PartsTrayWithParts *_hadByPart_PartsTrayWithParts)`
- `void sethasPart_SerialNumber (std::string _hasPart_SerialNumber)`
- `void sethasPart_SkuRef (std::string _hasPart_SkuRef)`
- `~Part ()`

Private Attributes

- `DAO * dao`
- `Kit * hadByPart_Kit`
- `PartsTrayWithParts * hadByPart_PartsTrayWithParts`
- `std::string hasPart_SerialNumber`
- `std::string hasPart_SkuRef`
- `int PartID`

Additional Inherited Members

8.37.1 Detailed Description

Definition at line 29 of file Part.h.

8.37.2 Constructor & Destructor Documentation

8.37.2.1 Part::Part (std::string name)

Definition at line 21 of file Part.cpp.

8.37.2.2 Part::~Part ()

Definition at line 26 of file Part.cpp.

8.37.3 Member Function Documentation

8.37.3.1 void Part::copy (std::map< std::string, std::string > object)

Definition at line 94 of file Part.cpp.

Here is the caller graph for this function:

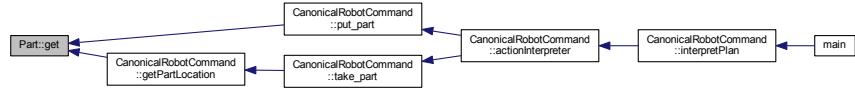


8.37.3.2 std::vector< std::string > Part::Explode (const std::string & str, char separator)

Definition at line 111 of file Part.cpp.

8.37.3.3 void Part::get (int id)

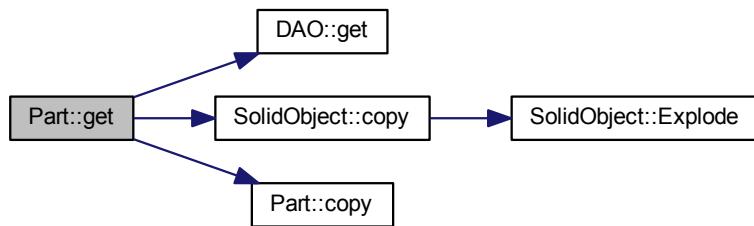
Here is the caller graph for this function:



8.37.3.4 void Part::get (std::string name)

Definition at line 64 of file Part.cpp.

Here is the call graph for this function:



8.37.3.5 DAO * Part::getdao()

Definition at line 40 of file Part.cpp.

8.37.3.6 Kit * Part::getadByPart_Kit()

Definition at line 43 of file Part.cpp.

8.37.3.7 PartsTrayWithParts * Part::getadByPart_PartsTrayWithParts()

Definition at line 46 of file Part.cpp.

8.37.3.8 std::string Part::gethasPart_SerialNumber()

Definition at line 34 of file Part.cpp.

8.37.3.9 std::string Part::gethasPart_SkuRef()

Definition at line 31 of file Part.cpp.

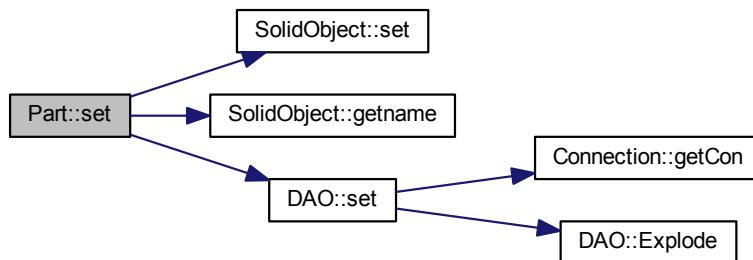
8.37.3.10 int Part::getPartID()

Definition at line 37 of file Part.cpp.

8.37.3.11 void Part::set(int *id*, Part * *obj*)**8.37.3.12 void Part::set(std::string *name*)**

Definition at line 74 of file Part.cpp.

Here is the call graph for this function:



8.37.3.13 void Part::setdao (DAO * *_dao*)

Definition at line 55 of file Part.cpp.

8.37.3.14 void Part::sethadByPart_Kit (Kit * *_hadByPart_Kit*)

Definition at line 58 of file Part.cpp.

8.37.3.15 void Part::sethadByPart_PartsTrayWithParts (PartsTrayWithParts * *_hadByPart_PartsTrayWithParts*)

Definition at line 61 of file Part.cpp.

8.37.3.16 void Part::sethasPart_SerialNumber (std::string *_hasPart_SerialNumber*)

Definition at line 52 of file Part.cpp.

8.37.3.17 void Part::sethasPart_SkuRef (std::string *_hasPart_SkuRef*)

Definition at line 49 of file Part.cpp.

8.37.4 Member Data Documentation

8.37.4.1 DAO* Part::*dao* [private]

Definition at line 33 of file Part.h.

8.37.4.2 Kit* Part::*hadByPart_Kit* [private]

Definition at line 34 of file Part.h.

8.37.4.3 PartsTrayWithParts* Part::*hadByPart_PartsTrayWithParts* [private]

Definition at line 35 of file Part.h.

8.37.4.4 std::string Part::*hasPart_SerialNumber* [private]

Definition at line 31 of file Part.h.

8.37.4.5 std::string Part::*hasPart_SkuRef* [private]

Definition at line 30 of file Part.h.

8.37.4.6 int Part::PartID [private]

Definition at line 32 of file Part.h.

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

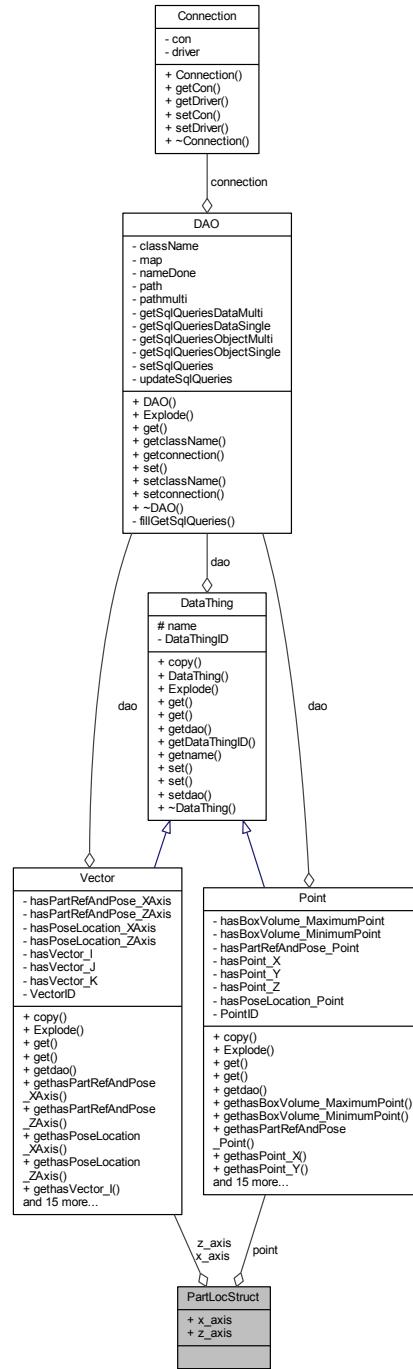
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[Part.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[Part.cpp](#)

8.38 PartLocStruct Struct Reference

Represents a part location.

```
#include <Structdef.h>
```

Collaboration diagram for PartLocStruct:



Public Attributes

- `Point * point`
- `Vector * x_axis`

- `Vector * z_axis`

8.38.1 Detailed Description

Represents a part location.

Definition at line 32 of file Structdef.h.

8.38.2 Member Data Documentation

8.38.2.1 PartLocStruct::point

Member *point* contains the coordinates of the part

Definition at line 34 of file Structdef.h.

8.38.2.2 PartLocStruct::x_axis

Member *x_axis* represents the X axis for the part

Definition at line 35 of file Structdef.h.

8.38.2.3 PartLocStruct::z_axis

Member *z_axis* represents the Z axis for the part

Definition at line 36 of file Structdef.h.

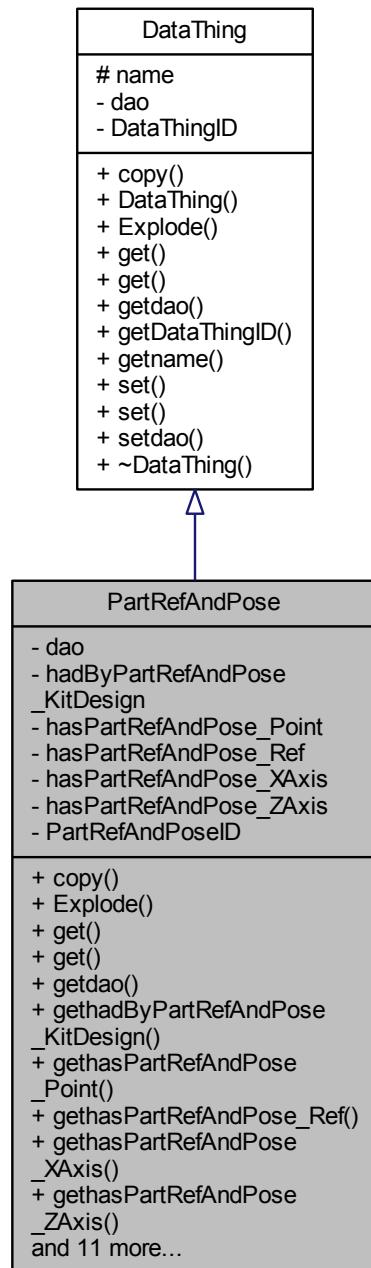
The documentation for this struct was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/[Structdef.h](#)

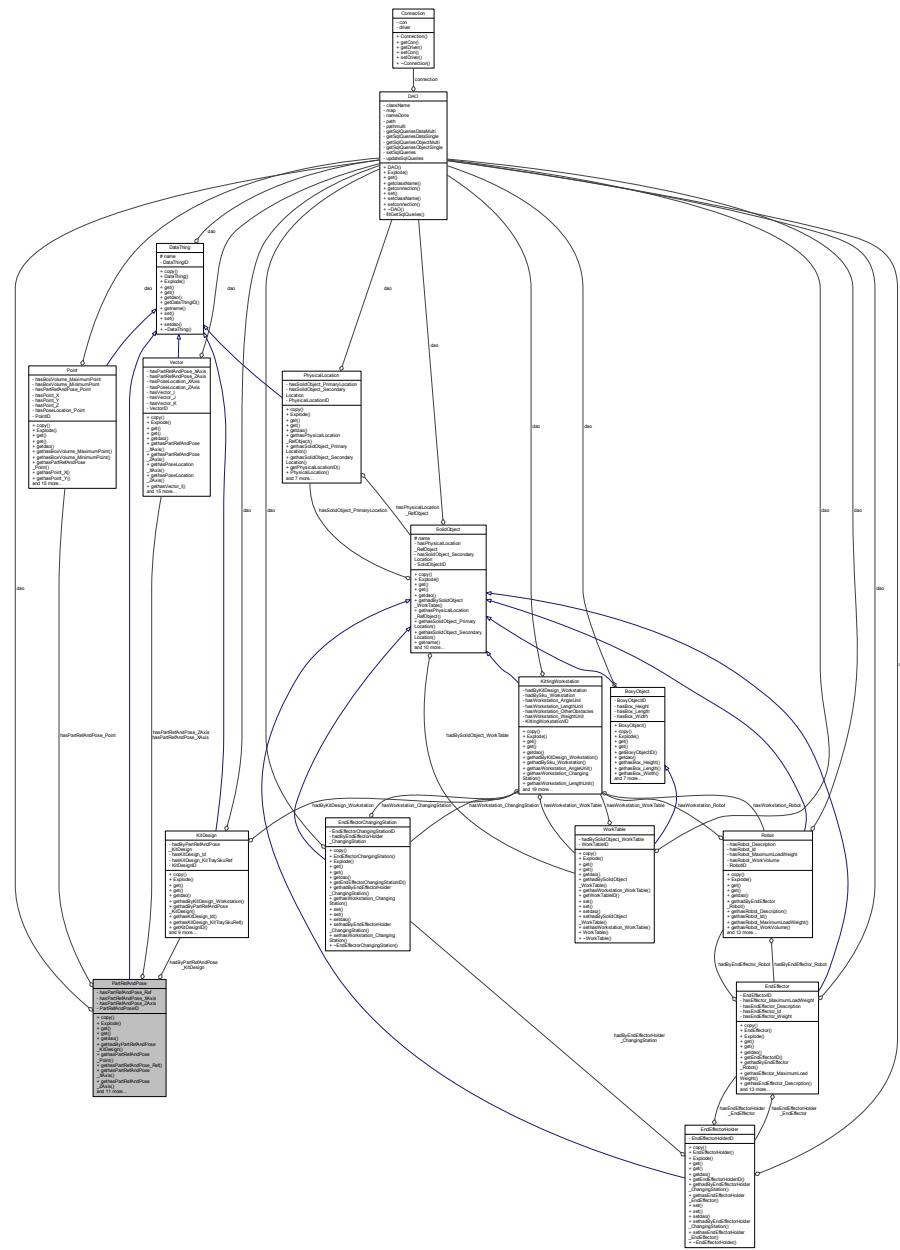
8.39 PartRefAndPose Class Reference

```
#include <PartRefAndPose.h>
```

Inheritance diagram for PartRefAndPose:



Collaboration diagram for PartRefAndPose:



Public Member Functions

- void [copy](#) (std::map< std::string, std::string > object)
 - std::vector< std::string > [Explode](#) (const std::string &str, char separator)
 - void [get](#) (int id)
 - void [get](#) (std::string name)
 - DAO * [getdao](#) ()
 - KitDesign * [gethadByPartRefAndPose_KitDesign](#) ()
 - Point * [gethasPartRefAndPose_Point](#) ()

- std::string `gethasPartRefAndPose_Ref ()`
- `Vector * gethasPartRefAndPose_XAxis ()`
- `Vector * gethasPartRefAndPose_ZAxis ()`
- int `getPartRefAndPoseID ()`
- `PartRefAndPose (std::string name)`
- void `set (int id, PartRefAndPose *obj)`
- void `set (std::string name)`
- void `setdao (DAO *_dao)`
- void `sethadByPartRefAndPose_KitDesign (KitDesign *_hadByPartRefAndPose_KitDesign)`
- void `sethasPartRefAndPose_Point (Point *_hasPartRefAndPose_Point)`
- void `sethasPartRefAndPose_Ref (std::string _hasPartRefAndPose_Ref)`
- void `sethasPartRefAndPose_XAxis (Vector *_hasPartRefAndPose_XAxis)`
- void `sethasPartRefAndPose_ZAxis (Vector *_hasPartRefAndPose_ZAxis)`
- `~PartRefAndPose ()`

Private Attributes

- DAO * `dao`
- KitDesign * `hadByPartRefAndPose_KitDesign`
- Point * `hasPartRefAndPose_Point`
- std::string `hasPartRefAndPose_Ref`
- Vector * `hasPartRefAndPose_XAxis`
- Vector * `hasPartRefAndPose_ZAxis`
- int `PartRefAndPoseID`

Additional Inherited Members

8.39.1 Detailed Description

Definition at line 30 of file PartRefAndPose.h.

8.39.2 Constructor & Destructor Documentation

8.39.2.1 PartRefAndPose::PartRefAndPose (std::string *name*)

Definition at line 22 of file PartRefAndPose.cpp.

8.39.2.2 PartRefAndPose::~PartRefAndPose ()

Definition at line 29 of file PartRefAndPose.cpp.

8.39.3 Member Function Documentation

8.39.3.1 void PartRefAndPose::copy (std::map< std::string, std::string > *object*)

Definition at line 108 of file PartRefAndPose.cpp.

Here is the caller graph for this function:

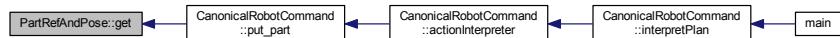


8.39.3.2 std::vector< std::string > PartRefAndPose::Explode (const std::string & str, char separator)

Definition at line 130 of file PartRefAndPose.cpp.

8.39.3.3 void PartRefAndPose::get (int id)

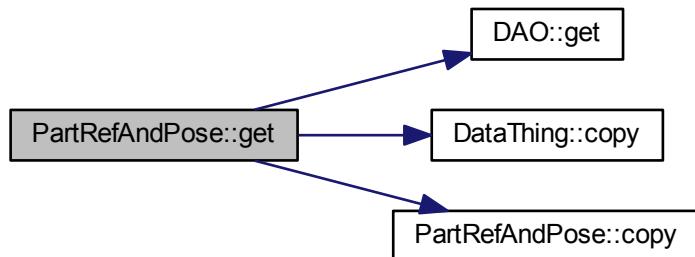
Here is the caller graph for this function:



8.39.3.4 void PartRefAndPose::get (std::string name)

Definition at line 75 of file PartRefAndPose.cpp.

Here is the call graph for this function:



8.39.3.5 DAO * PartRefAndPose::getdao()

Definition at line 42 of file PartRefAndPose.cpp.

8.39.3.6 KitDesign * PartRefAndPose::gethadByPartRefAndPose_KitDesign()

Definition at line 48 of file PartRefAndPose.cpp.

8.39.3.7 Point * PartRefAndPose::gethasPartRefAndPose_Point()

Definition at line 51 of file PartRefAndPose.cpp.

Here is the caller graph for this function:



8.39.3.8 std::string PartRefAndPose::gethasPartRefAndPose_Ref()

Definition at line 36 of file PartRefAndPose.cpp.

8.39.3.9 Vector * PartRefAndPose::gethasPartRefAndPose_XAxis()

Definition at line 54 of file PartRefAndPose.cpp.

Here is the caller graph for this function:



8.39.3.10 Vector * PartRefAndPose::gethasPartRefAndPose_ZAxis()

Definition at line 45 of file PartRefAndPose.cpp.

Here is the caller graph for this function:



8.39.3.11 int PartRefAndPose::getPartRefAndPoseID()

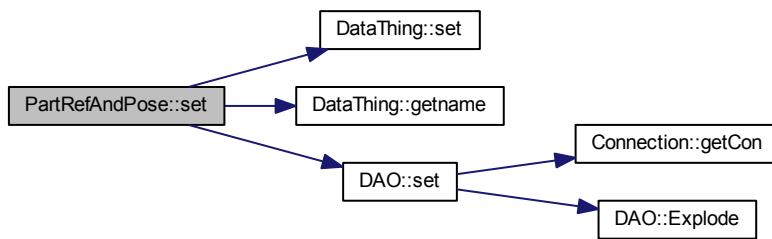
Definition at line 39 of file PartRefAndPose.cpp.

8.39.3.12 void PartRefAndPose::set(int *id*, PartRefAndPose * *obj*)

8.39.3.13 void PartRefAndPose::set(std::string *name*)

Definition at line 85 of file PartRefAndPose.cpp.

Here is the call graph for this function:



8.39.3.14 void PartRefAndPose::setdao(DAO * *_dao*)

Definition at line 60 of file PartRefAndPose.cpp.

8.39.3.15 void PartRefAndPose::sethadByPartRefAndPose_KitDesign(KitDesign * *_hadByPartRefAndPose_KitDesign*)

Definition at line 66 of file PartRefAndPose.cpp.

8.39.3.16 void PartRefAndPose::sethasPartRefAndPose_Point(Point * *_hasPartRefAndPose_Point*)

Definition at line 69 of file PartRefAndPose.cpp.

8.39.3.17 void PartRefAndPose::sethasPartRefAndPose_Ref(std::string *_hasPartRefAndPose_Ref*)

Definition at line 57 of file PartRefAndPose.cpp.

8.39.3.18 void PartRefAndPose::sethasPartRefAndPose_XAxis(Vector * *_hasPartRefAndPose_XAxis*)

Definition at line 72 of file PartRefAndPose.cpp.

8.39.3.19 void PartRefAndPose::sethasPartRefAndPose_ZAxis (`Vector * _hasPartRefAndPose_ZAxis`)

Definition at line 63 of file PartRefAndPose.cpp.

8.39.4 Member Data Documentation

8.39.4.1 DAO* PartRefAndPose::dao [private]

Definition at line 33 of file PartRefAndPose.h.

8.39.4.2 KitDesign* PartRefAndPose::hadByPartRefAndPose_KitDesign [private]

Definition at line 35 of file PartRefAndPose.h.

8.39.4.3 Point* PartRefAndPose::hasPartRefAndPose_Point [private]

Definition at line 36 of file PartRefAndPose.h.

8.39.4.4 std::string PartRefAndPose::hasPartRefAndPose_Ref [private]

Definition at line 31 of file PartRefAndPose.h.

8.39.4.5 Vector* PartRefAndPose::hasPartRefAndPose_XAxis [private]

Definition at line 37 of file PartRefAndPose.h.

8.39.4.6 Vector* PartRefAndPose::hasPartRefAndPose_ZAxis [private]

Definition at line 34 of file PartRefAndPose.h.

8.39.4.7 int PartRefAndPose::PartRefAndPoseID [private]

Definition at line 32 of file PartRefAndPose.h.

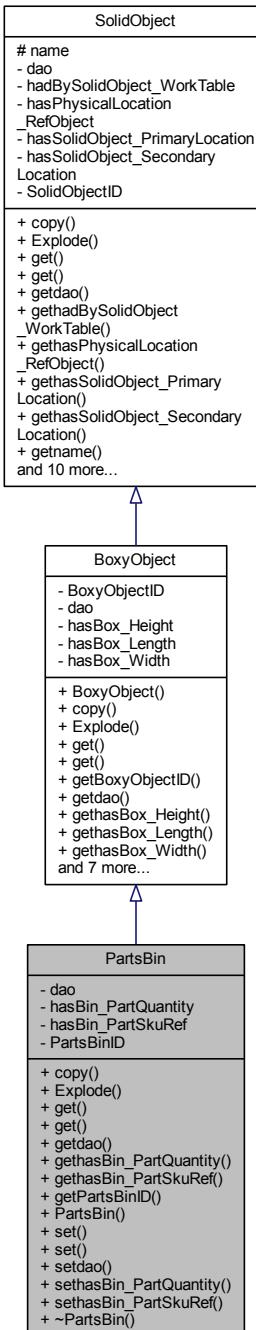
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[PartRefAndPose.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[PartRefAndPose.cpp](#)

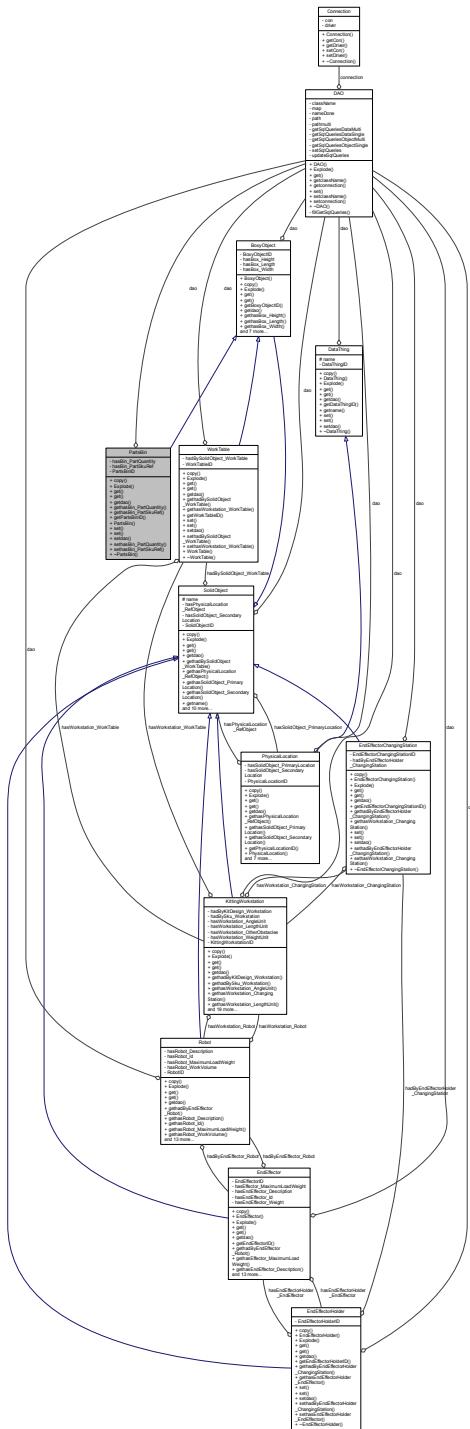
8.40 PartsBin Class Reference

```
#include <PartsBin.h>
```

Inheritance diagram for PartsBin:



Collaboration diagram for PartsBin:



Public Member Functions

- void **copy** (std::map< std::string, std::string > object)
 - std::vector< std::string > **Explode** (const std::string &str, char separator)

- void [get](#) (int id)
- void [get](#) (std::string name)
- DAO * [getdao](#) ()
- std::string [gethasBin_PartQuantity](#) ()
- std::string [gethasBin_PartSkuRef](#) ()
- int [getPartsBinID](#) ()
- PartsBin (std::string name)
- void [set](#) (int id, PartsBin *obj)
- void [set](#) (std::string name)
- void [setdao](#) (DAO *_dao)
- void [sethasBin_PartQuantity](#) (std::string _hasBin_PartQuantity)
- void [sethasBin_PartSkuRef](#) (std::string _hasBin_PartSkuRef)
- ~PartsBin ()

Private Attributes

- DAO * dao
- std::string hasBin_PartQuantity
- std::string hasBin_PartSkuRef
- int PartsBinID

Additional Inherited Members

8.40.1 Detailed Description

Definition at line 27 of file PartsBin.h.

8.40.2 Constructor & Destructor Documentation

8.40.2.1 PartsBin::PartsBin (std::string name)

Definition at line 19 of file PartsBin.cpp.

8.40.2.2 PartsBin::~PartsBin ()

Definition at line 22 of file PartsBin.cpp.

8.40.3 Member Function Documentation

8.40.3.1 void PartsBin::copy (std::map< std::string, std::string > object)

Definition at line 77 of file PartsBin.cpp.

Here is the caller graph for this function:



8.40.3.2 std::vector< std::string > PartsBin::Explode (const std::string & str, char separator)

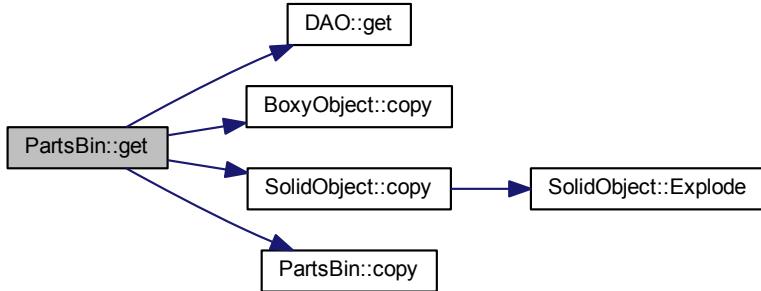
Definition at line 88 of file PartsBin.cpp.

8.40.3.3 void PartsBin::get (int id)

8.40.3.4 void PartsBin::get (std::string name)

Definition at line 46 of file PartsBin.cpp.

Here is the call graph for this function:



8.40.3.5 DAO * PartsBin::getdao ()

Definition at line 34 of file PartsBin.cpp.

8.40.3.6 std::string PartsBin::gethasBin_PartQuantity ()

Definition at line 25 of file PartsBin.cpp.

8.40.3.7 `std::string PartsBin::getHasBin_PartSkuRef()`

Definition at line 28 of file PartsBin.cpp.

8.40.3.8 `int PartsBin::getPartsBinID()`

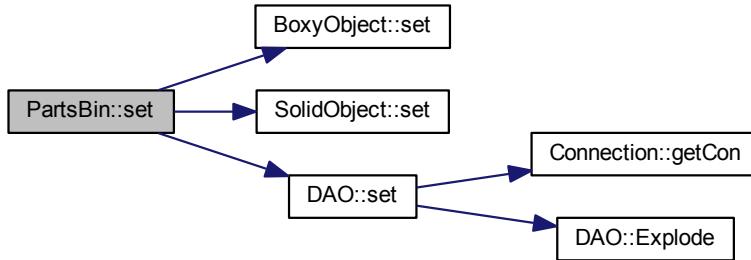
Definition at line 31 of file PartsBin.cpp.

8.40.3.9 `void PartsBin::set(int id, PartsBin * obj)`

8.40.3.10 `void PartsBin::set(std::string name)`

Definition at line 59 of file PartsBin.cpp.

Here is the call graph for this function:



8.40.3.11 `void PartsBin::setdao(DAO * _dao)`

Definition at line 43 of file PartsBin.cpp.

8.40.3.12 `void PartsBin::setHasBin_PartQuantity(std::string _hasBin_PartQuantity)`

Definition at line 37 of file PartsBin.cpp.

8.40.3.13 `void PartsBin::setHasBin_PartSkuRef(std::string _hasBin_PartSkuRef)`

Definition at line 40 of file PartsBin.cpp.

8.40.4 Member Data Documentation

8.40.4.1 `DAO* PartsBin::dao [private]`

Definition at line 31 of file PartsBin.h.

8.40.4.2 std::string PartsBin::hasBin_PartQuantity [private]

Definition at line 28 of file PartsBin.h.

8.40.4.3 std::string PartsBin::hasBin_PartSkuRef [private]

Definition at line 29 of file PartsBin.h.

8.40.4.4 int PartsBin::PartsBinID [private]

Definition at line 30 of file PartsBin.h.

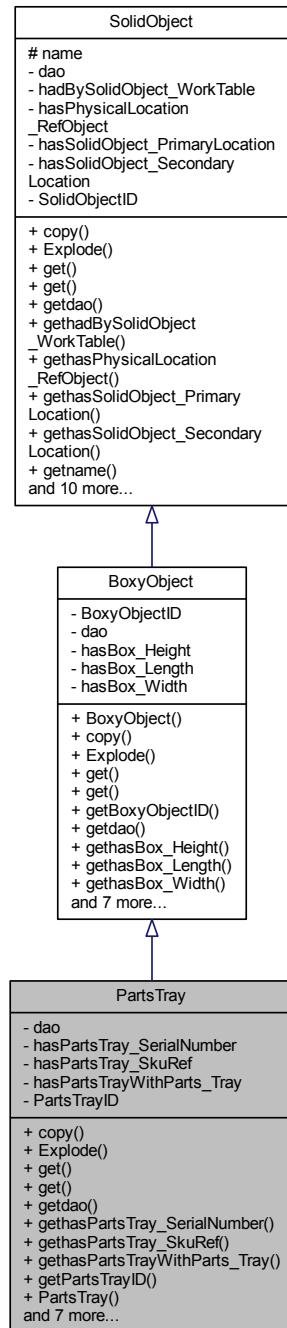
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[PartsBin.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[PartsBin.cpp](#)

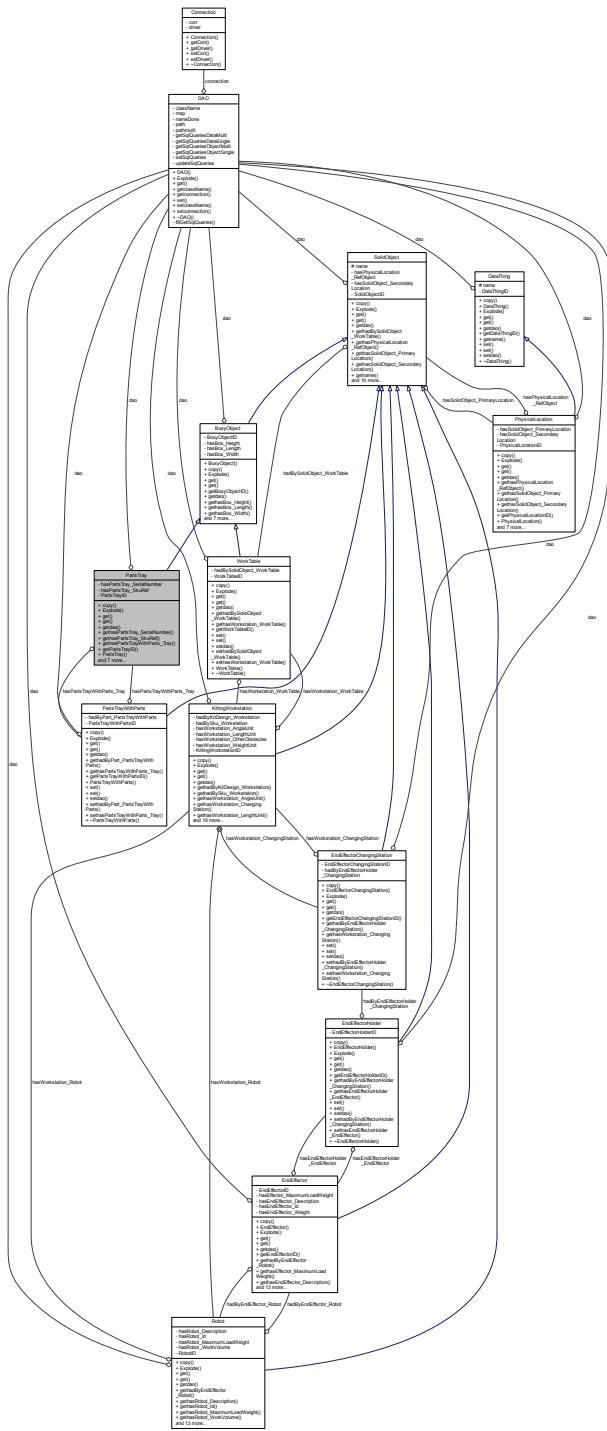
8.41 PartsTray Class Reference

```
#include <PartsTray.h>
```

Inheritance diagram for PartsTray:



Collaboration diagram for PartsTray:



Public Member Functions

- void **copy** (std::map< std::string, std::string > object)
 - std::vector< std::string > **Explode** (const std::string &str, char separator)

- void `get` (int id)
- void `get` (std::string name)
- DAO * `getdao` ()
- std::string `gethasPartsTray_SerialNumber` ()
- std::string `gethasPartsTray_SkuRef` ()
- PartsTrayWithParts * `gethasPartsTrayWithParts_Tray` ()
- int `getPartsTrayID` ()
- PartsTray (std::string name)
- void `set` (int id, PartsTray *obj)
- void `set` (std::string name)
- void `setdao` (DAO *_dao)
- void `sethasPartsTray_SerialNumber` (std::string _hasPartsTray_SerialNumber)
- void `sethasPartsTray_SkuRef` (std::string _hasPartsTray_SkuRef)
- void `sethasPartsTrayWithParts_Tray` (PartsTrayWithParts *_hasPartsTrayWithParts_Tray)
- ~PartsTray ()

Private Attributes

- DAO * dao
- std::string hasPartsTray_SerialNumber
- std::string hasPartsTray_SkuRef
- PartsTrayWithParts * hasPartsTrayWithParts_Tray
- int PartsTrayID

Additional Inherited Members

8.41.1 Detailed Description

Definition at line 28 of file PartsTray.h.

8.41.2 Constructor & Destructor Documentation

8.41.2.1 PartsTray::PartsTray (std::string name)

Definition at line 20 of file PartsTray.cpp.

8.41.2.2 PartsTray::~PartsTray ()

Definition at line 24 of file PartsTray.cpp.

8.41.3 Member Function Documentation

8.41.3.1 void PartsTray::copy (std::map< std::string, std::string > object)

Definition at line 88 of file PartsTray.cpp.

Here is the caller graph for this function:



8.41.3.2 std::vector< std::string > PartsTray::Explode (const std::string & str, char separator)

Definition at line 102 of file PartsTray.cpp.

8.41.3.3 void PartsTray::get (int id)

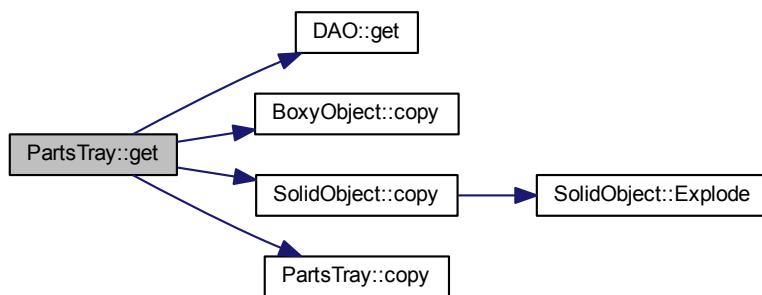
Here is the caller graph for this function:



8.41.3.4 void PartsTray::get (std::string name)

Definition at line 55 of file PartsTray.cpp.

Here is the call graph for this function:



8.41.3.5 DAO * PartsTray::getdao ()

Definition at line 37 of file PartsTray.cpp.

8.41.3.6 std::string PartsTray::gethasPartsTray_SerialNumber ()

Definition at line 31 of file PartsTray.cpp.

8.41.3.7 std::string PartsTray::gethasPartsTray_SkuRef ()

Definition at line 28 of file PartsTray.cpp.

8.41.3.8 PartsTrayWithParts * PartsTray::gethasPartsTrayWithParts_Tray ()

Definition at line 40 of file PartsTray.cpp.

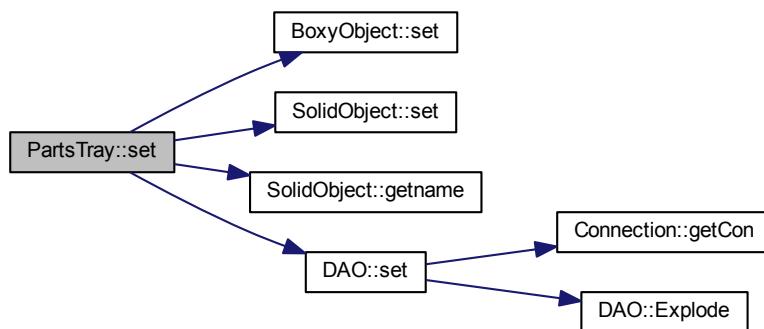
8.41.3.9 int PartsTray::getPartsTrayID ()

Definition at line 34 of file PartsTray.cpp.

8.41.3.10 void PartsTray::set (int id, PartsTray * obj)**8.41.3.11 void PartsTray::set (std::string name)**

Definition at line 68 of file PartsTray.cpp.

Here is the call graph for this function:

**8.41.3.12 void PartsTray::setdao (DAO * _dao)**

Definition at line 49 of file PartsTray.cpp.

8.41.3.13 void PartsTray::sethasPartsTray_SerialNumber (std::string *_hasPartsTray_SerialNumber*)

Definition at line 46 of file PartsTray.cpp.

8.41.3.14 void PartsTray::sethasPartsTray_SkuRef (std::string *_hasPartsTray_SkuRef*)

Definition at line 43 of file PartsTray.cpp.

8.41.3.15 void PartsTray::sethasPartsTrayWithParts_Tray (PartsTrayWithParts * *_hasPartsTrayWithParts_Tray*)

Definition at line 52 of file PartsTray.cpp.

8.41.4 Member Data Documentation

8.41.4.1 DAO* PartsTray::dao [private]

Definition at line 32 of file PartsTray.h.

8.41.4.2 std::string PartsTray::hasPartsTray_SerialNumber [private]

Definition at line 30 of file PartsTray.h.

8.41.4.3 std::string PartsTray::hasPartsTray_SkuRef [private]

Definition at line 29 of file PartsTray.h.

8.41.4.4 PartsTrayWithParts* PartsTray::hasPartsTrayWithParts_Tray [private]

Definition at line 33 of file PartsTray.h.

8.41.4.5 int PartsTray::PartsTrayID [private]

Definition at line 31 of file PartsTray.h.

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

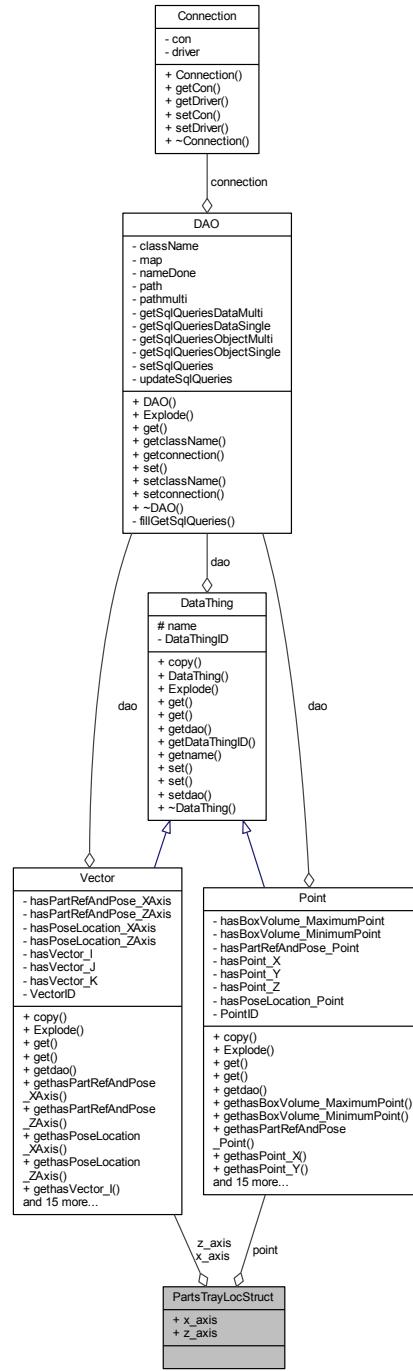
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[PartsTray.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[PartsTray.cpp](#)

8.42 PartsTrayLocStruct Struct Reference

This structure represents a parts tray location.

```
#include <Structdef.h>
```

Collaboration diagram for PartsTrayLocStruct:



Public Attributes

- `Point * point`
- `Vector * x_axis`

- `Vector * z_axis`

8.42.1 Detailed Description

This structure represents a parts tray location.

Definition at line 50 of file Structdef.h.

8.42.2 Member Data Documentation

8.42.2.1 PartsTrayLocStruct::point

Member *point* contains the coordinates of the parts tray

Definition at line 52 of file Structdef.h.

8.42.2.2 PartsTrayLocStruct::x_axis

Member *x_axis* represents the X axis for the parts tray

Definition at line 53 of file Structdef.h.

8.42.2.3 PartsTrayLocStruct::z_axis

Member *z_axis* represents the Z axis for the parts tray

Definition at line 54 of file Structdef.h.

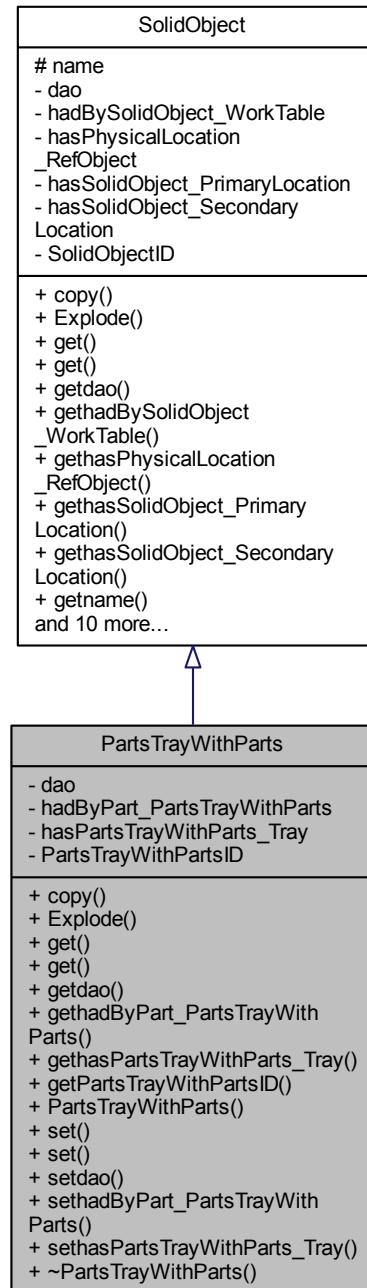
The documentation for this struct was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/[Structdef.h](#)

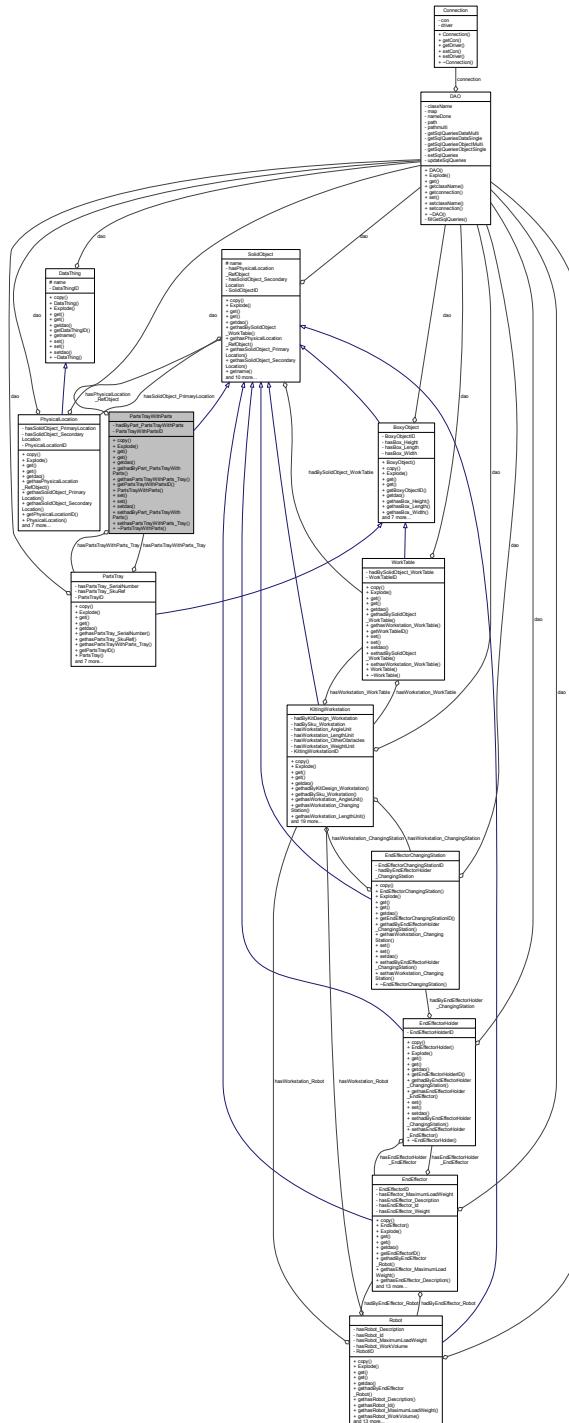
8.43 PartsTrayWithParts Class Reference

```
#include <PartsTrayWithParts.h>
```

Inheritance diagram for PartsTrayWithParts:



Collaboration diagram for PartsTrayWithParts:



Public Member Functions

- void **copy** (std::map< std::string, std::string > object)
 - std::vector< std::string > **Explode** (const std::string &str, char separator)

- void [get](#) (int id)
- void [get](#) (std::string name)
- DAO * [getdao](#) ()
- std::vector< Part * > [gethadByPart_PartsTrayWithParts](#) ()
- PartsTray * [gethasPartsTrayWithParts_Tray](#) ()
- int [getPartsTrayWithPartsID](#) ()
- PartsTrayWithParts ([std::string name](#))
- void [set](#) (int id, PartsTrayWithParts *obj)
- void [set](#) (std::string name)
- void [setdao](#) (DAO *_dao)
- void [sethadByPart_PartsTrayWithParts](#) (std::vector< Part * > _hadByPart_PartsTrayWithParts)
- void [sethasPartsTrayWithParts_Tray](#) (PartsTray * _hasPartsTrayWithParts_Tray)
- ~[PartsTrayWithParts](#) ()

Private Attributes

- DAO * [dao](#)
- std::vector< Part * > [hadByPart_PartsTrayWithParts](#)
- PartsTray * [hasPartsTrayWithParts_Tray](#)
- int [PartsTrayWithPartsID](#)

Additional Inherited Members

8.43.1 Detailed Description

Definition at line 29 of file PartsTrayWithParts.h.

8.43.2 Constructor & Destructor Documentation

8.43.2.1 PartsTrayWithParts::PartsTrayWithParts (std::string name)

Definition at line 21 of file PartsTrayWithParts.cpp.

8.43.2.2 PartsTrayWithParts::~PartsTrayWithParts ()

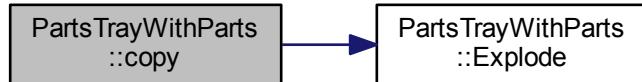
Definition at line 25 of file PartsTrayWithParts.cpp.

8.43.3 Member Function Documentation

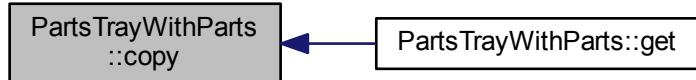
8.43.3.1 void PartsTrayWithParts::copy (std::map< std::string, std::string > object)

Definition at line 84 of file PartsTrayWithParts.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



8.43.3.2 `std::vector< std::string > PartsTrayWithParts::Explode (const std::string & str, char separator)`

Definition at line 102 of file PartsTrayWithParts.cpp.

Here is the caller graph for this function:

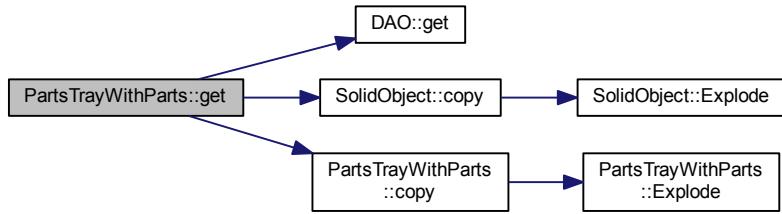


8.43.3.3 `void PartsTrayWithParts::get (int id)`

8.43.3.4 `void PartsTrayWithParts::get (std::string name)`

Definition at line 52 of file PartsTrayWithParts.cpp.

Here is the call graph for this function:



8.43.3.5 DAO * PartsTrayWithParts::getdao ()

Definition at line 34 of file `PartsTrayWithParts.cpp`.

8.43.3.6 std::vector< Part * > PartsTrayWithParts::gethadByPart_PartsTrayWithParts ()

Definition at line 40 of file `PartsTrayWithParts.cpp`.

8.43.3.7 PartsTray * PartsTrayWithParts::gethasPartsTrayWithParts_Tray ()

Definition at line 37 of file `PartsTrayWithParts.cpp`.

8.43.3.8 int PartsTrayWithParts::getPartsTrayWithPartsID ()

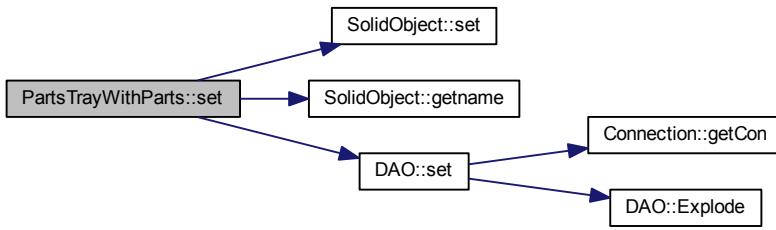
Definition at line 31 of file `PartsTrayWithParts.cpp`.

8.43.3.9 void PartsTrayWithParts::set (int *id*, PartsTrayWithParts * *obj*)

8.43.3.10 void PartsTrayWithParts::set (std::string *name*)

Definition at line 62 of file `PartsTrayWithParts.cpp`.

Here is the call graph for this function:



8.43.3.11 void PartsTrayWithParts::setdao (DAO * _dao)

Definition at line 43 of file PartsTrayWithParts.cpp.

8.43.3.12 void PartsTrayWithParts::sethadByPart_PartsTrayWithParts (std::vector< Part * > _hadByPart_PartsTrayWithParts)

Definition at line 49 of file PartsTrayWithParts.cpp.

8.43.3.13 void PartsTrayWithParts::sethasPartsTrayWithParts_Tray (PartsTray * _hasPartsTrayWithParts_Tray)

Definition at line 46 of file PartsTrayWithParts.cpp.

8.43.4 Member Data Documentation

8.43.4.1 DAO* PartsTrayWithParts::dao [private]

Definition at line 31 of file PartsTrayWithParts.h.

8.43.4.2 std::vector<Part*> PartsTrayWithParts::hadByPart_PartsTrayWithParts [private]

Definition at line 33 of file PartsTrayWithParts.h.

8.43.4.3 PartsTray* PartsTrayWithParts::hasPartsTrayWithParts_Tray [private]

Definition at line 32 of file PartsTrayWithParts.h.

8.43.4.4 int PartsTrayWithParts::PartsTrayWithPartsID [private]

Definition at line 30 of file PartsTrayWithParts.h.

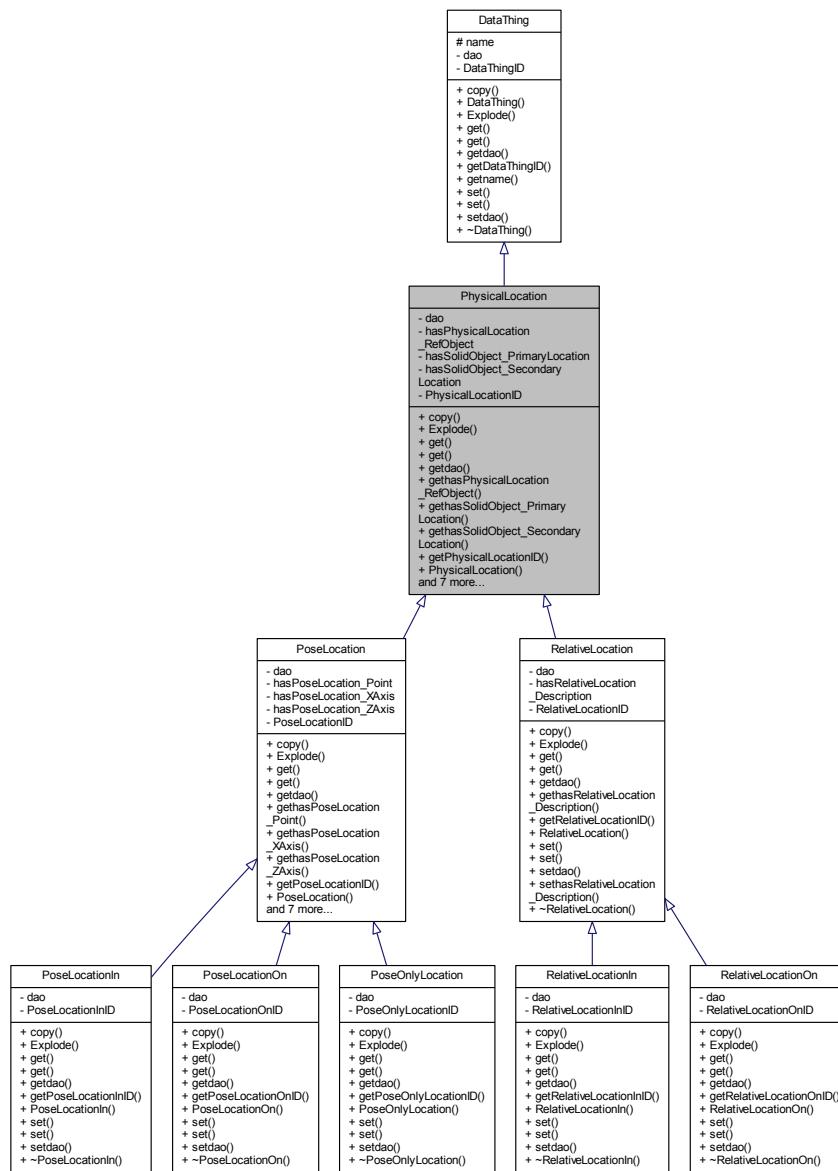
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsTrayWithParts.h
 - C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsTrayWithParts.cpp

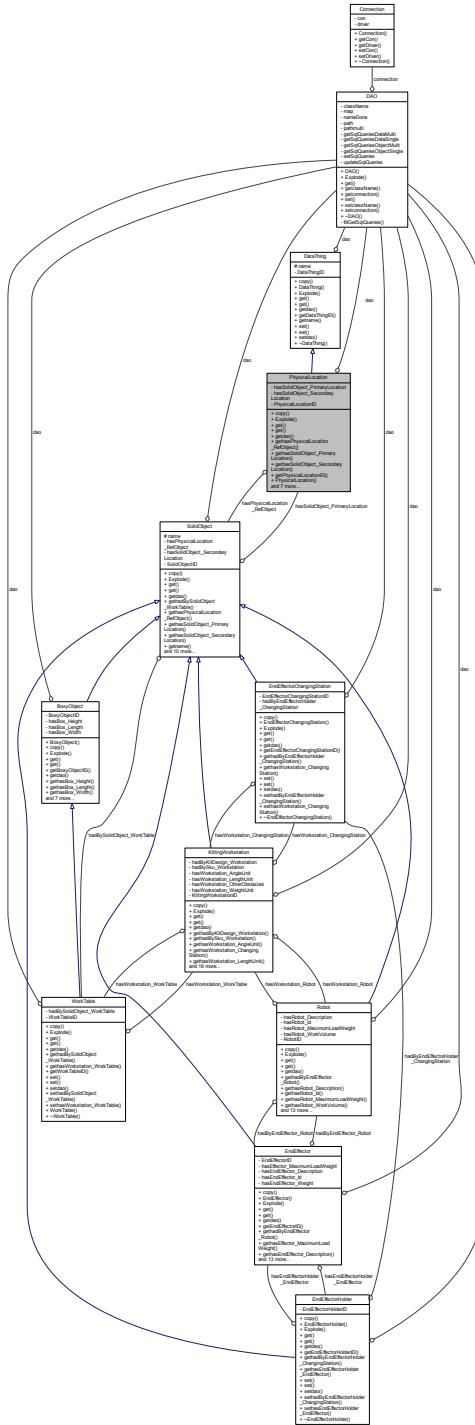
8.44 PhysicalLocation Class Reference

```
#include <PhysicalLocation.h>
```

Inheritance diagram for PhysicalLocation:



Collaboration diagram for PhysicalLocation:



Public Member Functions

- void **copy** (std::map< std::string, std::string > object)
 - std::vector< std::string > **Explode** (const std::string &str, char separator)

- void `get` (int id)
- void `get` (std::string name)
- DAO * `getdao` ()
- SolidObject * `gethasPhysicalLocation_RefObject` ()
- std::vector< SolidObject * > `gethasSolidObject_PrimaryLocation` ()
- std::vector< SolidObject * > `gethasSolidObject_SecondaryLocation` ()
- int `getPhysicalLocationID` ()
- PhysicalLocation (std::string name)
- void `set` (int id, PhysicalLocation *obj)
- void `set` (std::string name)
- void `setdao` (DAO *_dao)
- void `sethasPhysicalLocation_RefObject` (SolidObject *_hasPhysicalLocation_RefObject)
- void `sethasSolidObject_PrimaryLocation` (std::vector< SolidObject * > _hasSolidObject_PrimaryLocation)
- void `sethasSolidObject_SecondaryLocation` (std::vector< SolidObject * > _hasSolidObject_SecondaryLocation)
- ~PhysicalLocation ()

Private Attributes

- DAO * dao
- SolidObject * hasPhysicalLocation_RefObject
- std::vector< SolidObject * > hasSolidObject_PrimaryLocation
- std::vector< SolidObject * > hasSolidObject_SecondaryLocation
- int PhysicalLocationID

Additional Inherited Members

8.44.1 Detailed Description

Definition at line 28 of file PhysicalLocation.h.

8.44.2 Constructor & Destructor Documentation

8.44.2.1 PhysicalLocation::PhysicalLocation (std::string name)

Definition at line 20 of file PhysicalLocation.cpp.

8.44.2.2 PhysicalLocation::~PhysicalLocation ()

Definition at line 24 of file PhysicalLocation.cpp.

8.44.3 Member Function Documentation

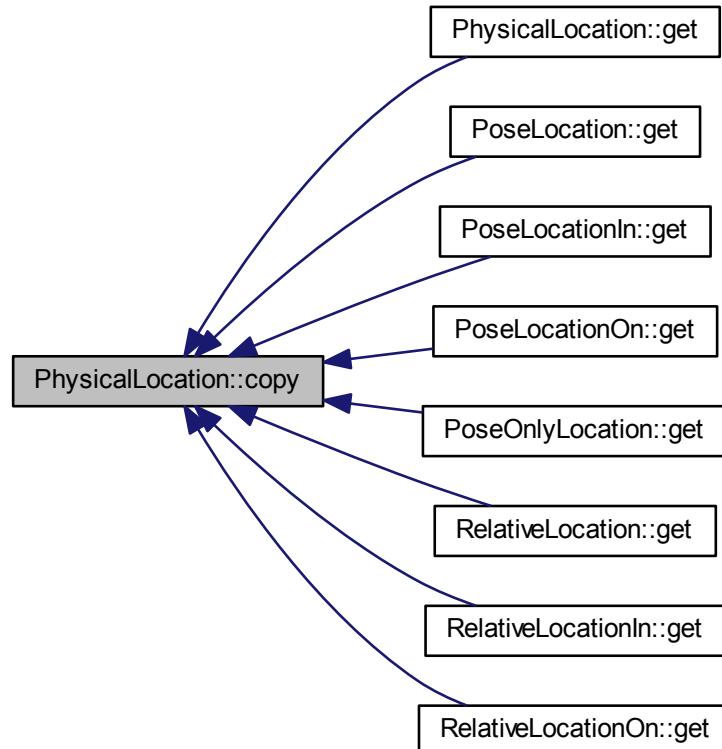
8.44.3.1 void PhysicalLocation::copy (std::map< std::string, std::string > object)

Definition at line 97 of file PhysicalLocation.cpp.

Here is the call graph for this function:



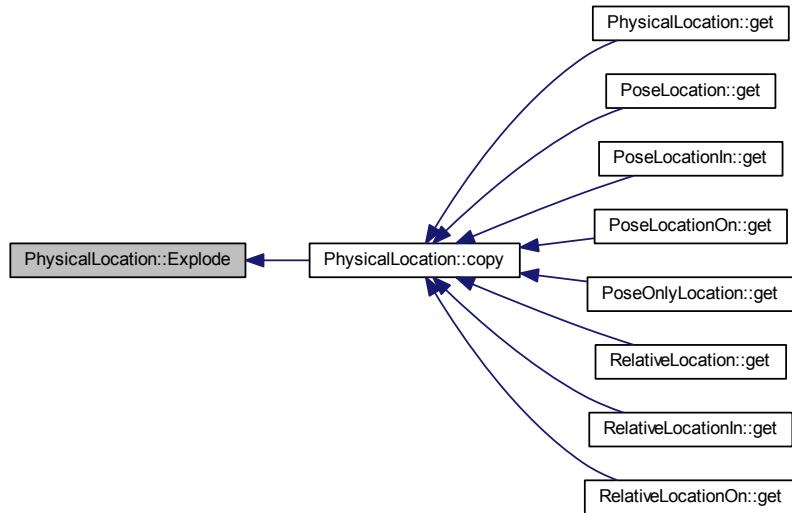
Here is the caller graph for this function:



8.44.3.2 std::vector< std::string > PhysicalLocation::Explode (const std::string & str, char separator)

Definition at line 121 of file PhysicalLocation.cpp.

Here is the caller graph for this function:

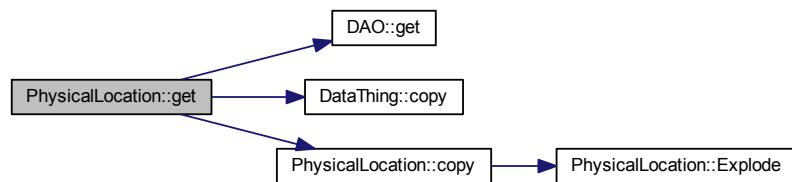


8.44.3.3 void PhysicalLocation::get(int id)

8.44.3.4 void PhysicalLocation::get(std::string name)

Definition at line 59 of file PhysicalLocation.cpp.

Here is the call graph for this function:



8.44.3.5 DAO * PhysicalLocation::getdao()

Definition at line 35 of file PhysicalLocation.cpp.

8.44.3.6 `SolidObject * PhysicalLocation::gethasPhysicalLocation_RefObject()`

Definition at line 38 of file PhysicalLocation.cpp.

8.44.3.7 `std::vector< SolidObject * > PhysicalLocation::gethasSolidObject_PrimaryLocation()`

Definition at line 44 of file PhysicalLocation.cpp.

8.44.3.8 `std::vector< SolidObject * > PhysicalLocation::gethasSolidObject_SecondaryLocation()`

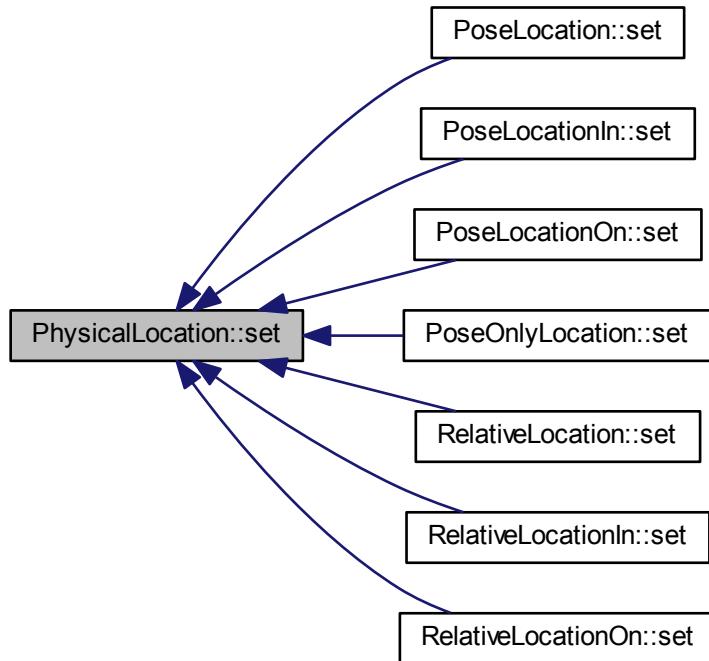
Definition at line 41 of file PhysicalLocation.cpp.

8.44.3.9 `int PhysicalLocation::getPhysicalLocationID()`

Definition at line 32 of file PhysicalLocation.cpp.

8.44.3.10 `void PhysicalLocation::set(int id, PhysicalLocation * obj)`

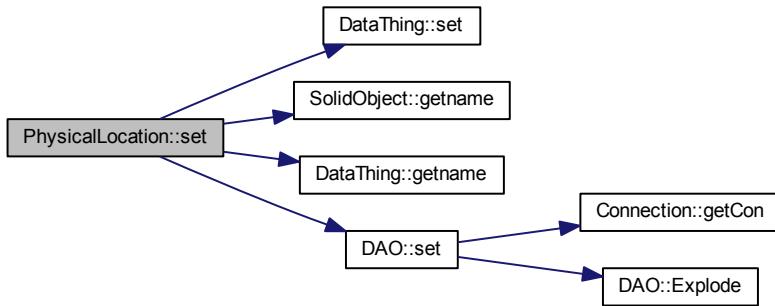
Here is the caller graph for this function:



8.44.3.11 void PhysicalLocation::set (std::string name)

Definition at line 69 of file PhysicalLocation.cpp.

Here is the call graph for this function:



8.44.3.12 void PhysicalLocation::setdao (DAO * _dao)

Definition at line 47 of file PhysicalLocation.cpp.

8.44.3.13 void PhysicalLocation::sethasPhysicalLocation_RefObject (SolidObject * _hasPhysicalLocation_RefObject)

Definition at line 50 of file PhysicalLocation.cpp.

8.44.3.14 void PhysicalLocation::sethasSolidObject_PrimaryLocation (std::vector< SolidObject * > _hasSolidObject_PrimaryLocation)

Definition at line 56 of file PhysicalLocation.cpp.

8.44.3.15 void PhysicalLocation::sethasSolidObject_SecondaryLocation (std::vector< SolidObject * > _hasSolidObject_SecondaryLocation)

Definition at line 53 of file PhysicalLocation.cpp.

8.44.4 Member Data Documentation

8.44.4.1 DAO* PhysicalLocation::dao [private]

Definition at line 30 of file PhysicalLocation.h.

8.44.4.2 SolidObject* PhysicalLocation::hasPhysicalLocation_RefObject [private]

Definition at line 31 of file PhysicalLocation.h.

8.44.4.3 `std::vector<SolidObject*> PhysicalLocation::hasSolidObject_PrimaryLocation [private]`

Definition at line 33 of file PhysicalLocation.h.

8.44.4.4 `std::vector<SolidObject*> PhysicalLocation::hasSolidObject_SecondaryLocation [private]`

Definition at line 32 of file PhysicalLocation.h.

8.44.4.5 `int PhysicalLocation::PhysicalLocationID [private]`

Definition at line 29 of file PhysicalLocation.h.

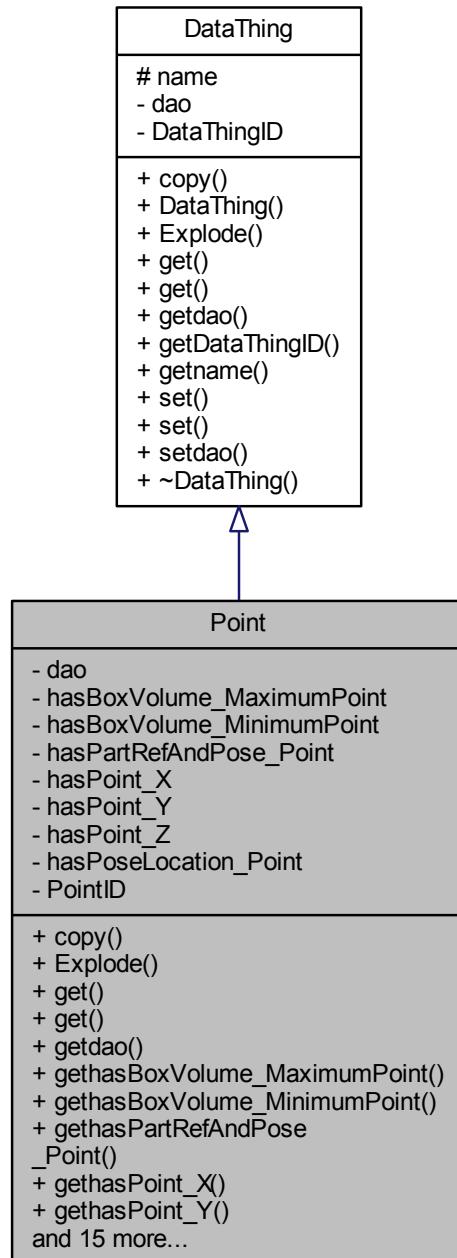
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[PhysicalLocation.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[PhysicalLocation.cpp](#)

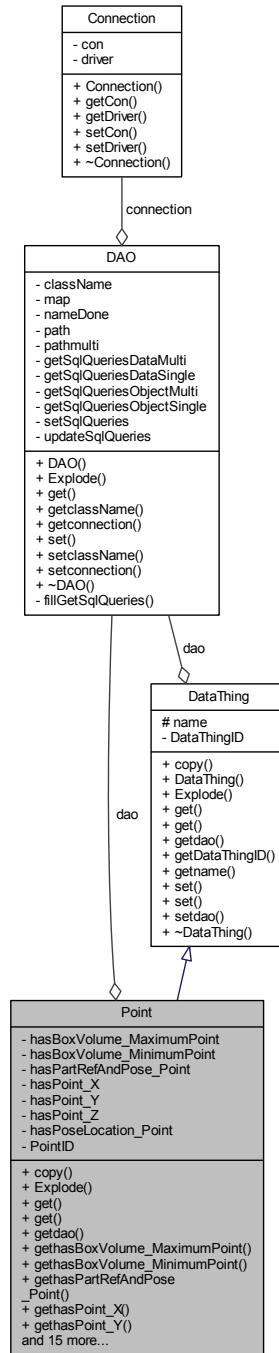
8.45 Point Class Reference

```
#include <Point.h>
```

Inheritance diagram for Point:



Collaboration diagram for Point:



Public Member Functions

- void [copy](#) (std::map< std::string, std::string > object)
- std::vector< std::string > [Explode](#) (const std::string &str, char separator)

- void `get` (int id)
- void `get` (std::string name)
- DAO * `getdao` ()
- std::vector< BoxVolume * > `gethasBoxVolume_MaximumPoint` ()
- std::vector< BoxVolume * > `gethasBoxVolume_MinimumPoint` ()
- std::vector< PartRefAndPose * > `gethasPartRefAndPose_Point` ()
- double `gethasPoint_X` ()
- double `gethasPoint_Y` ()
- double `gethasPoint_Z` ()
- std::vector< PoseLocation * > `gethasPoseLocation_Point` ()
- int `getPointID` ()
- Point (std::string name)
- void `set` (int id, Point *obj)
- void `set` (std::string name)
- void `setdao` (DAO *_dao)
- void `sethasBoxVolume_MaximumPoint` (std::vector< BoxVolume * > _hasBoxVolume_MaximumPoint)
- void `sethasBoxVolume_MinimumPoint` (std::vector< BoxVolume * > _hasBoxVolume_MinimumPoint)
- void `sethasPartRefAndPose_Point` (std::vector< PartRefAndPose * > _hasPartRefAndPose_Point)
- void `sethasPoint_X` (double _hasPoint_X)
- void `sethasPoint_Y` (double _hasPoint_Y)
- void `sethasPoint_Z` (double _hasPoint_Z)
- void `sethasPoseLocation_Point` (std::vector< PoseLocation * > _hasPoseLocation_Point)
- `~Point` ()

Private Attributes

- DAO * `dao`
- std::vector< BoxVolume * > `hasBoxVolume_MaximumPoint`
- std::vector< BoxVolume * > `hasBoxVolume_MinimumPoint`
- std::vector< PartRefAndPose * > `hasPartRefAndPose_Point`
- double `hasPoint_X`
- double `hasPoint_Y`
- double `hasPoint_Z`
- std::vector< PoseLocation * > `hasPoseLocation_Point`
- int `PointID`

Additional Inherited Members

8.45.1 Detailed Description

Definition at line 30 of file Point.h.

8.45.2 Constructor & Destructor Documentation

8.45.2.1 Point::Point (std::string name)

Definition at line 22 of file Point.cpp.

8.45.2.2 Point::~Point()

Definition at line 25 of file Point.cpp.

8.45.3 Member Function Documentation

8.45.3.1 void Point::copy(std::map< std::string, std::string > object)

Definition at line 144 of file Point.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



8.45.3.2 std::vector< std::string > Point::Explode(const std::string & str, char separator)

Definition at line 180 of file Point.cpp.

Here is the caller graph for this function:



8.45.3.3 void Point::get(int id)

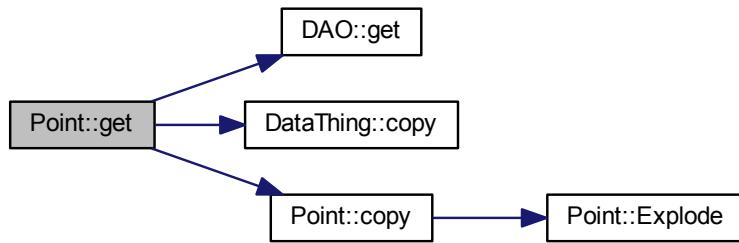
Here is the caller graph for this function:



8.45.3.4 void Point::get(std::string name)

Definition at line 87 of file Point.cpp.

Here is the call graph for this function:



8.45.3.5 DAO * Point::getdao()

Definition at line 48 of file Point.cpp.

8.45.3.6 std::vector< BoxVolume * > Point::gethasBoxVolume_MaximumPoint()

Definition at line 54 of file Point.cpp.

8.45.3.7 std::vector< BoxVolume * > Point::gethasBoxVolume_MinimumPoint()

Definition at line 57 of file Point.cpp.

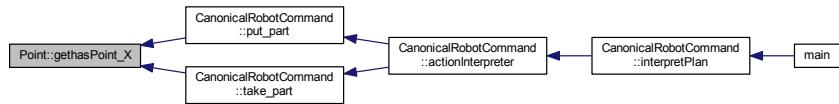
8.45.3.8 std::vector< PartRefAndPose * > Point::gethasPartRefAndPose_Point()

Definition at line 51 of file Point.cpp.

8.45.3.9 double Point::gethasPoint_X()

Definition at line 36 of file Point.cpp.

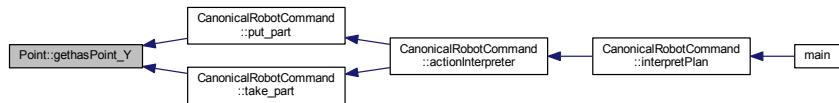
Here is the caller graph for this function:



8.45.3.10 double Point::gethasPoint_Y()

Definition at line 39 of file Point.cpp.

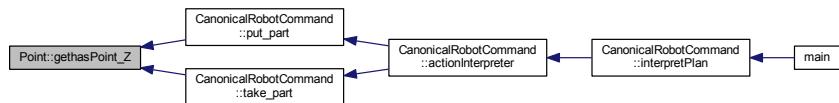
Here is the caller graph for this function:



8.45.3.11 double Point::gethasPoint_Z()

Definition at line 42 of file Point.cpp.

Here is the caller graph for this function:



8.45.3.12 std::vector< PoseLocation * > Point::gethasPoseLocation_Point()

Definition at line 60 of file Point.cpp.

8.45.3.13 int Point::getPointID()

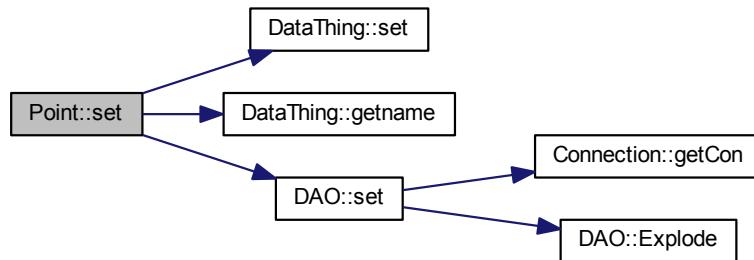
Definition at line 45 of file Point.cpp.

8.45.3.14 void Point::set (int *id*, Point * *obj*)

8.45.3.15 void Point::set (std::string *name*)

Definition at line 97 of file Point.cpp.

Here is the call graph for this function:



8.45.3.16 void Point::setdao (DAO * *_dao*)

Definition at line 72 of file Point.cpp.

8.45.3.17 void Point::sethasBoxVolume_MaximumPoint (std::vector< BoxVolume * > *_hasBoxVolume_MaximumPoint*)

Definition at line 78 of file Point.cpp.

8.45.3.18 void Point::sethasBoxVolume_MinimumPoint (std::vector< BoxVolume * > *_hasBoxVolume_MinimumPoint*)

Definition at line 81 of file Point.cpp.

8.45.3.19 void Point::sethasPartRefAndPose_Point (std::vector< PartRefAndPose * > *_hasPartRefAndPose_Point*)

Definition at line 75 of file Point.cpp.

8.45.3.20 void Point::sethasPoint_X (double *_hasPoint_X*)

Definition at line 63 of file Point.cpp.

8.45.3.21 void Point::sethasPoint_Y (double *_hasPoint_Y*)

Definition at line 66 of file Point.cpp.

8.45.3.22 void Point::sethasPoint_Z(double _hasPoint_Z)

Definition at line 69 of file Point.cpp.

8.45.3.23 void Point::sethasPoseLocation_Point(std::vector< PoseLocation * > _hasPoseLocation_Point)

Definition at line 84 of file Point.cpp.

8.45.4 Member Data Documentation

8.45.4.1 DAO* Point::dao [private]

Definition at line 35 of file Point.h.

8.45.4.2 std::vector<BoxVolume*> Point::hasBoxVolume_MaximumPoint [private]

Definition at line 37 of file Point.h.

8.45.4.3 std::vector<BoxVolume*> Point::hasBoxVolume_MinimumPoint [private]

Definition at line 38 of file Point.h.

8.45.4.4 std::vector<PartRefAndPose*> Point::hasPartRefAndPose_Point [private]

Definition at line 36 of file Point.h.

8.45.4.5 double Point::hasPoint_X [private]

Definition at line 31 of file Point.h.

8.45.4.6 double Point::hasPoint_Y [private]

Definition at line 32 of file Point.h.

8.45.4.7 double Point::hasPoint_Z [private]

Definition at line 33 of file Point.h.

8.45.4.8 std::vector<PoseLocation*> Point::hasPoseLocation_Point [private]

Definition at line 39 of file Point.h.

8.45.4.9 int Point::PointID [private]

Definition at line 34 of file Point.h.

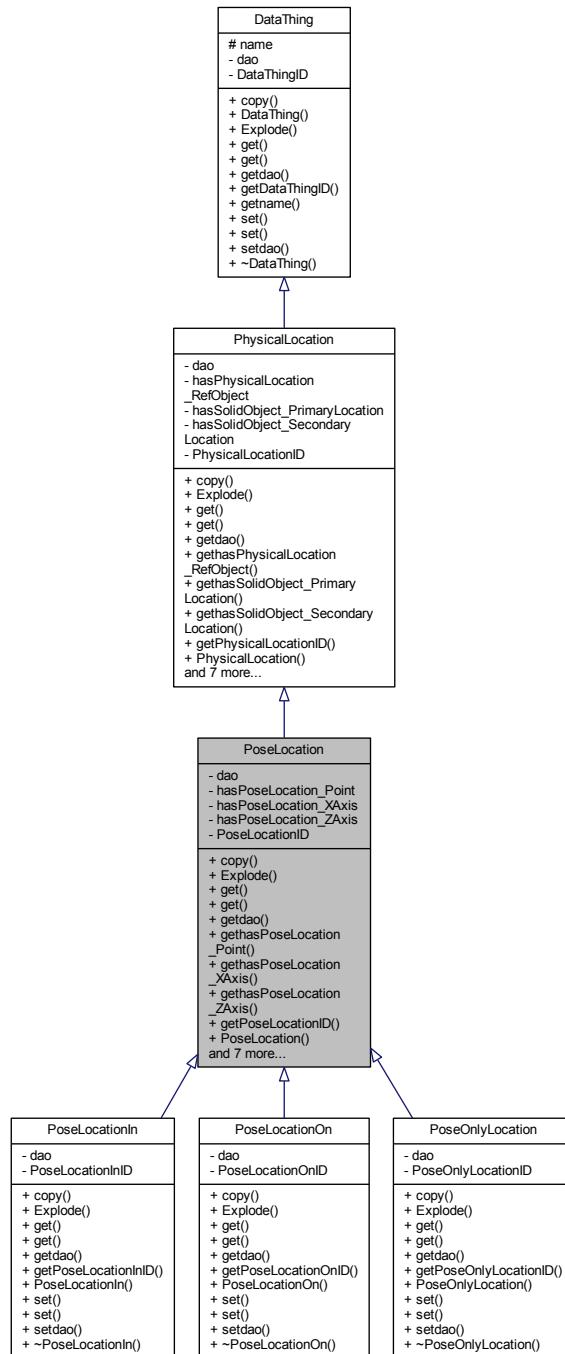
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[Point.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[Point.cpp](#)

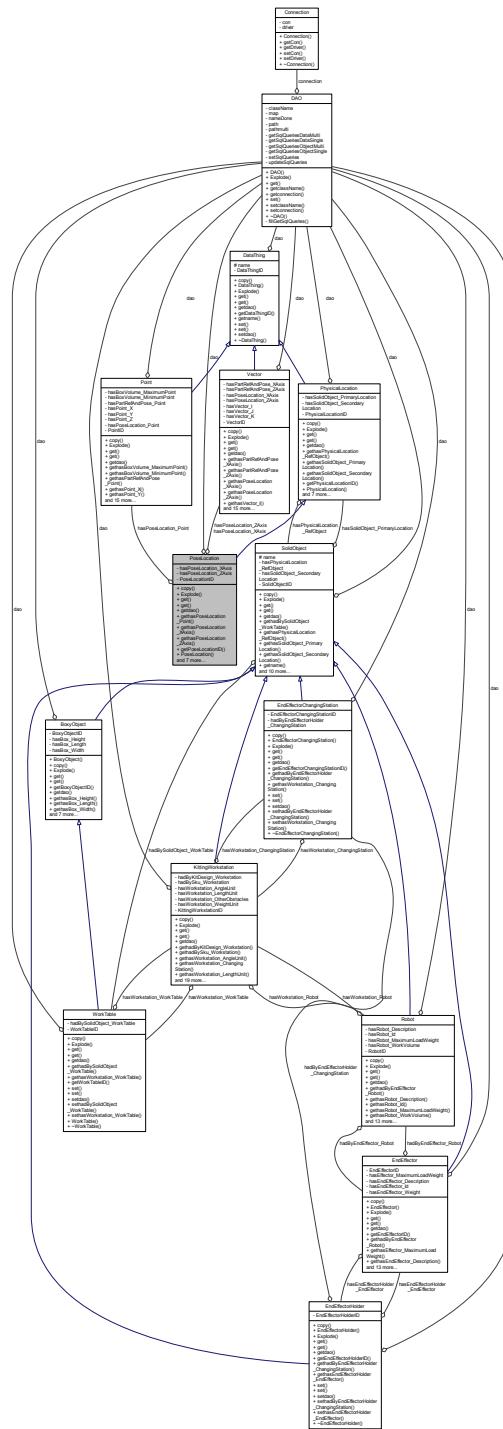
8.46 PoseLocation Class Reference

```
#include <PoseLocation.h>
```

Inheritance diagram for PoseLocation:



Collaboration diagram for PoseLocation:



Public Member Functions

- void **copy** (std::map< std::string, std::string > object)
 - std::vector< std::string > **Explode** (const std::string &str, char separator)

- void [get](#) (int id)
- void [get](#) (std::string name)
- DAO * [getdao](#) ()
- Point * [gethasPoseLocation_Point](#) ()
- Vector * [gethasPoseLocation_XAxis](#) ()
- Vector * [gethasPoseLocation_ZAxis](#) ()
- int [getPoseLocationID](#) ()
- PoseLocation ([std::string name](#))
- void [set](#) (int id, PoseLocation *obj)
- void [set](#) (std::string name)
- void [setdao](#) (DAO *_dao)
- void [sethasPoseLocation_Point](#) (Point *_hasPoseLocation_Point)
- void [sethasPoseLocation_XAxis](#) (Vector *_hasPoseLocation_XAxis)
- void [sethasPoseLocation_ZAxis](#) (Vector *_hasPoseLocation_ZAxis)
- ~[PoseLocation](#) ()

Private Attributes

- DAO * dao
- Point * hasPoseLocation_Point
- Vector * hasPoseLocation_XAxis
- Vector * hasPoseLocation_ZAxis
- int PoseLocationID

Additional Inherited Members

8.46.1 Detailed Description

Definition at line 29 of file PoseLocation.h.

8.46.2 Constructor & Destructor Documentation

8.46.2.1 PoseLocation::PoseLocation (std::string name)

Definition at line 21 of file PoseLocation.cpp.

8.46.2.2 PoseLocation::~PoseLocation ()

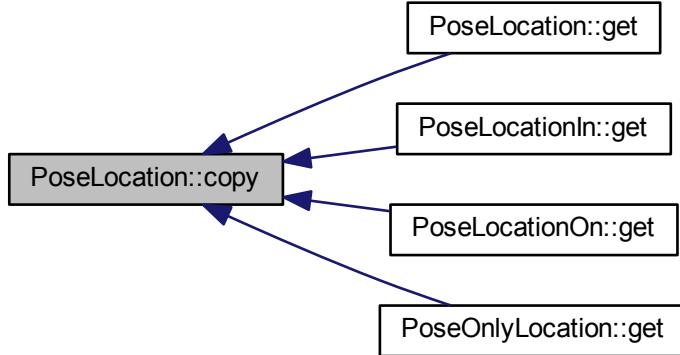
Definition at line 27 of file PoseLocation.cpp.

8.46.3 Member Function Documentation

8.46.3.1 void PoseLocation::copy (std::map< std::string, std::string > object)

Definition at line 95 of file PoseLocation.cpp.

Here is the caller graph for this function:

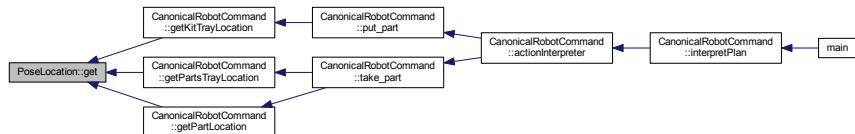


8.46.3.2 `std::vector< std::string > PoseLocation::Explode (const std::string & str, char separator)`

Definition at line 113 of file `PoseLocation.cpp`.

8.46.3.3 `void PoseLocation::get (int id)`

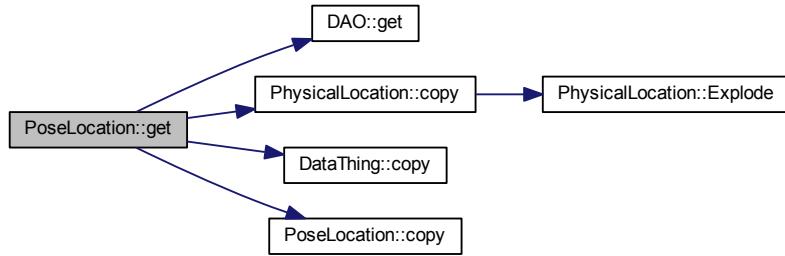
Here is the caller graph for this function:



8.46.3.4 `void PoseLocation::get (std::string name)`

Definition at line 60 of file `PoseLocation.cpp`.

Here is the call graph for this function:



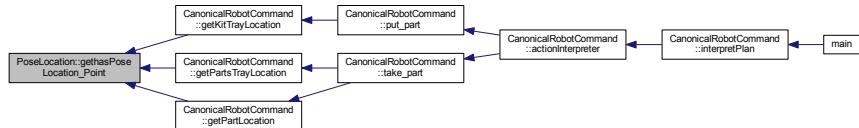
8.46.3.5 DAO * PoseLocation::getdao()

Definition at line 36 of file PoseLocation.cpp.

8.46.3.6 Point * PoseLocation::gethasPoseLocation_Point()

Definition at line 39 of file PoseLocation.cpp.

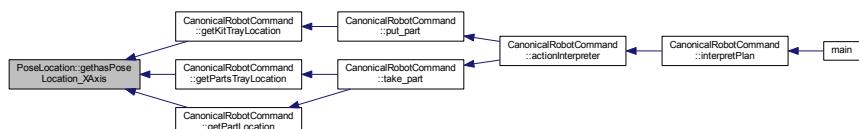
Here is the caller graph for this function:



8.46.3.7 Vector * PoseLocation::gethasPoseLocation_XAxis()

Definition at line 45 of file PoseLocation.cpp.

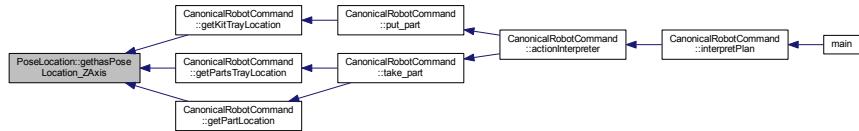
Here is the caller graph for this function:



8.46.3.8 `Vector * PoseLocation::getHasPoseLocation_ZAxis()`

Definition at line 42 of file PoseLocation.cpp.

Here is the caller graph for this function:

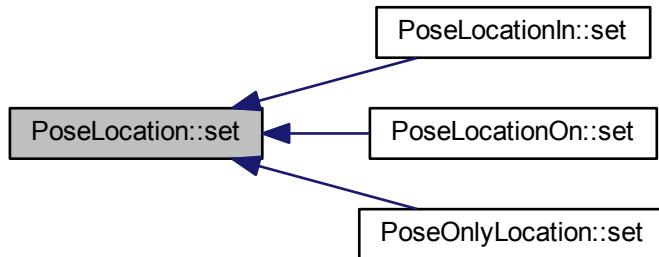


8.46.3.9 `int PoseLocation::getPoseLocationID()`

Definition at line 33 of file PoseLocation.cpp.

8.46.3.10 `void PoseLocation::set(int id, PoseLocation * obj)`

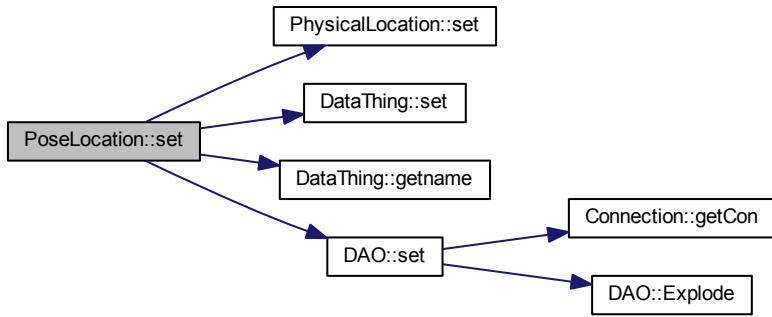
Here is the caller graph for this function:



8.46.3.11 `void PoseLocation::set(std::string name)`

Definition at line 73 of file PoseLocation.cpp.

Here is the call graph for this function:



8.46.3.12 void PoseLocation::setdao (DAO * _dao)

Definition at line 48 of file PoseLocation.cpp.

8.46.3.13 void PoseLocation::sethasPoseLocation_Point (Point * _hasPoseLocation_Point)

Definition at line 51 of file PoseLocation.cpp.

8.46.3.14 void PoseLocation::sethasPoseLocation_XAxis (Vector * _hasPoseLocation_XAxis)

Definition at line 57 of file PoseLocation.cpp.

8.46.3.15 void PoseLocation::sethasPoseLocation_ZAxis (Vector * _hasPoseLocation_ZAxis)

Definition at line 54 of file PoseLocation.cpp.

8.46.4 Member Data Documentation

8.46.4.1 DAO* PoseLocation::dao [private]

Definition at line 31 of file PoseLocation.h.

8.46.4.2 Point* PoseLocation::hasPoseLocation_Point [private]

Definition at line 32 of file PoseLocation.h.

8.46.4.3 Vector* PoseLocation::hasPoseLocation_XAxis [private]

Definition at line 34 of file PoseLocation.h.

8.46.4.4 `Vector* PoseLocation::hasPoseLocation_ZAxis [private]`

Definition at line 33 of file PoseLocation.h.

8.46.4.5 `int PoseLocation::PoseLocationID [private]`

Definition at line 30 of file PoseLocation.h.

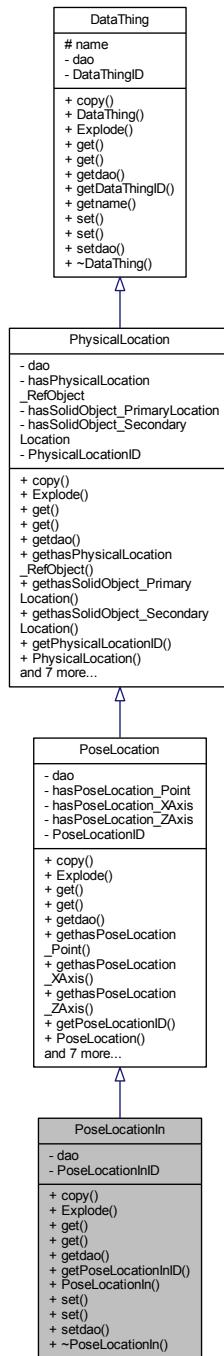
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[PoseLocation.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[PoseLocation.cpp](#)

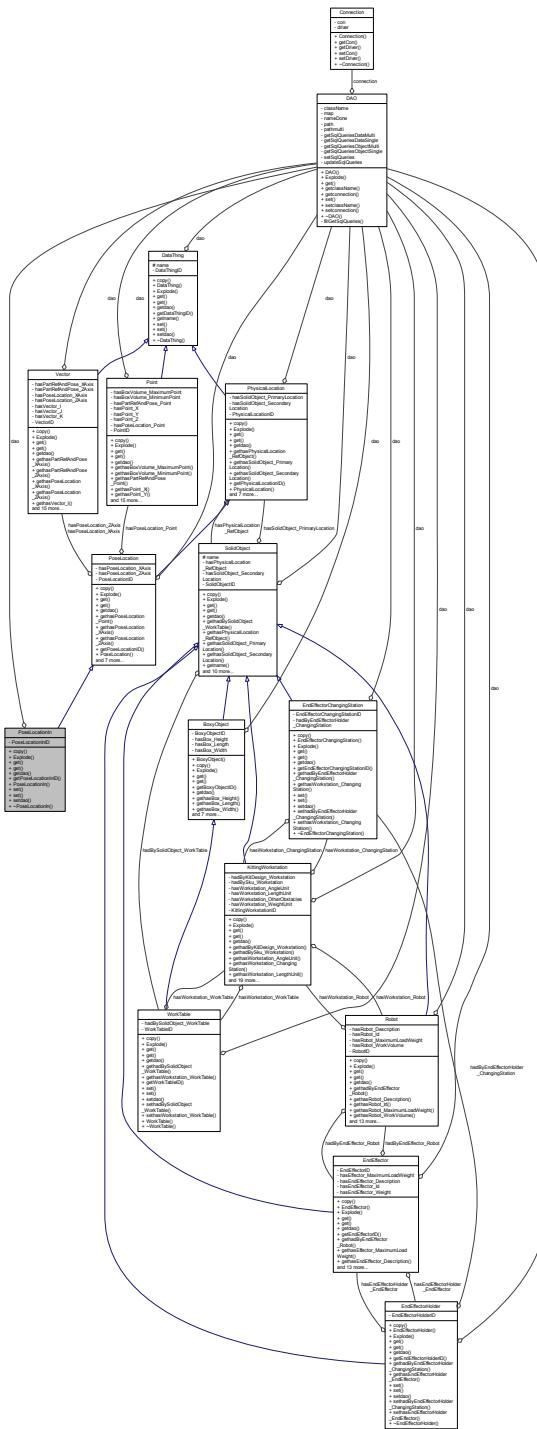
8.47 PoseLocationIn Class Reference

```
#include <PoseLocationIn.h>
```

Inheritance diagram for PoseLocationIn:



Collaboration diagram for PoseLocationIn:



Public Member Functions

- void `copy` (std::map< std::string, std::string > object)
 - std::vector< std::string > `Explode` (const std::string &str, char separator)

- void `get` (int id)
- void `get` (std::string name)
- DAO * `getdao` ()
- int `getPoseLocationInID` ()
- PoseLocationIn (std::string name)
- void `set` (int id, PoseLocationIn *obj)
- void `set` (std::string name)
- void `setdao` (DAO *_dao)
- ~PoseLocationIn ()

Private Attributes

- DAO * dao
- int PoseLocationInID

Additional Inherited Members

8.47.1 Detailed Description

Definition at line 27 of file PoseLocationIn.h.

8.47.2 Constructor & Destructor Documentation

8.47.2.1 PoseLocationIn::PoseLocationIn (std::string name)

Definition at line 19 of file PoseLocationIn.cpp.

8.47.2.2 PoseLocationIn::~PoseLocationIn ()

Definition at line 22 of file PoseLocationIn.cpp.

8.47.3 Member Function Documentation

8.47.3.1 void PoseLocationIn::copy (std::map< std::string, std::string > object)

Definition at line 68 of file PoseLocationIn.cpp.

Here is the caller graph for this function:



8.47.3.2 `std::vector< std::string > PoseLocationIn::Explode (const std::string & str, char separator)`

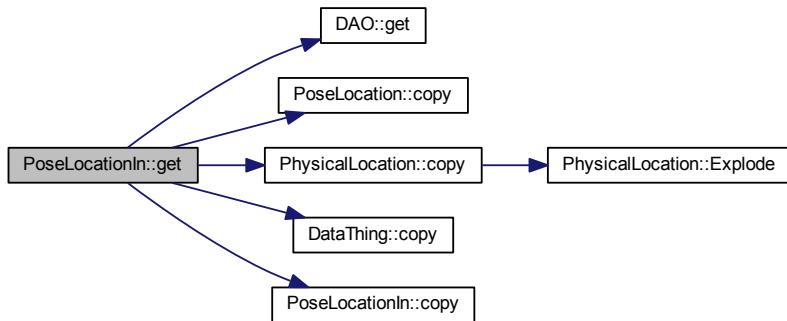
Definition at line 77 of file PoseLocationIn.cpp.

8.47.3.3 `void PoseLocationIn::get (int id)`

8.47.3.4 `void PoseLocationIn::get (std::string name)`

Definition at line 34 of file PoseLocationIn.cpp.

Here is the call graph for this function:



8.47.3.5 `DAO * PoseLocationIn::getdao ()`

Definition at line 28 of file PoseLocationIn.cpp.

8.47.3.6 `int PoseLocationIn::getPoseLocationInID ()`

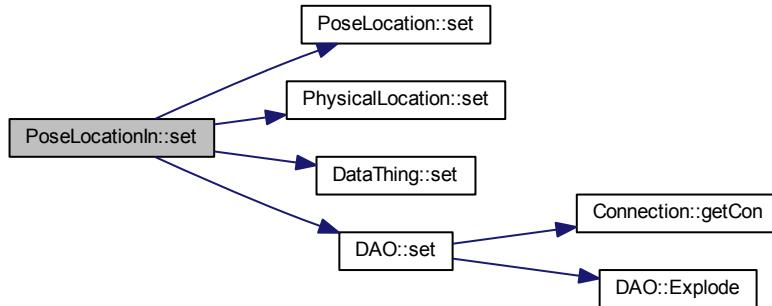
Definition at line 25 of file PoseLocationIn.cpp.

8.47.3.7 `void PoseLocationIn::set (int id, PoseLocationIn * obj)`

8.47.3.8 `void PoseLocationIn::set (std::string name)`

Definition at line 50 of file PoseLocationIn.cpp.

Here is the call graph for this function:



8.47.3.9 void PoseLocationIn::setdao (DAO * _dao)

Definition at line 31 of file `PoseLocationIn.cpp`.

8.47.4 Member Data Documentation

8.47.4.1 DAO* PoseLocationIn::dao [private]

Definition at line 29 of file `PoseLocationIn.h`.

8.47.4.2 int PoseLocationIn::PoseLocationInID [private]

Definition at line 28 of file `PoseLocationIn.h`.

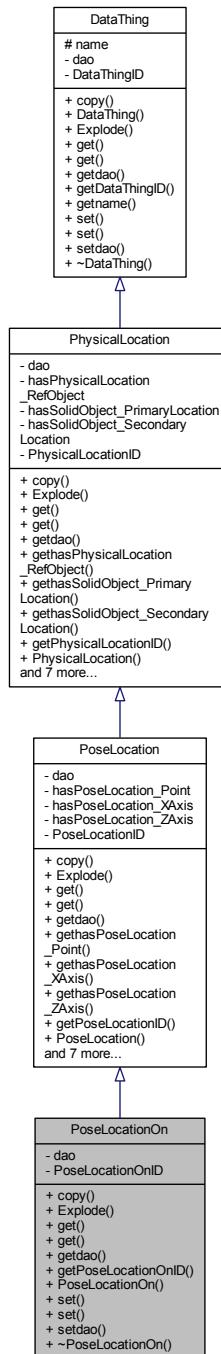
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[PoseLocationIn.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[PoseLocationIn.cpp](#)

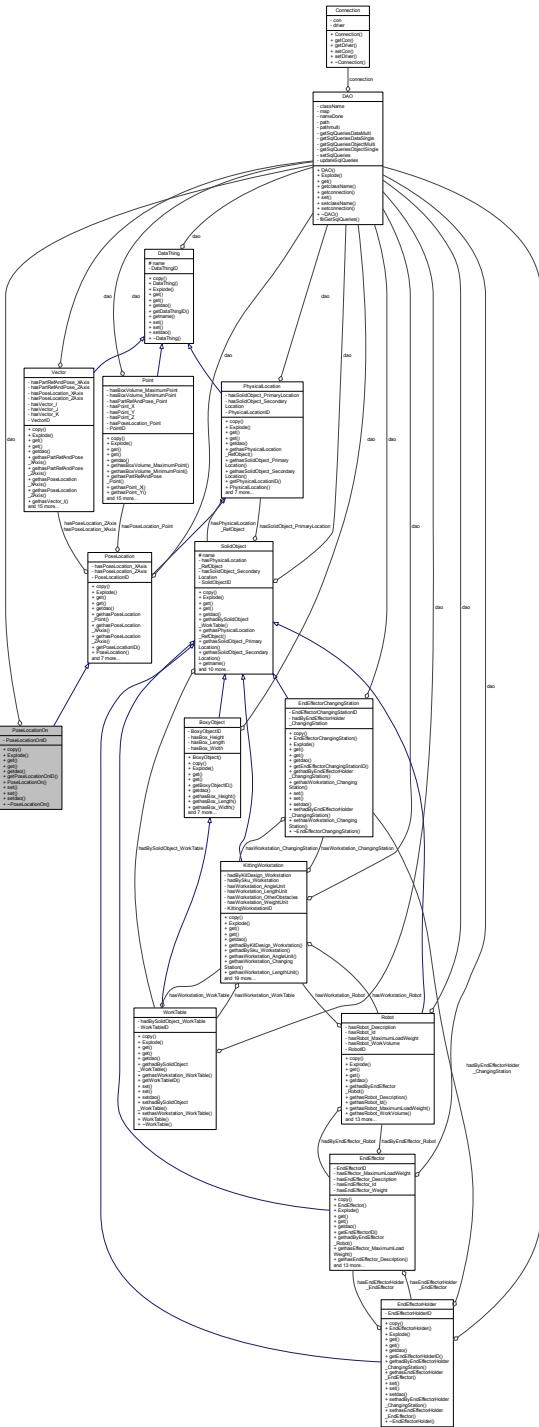
8.48 PoseLocationOn Class Reference

```
#include <PoseLocationOn.h>
```

Inheritance diagram for PoseLocationOn:



Collaboration diagram for PoseLocationOn:



Public Member Functions

- void **copy** (std::map< std::string, std::string > object)
 - std::vector< std::string > **Explode** (const std::string &str, char separator)

- void [get](#) (int id)
- void [get](#) (std::string name)
- DAO * [getdao](#) ()
- int [getPoseLocationOnID](#) ()
- PoseLocationOn ([std::string name](#))
- void [set](#) (int id, PoseLocationOn *obj)
- void [set](#) (std::string name)
- void [setdao](#) (DAO *_dao)
- ~[PoseLocationOn](#) ()

Private Attributes

- DAO * [dao](#)
- int [PoseLocationOnID](#)

Additional Inherited Members

8.48.1 Detailed Description

Definition at line 27 of file PoseLocationOn.h.

8.48.2 Constructor & Destructor Documentation

8.48.2.1 PoseLocationOn::PoseLocationOn (std::string name)

Definition at line 19 of file PoseLocationOn.cpp.

8.48.2.2 PoseLocationOn::~PoseLocationOn ()

Definition at line 22 of file PoseLocationOn.cpp.

8.48.3 Member Function Documentation

8.48.3.1 void PoseLocationOn::copy (std::map< std::string, std::string > object)

Definition at line 68 of file PoseLocationOn.cpp.

Here is the caller graph for this function:



8.48.3.2 `std::vector< std::string > PoseLocationOn::Explode (const std::string & str, char separator)`

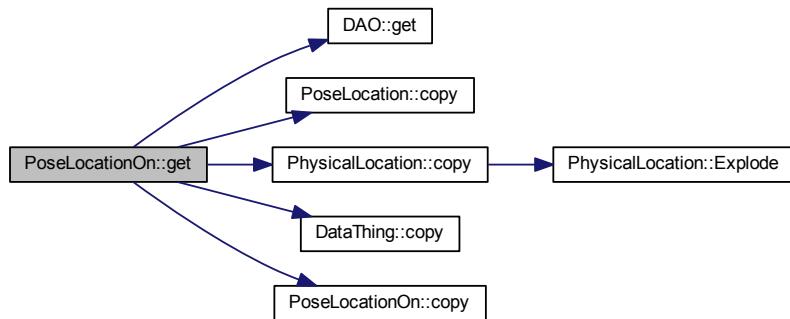
Definition at line 77 of file PoseLocationOn.cpp.

8.48.3.3 `void PoseLocationOn::get (int id)`

8.48.3.4 `void PoseLocationOn::get (std::string name)`

Definition at line 34 of file PoseLocationOn.cpp.

Here is the call graph for this function:



8.48.3.5 `DAO * PoseLocationOn::getdao ()`

Definition at line 28 of file PoseLocationOn.cpp.

8.48.3.6 `int PoseLocationOn::getPoseLocationOnID ()`

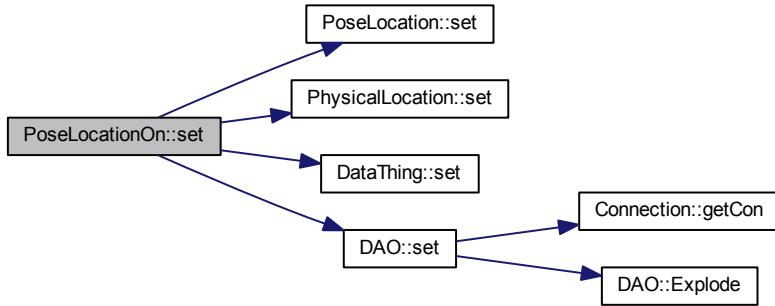
Definition at line 25 of file PoseLocationOn.cpp.

8.48.3.7 `void PoseLocationOn::set (int id, PoseLocationOn * obj)`

8.48.3.8 `void PoseLocationOn::set (std::string name)`

Definition at line 50 of file PoseLocationOn.cpp.

Here is the call graph for this function:



8.48.3.9 void PoseLocationOn::setdao (DAO * _dao)

Definition at line 31 of file PoseLocationOn.cpp.

8.48.4 Member Data Documentation

8.48.4.1 DAO* PoseLocationOn::dao [private]

Definition at line 29 of file PoseLocationOn.h.

8.48.4.2 int PoseLocationOn::PoseLocationOnID [private]

Definition at line 28 of file PoseLocationOn.h.

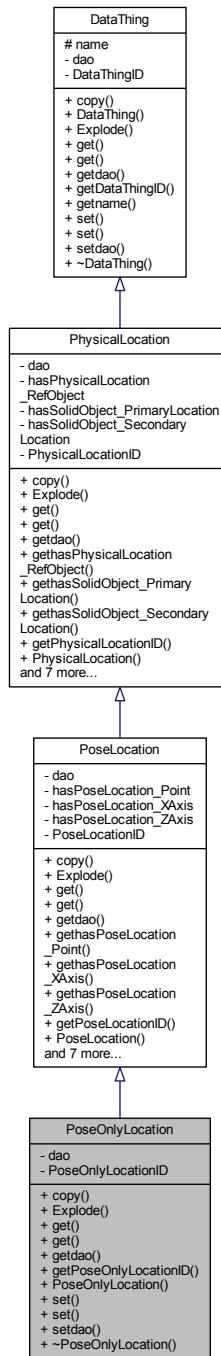
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[PoseLocationOn.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[PoseLocationOn.cpp](#)

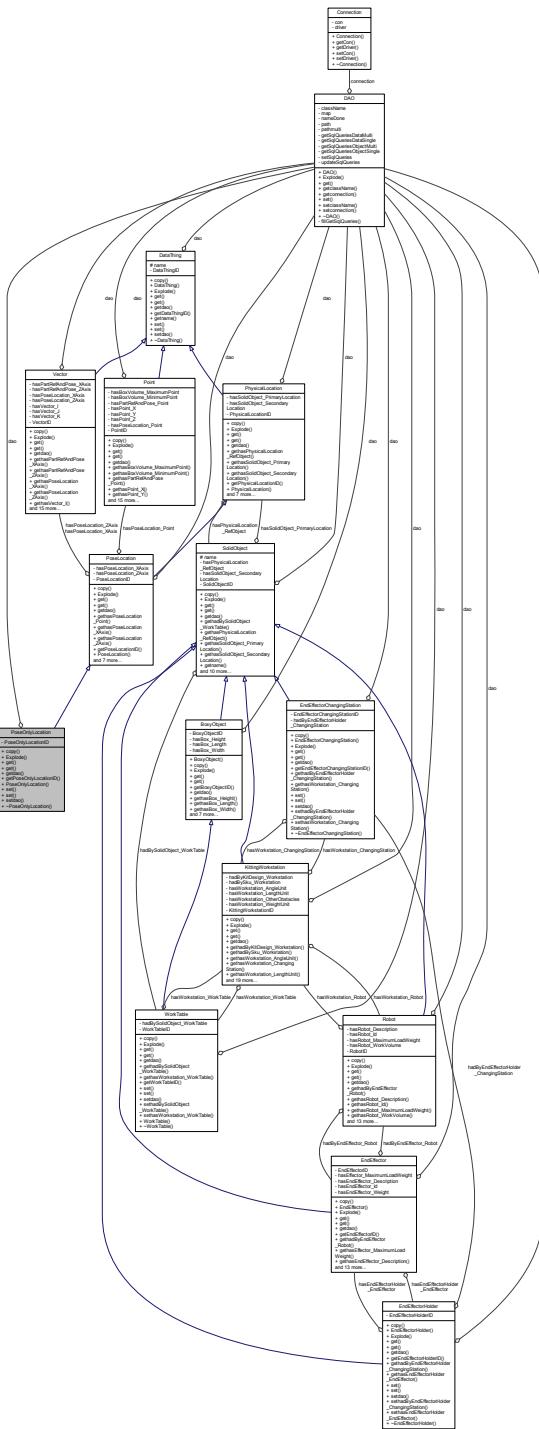
8.49 PoseOnlyLocation Class Reference

```
#include <PoseOnlyLocation.h>
```

Inheritance diagram for PoseOnlyLocation:



Collaboration diagram for PoseOnlyLocation:



Public Member Functions

- void **copy** (std::map< std::string, std::string > object)
 - std::vector< std::string > **Explode** (const std::string &str, char separator)

- void `get` (int id)
- void `get` (std::string name)
- DAO * `getdao` ()
- int `getPoseOnlyLocationID` ()
- PoseOnlyLocation (std::string name)
- void `set` (int id, PoseOnlyLocation *obj)
- void `set` (std::string name)
- void `setdao` (DAO *_dao)
- ~PoseOnlyLocation ()

Private Attributes

- DAO * dao
- int PoseOnlyLocationID

Additional Inherited Members

8.49.1 Detailed Description

Definition at line 27 of file PoseOnlyLocation.h.

8.49.2 Constructor & Destructor Documentation

8.49.2.1 PoseOnlyLocation::PoseOnlyLocation (std::string name)

Definition at line 19 of file PoseOnlyLocation.cpp.

8.49.2.2 PoseOnlyLocation::~PoseOnlyLocation ()

Definition at line 22 of file PoseOnlyLocation.cpp.

8.49.3 Member Function Documentation

8.49.3.1 void PoseOnlyLocation::copy (std::map< std::string, std::string > object)

Definition at line 68 of file PoseOnlyLocation.cpp.

Here is the caller graph for this function:



8.49.3.2 `std::vector< std::string > PoseOnlyLocation::Explode (const std::string & str, char separator)`

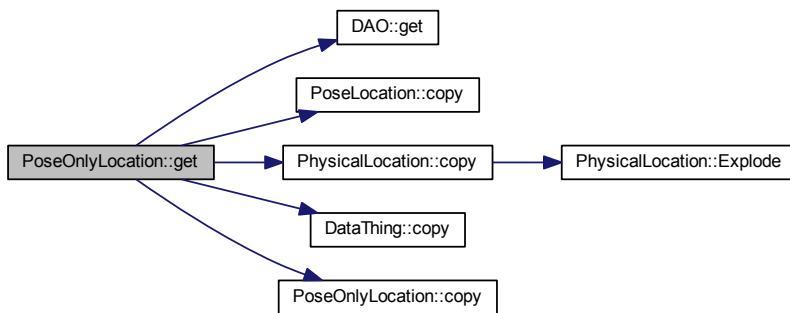
Definition at line 77 of file PoseOnlyLocation.cpp.

8.49.3.3 `void PoseOnlyLocation::get (int id)`

8.49.3.4 `void PoseOnlyLocation::get (std::string name)`

Definition at line 34 of file PoseOnlyLocation.cpp.

Here is the call graph for this function:



8.49.3.5 `DAO * PoseOnlyLocation::getdao ()`

Definition at line 28 of file PoseOnlyLocation.cpp.

8.49.3.6 `int PoseOnlyLocation::getPoseOnlyLocationID ()`

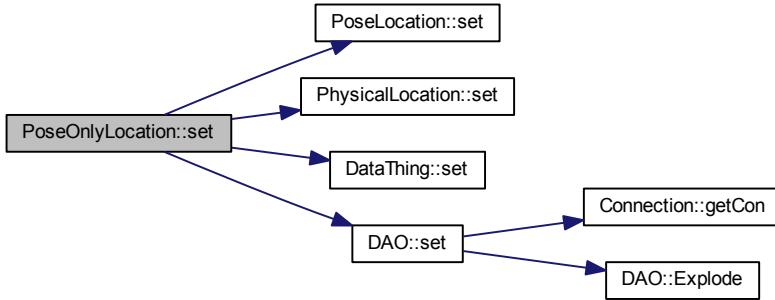
Definition at line 25 of file PoseOnlyLocation.cpp.

8.49.3.7 `void PoseOnlyLocation::set (int id, PoseOnlyLocation * obj)`

8.49.3.8 `void PoseOnlyLocation::set (std::string name)`

Definition at line 50 of file PoseOnlyLocation.cpp.

Here is the call graph for this function:



8.49.3.9 void PoseOnlyLocation::setdao (DAO * _dao)

Definition at line 31 of file PoseOnlyLocation.cpp.

8.49.4 Member Data Documentation

8.49.4.1 DAO* PoseOnlyLocation::dao [private]

Definition at line 29 of file PoseOnlyLocation.h.

8.49.4.2 int PoseOnlyLocation::PoseOnlyLocationID [private]

Definition at line 28 of file PoseOnlyLocation.h.

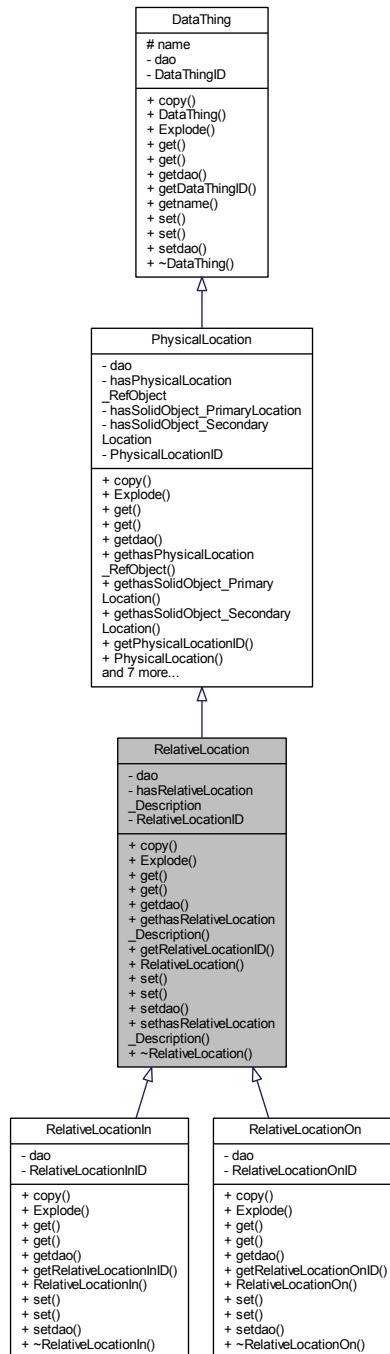
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[PoseOnlyLocation.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[PoseOnlyLocation.cpp](#)

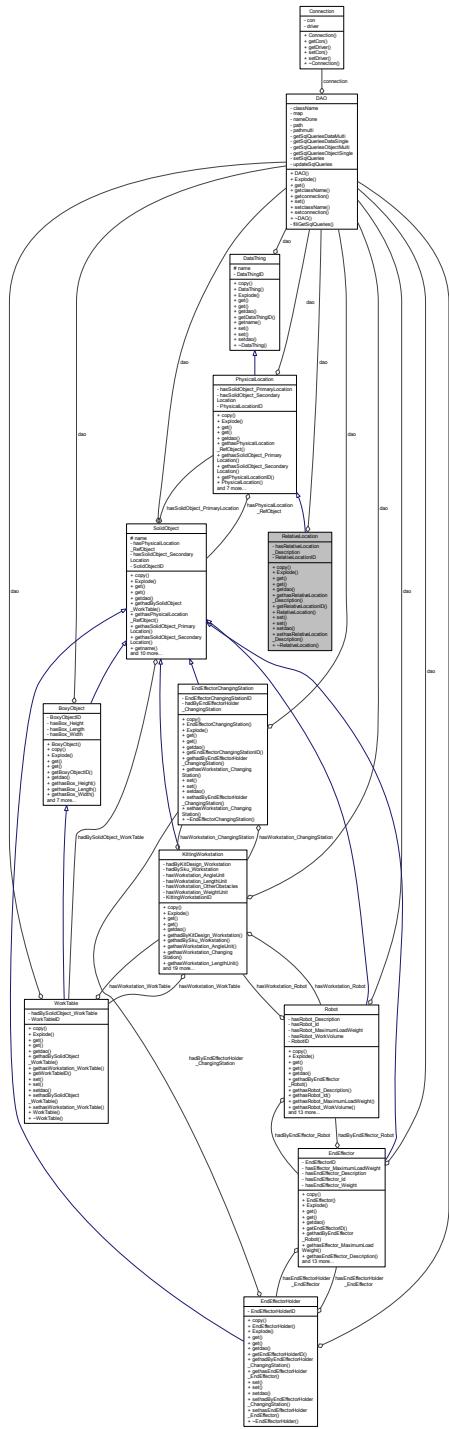
8.50 RelativeLocation Class Reference

```
#include <RelativeLocation.h>
```

Inheritance diagram for RelativeLocation:



Collaboration diagram for RelativeLocation:



Public Member Functions

- void **copy** (std::map< std::string, std::string > object)
 - std::vector< std::string > **Explode** (const std::string &str, char separator)

- void `get` (int *id*)
- void `get` (std::string *name*)
- DAO * `getdao` ()
- std::string `gethasRelativeLocation_Description` ()
- int `getRelativeLocationID` ()
- RelativeLocation (*std::string name*)
- void `set` (int *id*, RelativeLocation **obj*)
- void `set` (std::string *name*)
- void `setdao` (DAO *_*dao*)
- void `sethasRelativeLocation_Description` (std::string _*hasRelativeLocation_Description*)
- ~RelativeLocation ()

Private Attributes

- DAO * `dao`
- std::string `hasRelativeLocation_Description`
- int `RelativeLocationID`

Additional Inherited Members

8.50.1 Detailed Description

Definition at line 27 of file RelativeLocation.h.

8.50.2 Constructor & Destructor Documentation

8.50.2.1 RelativeLocation::RelativeLocation (std::string *name*)

Definition at line 19 of file RelativeLocation.cpp.

8.50.2.2 RelativeLocation::~RelativeLocation ()

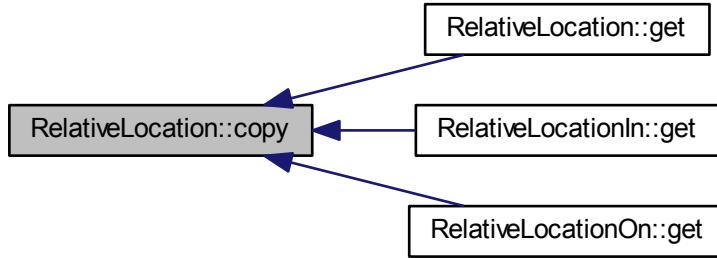
Definition at line 22 of file RelativeLocation.cpp.

8.50.3 Member Function Documentation

8.50.3.1 void RelativeLocation::copy (std::map< std::string, std::string > *object*)

Definition at line 70 of file RelativeLocation.cpp.

Here is the caller graph for this function:



8.50.3.2 `std::vector< std::string > RelativeLocation::Explode (const std::string & str, char separator)`

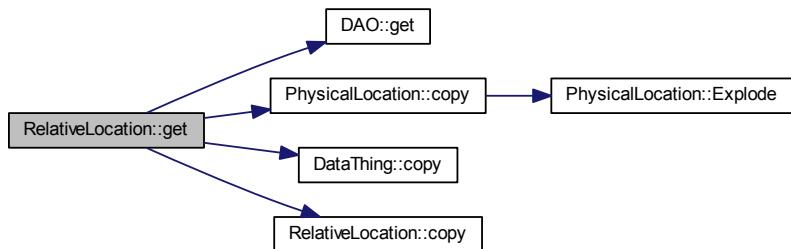
Definition at line 80 of file `RelativeLocation.cpp`.

8.50.3.3 `void RelativeLocation::get (int id)`

8.50.3.4 `void RelativeLocation::get (std::string name)`

Definition at line 40 of file `RelativeLocation.cpp`.

Here is the call graph for this function:



8.50.3.5 `DAO * RelativeLocation::getdao ()`

Definition at line 31 of file `RelativeLocation.cpp`.

8.50.3.6 `std::string RelativeLocation::getHasRelativeLocation_Description()`

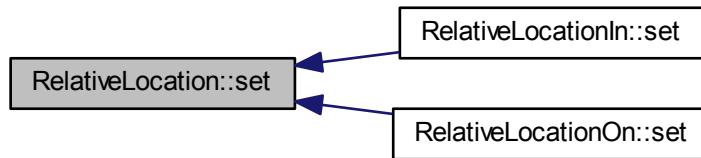
Definition at line 25 of file RelativeLocation.cpp.

8.50.3.7 `int RelativeLocation::getRelativeLocationID()`

Definition at line 28 of file RelativeLocation.cpp.

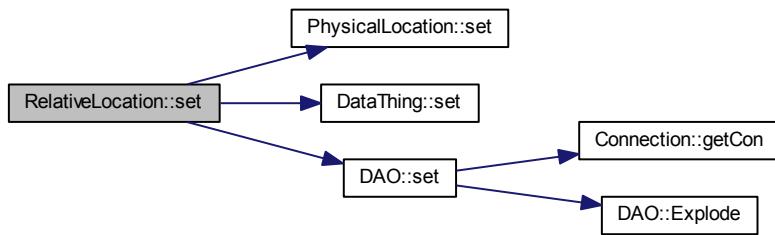
8.50.3.8 `void RelativeLocation::set(int id, RelativeLocation * obj)`

Here is the caller graph for this function:

8.50.3.9 `void RelativeLocation::set(std::string name)`

Definition at line 53 of file RelativeLocation.cpp.

Here is the call graph for this function:

8.50.3.10 `void RelativeLocation::setdao(DAO * _dao)`

Definition at line 37 of file RelativeLocation.cpp.

8.50.3.11 void RelativeLocation::sethasRelativeLocation_Description (std::string *_hasRelativeLocation_Description*)

Definition at line 34 of file RelativeLocation.cpp.

8.50.4 Member Data Documentation

8.50.4.1 DAO* RelativeLocation::dao [private]

Definition at line 30 of file RelativeLocation.h.

8.50.4.2 std::string RelativeLocation::hasRelativeLocation_Description [private]

Definition at line 28 of file RelativeLocation.h.

8.50.4.3 int RelativeLocation::RelativeLocationID [private]

Definition at line 29 of file RelativeLocation.h.

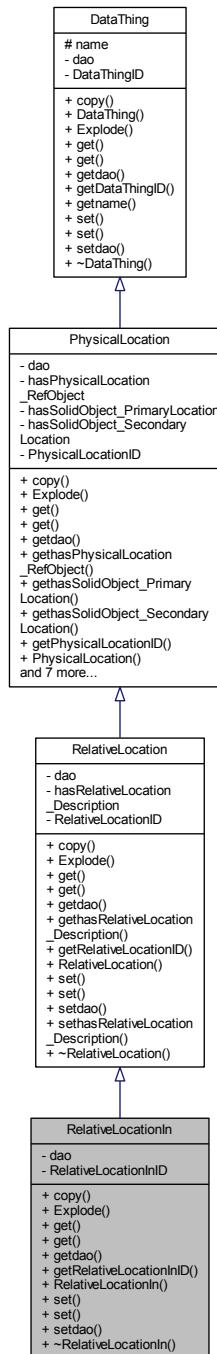
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[RelativeLocation.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[RelativeLocation.cpp](#)

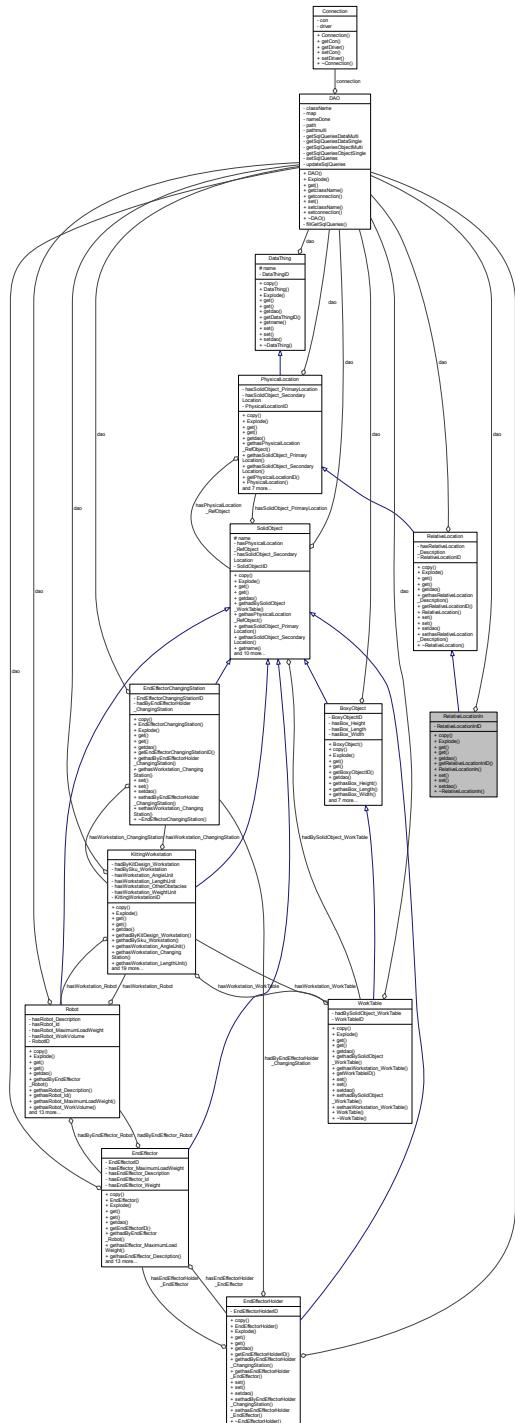
8.51 RelativeLocationIn Class Reference

```
#include <RelativeLocationIn.h>
```

Inheritance diagram for RelativeLocationIn:



Collaboration diagram for RelativeLocationIn:



Public Member Functions

- void **copy** (std::map< std::string, std::string > object)
 - std::vector< std::string > **Explode** (const std::string &str, char separator)

- void [get](#) (int id)
- void [get](#) (std::string name)
- DAO * [getdao](#) ()
- int [getRelativeLocationInID](#) ()
- RelativeLocationIn (std::string name)
- void [set](#) (int id, RelativeLocationIn *obj)
- void [set](#) (std::string name)
- void [setdao](#) (DAO *_dao)
- ~[RelativeLocationIn](#) ()

Private Attributes

- DAO * dao
- int [RelativeLocationInID](#)

Additional Inherited Members

8.51.1 Detailed Description

Definition at line 27 of file RelativeLocationIn.h.

8.51.2 Constructor & Destructor Documentation

8.51.2.1 RelativeLocationIn::RelativeLocationIn (std::string name)

Definition at line 19 of file RelativeLocationIn.cpp.

8.51.2.2 RelativeLocationIn::~RelativeLocationIn ()

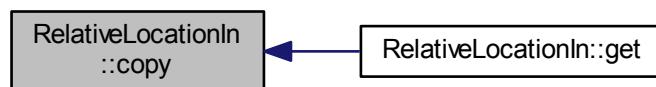
Definition at line 22 of file RelativeLocationIn.cpp.

8.51.3 Member Function Documentation

8.51.3.1 void RelativeLocationIn::copy (std::map< std::string, std::string > object)

Definition at line 68 of file RelativeLocationIn.cpp.

Here is the caller graph for this function:



8.51.3.2 `std::vector< std::string > RelativeLocationIn::Explode (const std::string & str, char separator)`

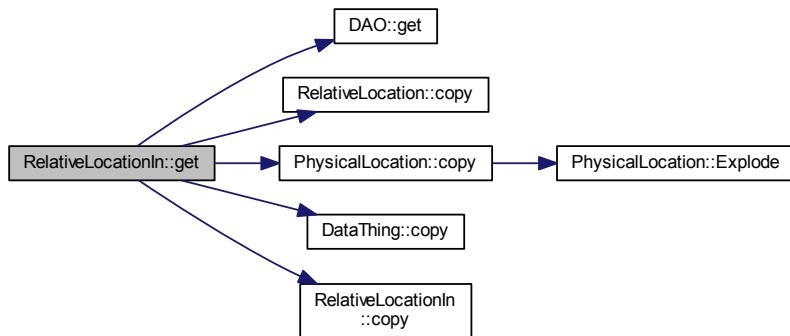
Definition at line 77 of file RelativeLocationIn.cpp.

8.51.3.3 `void RelativeLocationIn::get (int id)`

8.51.3.4 `void RelativeLocationIn::get (std::string name)`

Definition at line 34 of file RelativeLocationIn.cpp.

Here is the call graph for this function:



8.51.3.5 `DAO * RelativeLocationIn::getdao ()`

Definition at line 28 of file RelativeLocationIn.cpp.

8.51.3.6 `int RelativeLocationIn::getRelativeLocationInID ()`

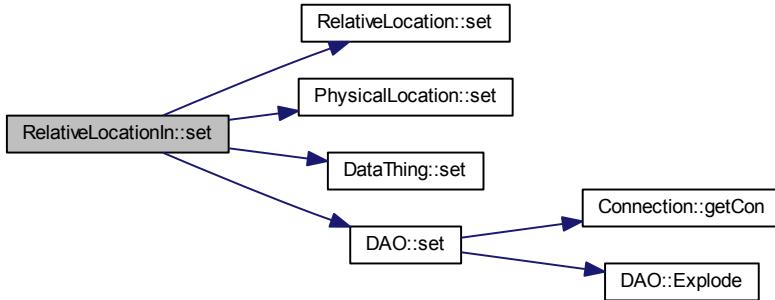
Definition at line 25 of file RelativeLocationIn.cpp.

8.51.3.7 `void RelativeLocationIn::set (int id, RelativeLocationIn * obj)`

8.51.3.8 `void RelativeLocationIn::set (std::string name)`

Definition at line 50 of file RelativeLocationIn.cpp.

Here is the call graph for this function:



8.51.3.9 void RelativeLocationIn::setdao (DAO * _dao)

Definition at line 31 of file `RelativeLocationIn.cpp`.

8.51.4 Member Data Documentation

8.51.4.1 DAO* RelativeLocationIn::dao [private]

Definition at line 29 of file `RelativeLocationIn.h`.

8.51.4.2 int RelativeLocationIn::RelativeLocationInID [private]

Definition at line 28 of file `RelativeLocationIn.h`.

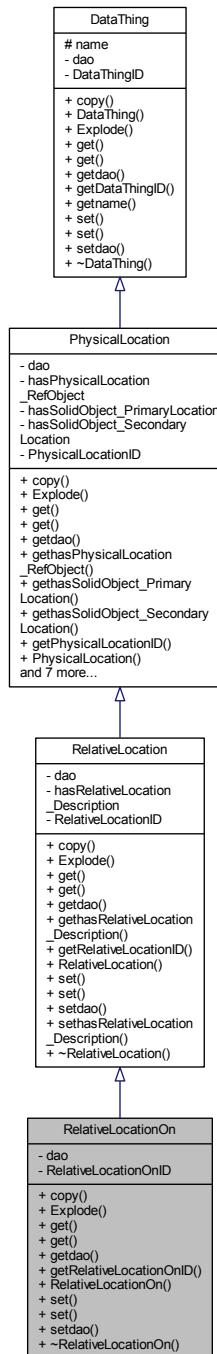
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[RelativeLocationIn.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[RelativeLocationIn.cpp](#)

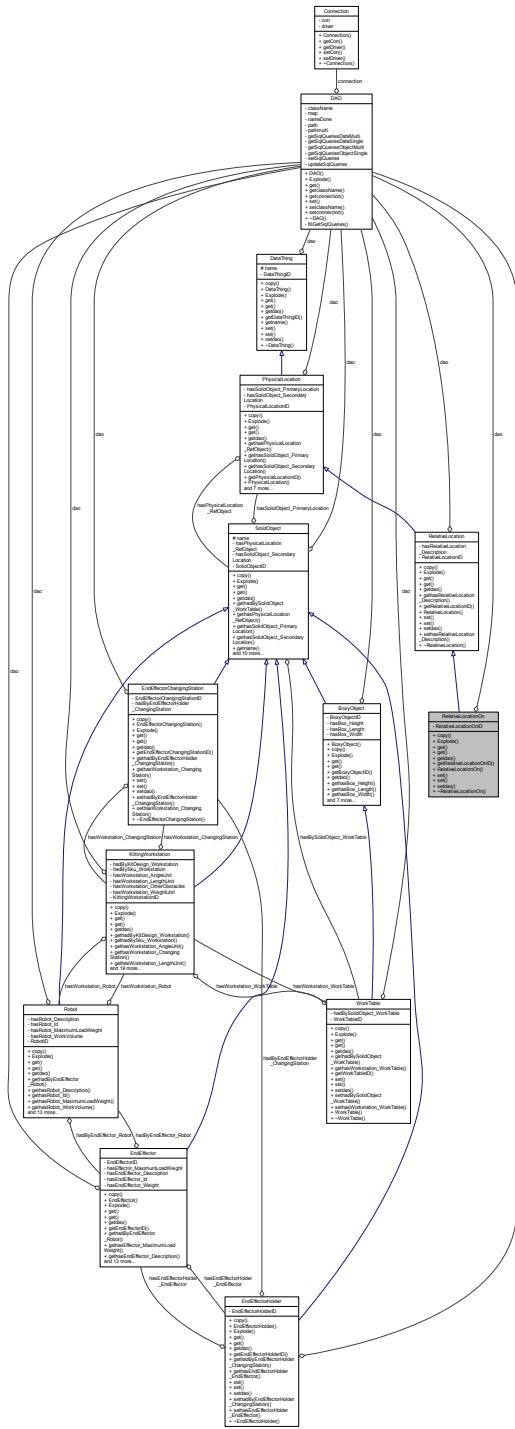
8.52 RelativeLocationOn Class Reference

```
#include <RelativeLocationOn.h>
```

Inheritance diagram for RelativeLocationOn:



Collaboration diagram for RelativeLocationOn:



Public Member Functions

- void **copy** (std::map< std::string, std::string > object)
 - std::vector< std::string > **Explode** (const std::string &str, char separator)

- void [get](#) (int id)
- void [get](#) (std::string name)
- DAO * [getdao](#) ()
- int [getRelativeLocationOnID](#) ()
- RelativeLocationOn [\(std::string name\)](#)
- void [set](#) (int id, RelativeLocationOn *obj)
- void [set](#) (std::string name)
- void [setdao](#) (DAO *_dao)
- ~[RelativeLocationOn](#) ()

Private Attributes

- DAO * dao
- int [RelativeLocationOnID](#)

Additional Inherited Members

8.52.1 Detailed Description

Definition at line 27 of file RelativeLocationOn.h.

8.52.2 Constructor & Destructor Documentation

8.52.2.1 RelativeLocationOn::RelativeLocationOn (std::string name)

Definition at line 19 of file RelativeLocationOn.cpp.

8.52.2.2 RelativeLocationOn::~RelativeLocationOn ()

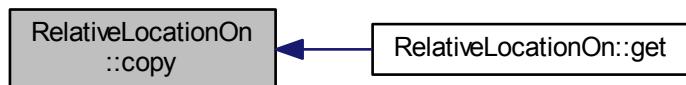
Definition at line 22 of file RelativeLocationOn.cpp.

8.52.3 Member Function Documentation

8.52.3.1 void RelativeLocationOn::copy (std::map< std::string, std::string > object)

Definition at line 68 of file RelativeLocationOn.cpp.

Here is the caller graph for this function:



8.52.3.2 `std::vector< std::string > RelativeLocationOn::Explode (const std::string & str, char separator)`

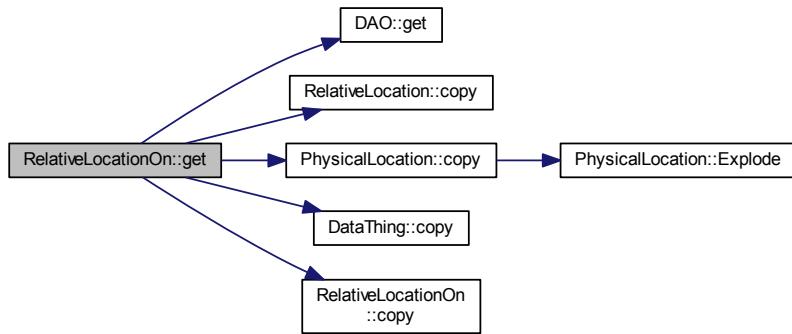
Definition at line 77 of file RelativeLocationOn.cpp.

8.52.3.3 `void RelativeLocationOn::get (int id)`

8.52.3.4 `void RelativeLocationOn::get (std::string name)`

Definition at line 34 of file RelativeLocationOn.cpp.

Here is the call graph for this function:



8.52.3.5 `DAO * RelativeLocationOn::getdao ()`

Definition at line 28 of file RelativeLocationOn.cpp.

8.52.3.6 `int RelativeLocationOn::getRelativeLocationOnID ()`

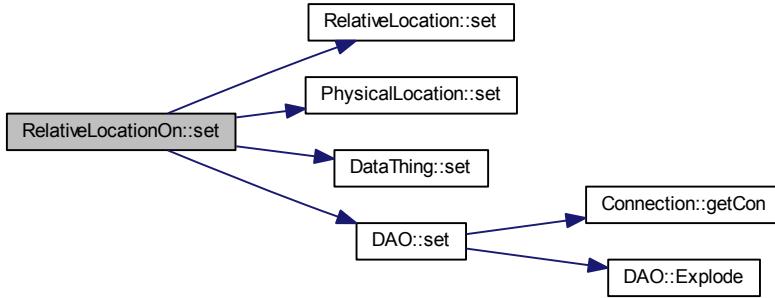
Definition at line 25 of file RelativeLocationOn.cpp.

8.52.3.7 `void RelativeLocationOn::set (int id, RelativeLocationOn * obj)`

8.52.3.8 `void RelativeLocationOn::set (std::string name)`

Definition at line 50 of file RelativeLocationOn.cpp.

Here is the call graph for this function:



8.52.3.9 void RelativeLocationOn::setdao (DAO * _dao)

Definition at line 31 of file `RelativeLocationOn.cpp`.

8.52.4 Member Data Documentation

8.52.4.1 DAO* RelativeLocationOn::dao [private]

Definition at line 29 of file `RelativeLocationOn.h`.

8.52.4.2 int RelativeLocationOn::RelativeLocationOnID [private]

Definition at line 28 of file `RelativeLocationOn.h`.

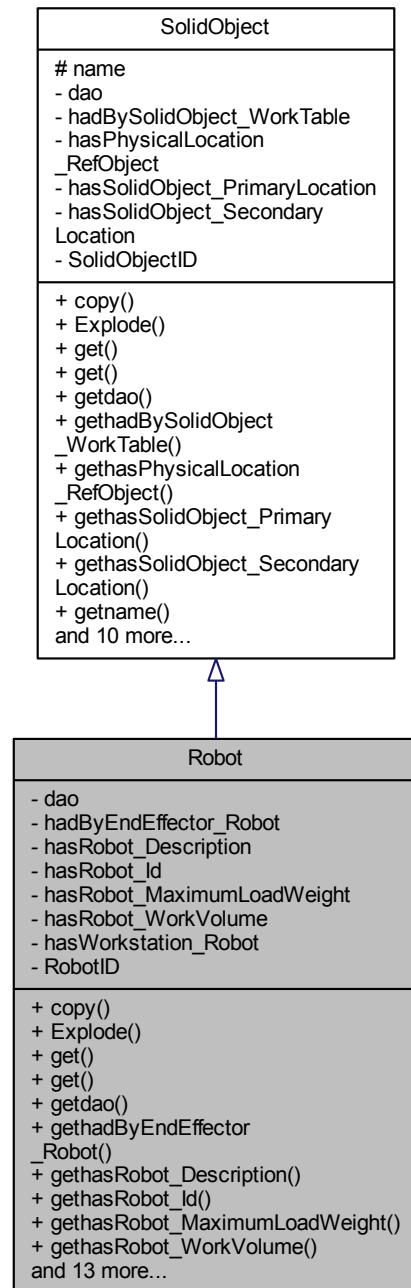
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[RelativeLocationOn.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[RelativeLocationOn.cpp](#)

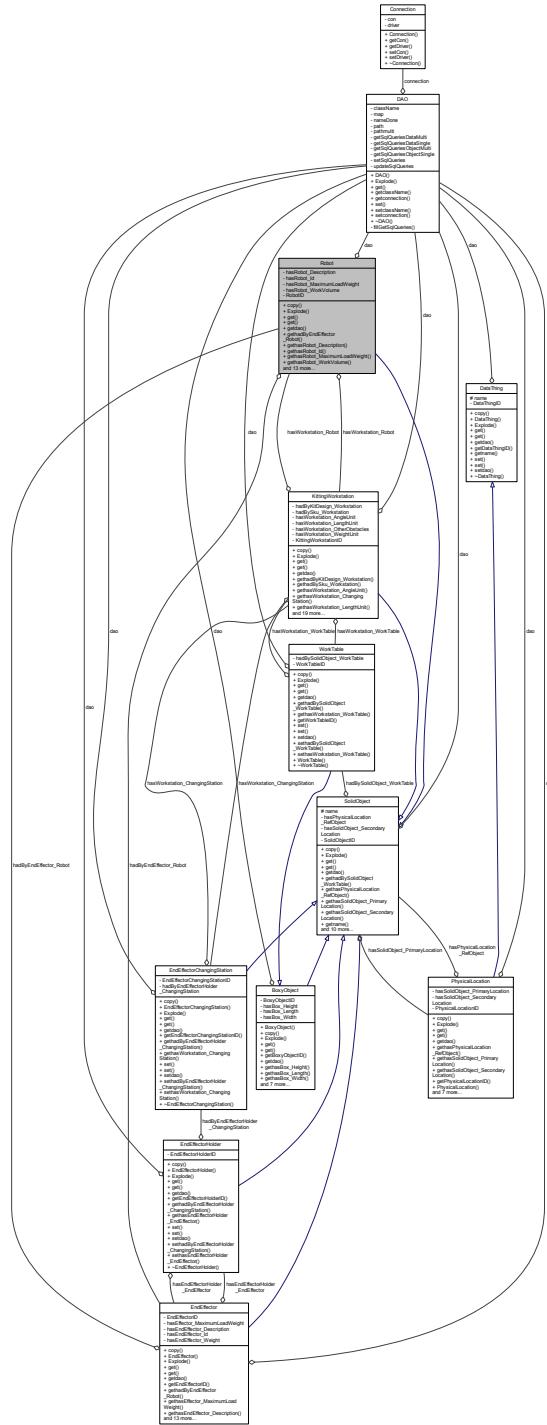
8.53 Robot Class Reference

```
#include <Robot.h>
```

Inheritance diagram for Robot:



Collaboration diagram for Robot:



Public Member Functions

- void **copy** (std::map< std::string, std::string > object)
 - std::vector< std::string > **Explode** (const std::string &str, char separator)

- void `get` (int id)
- void `get` (std::string name)
- DAO * `getdao` ()
- EndEffector * `gethadByEndEffector_Robot` ()
- std::string `gethasRobot_Description` ()
- std::string `gethasRobot_Id` ()
- double `gethasRobot_MaximumLoadWeight` ()
- std::vector< BoxVolume * > `gethasRobot_WorkVolume` ()
- KittingWorkstation * `gethasWorkstation_Robot` ()
- int `getRobotID` ()
- Robot (std::string name)
- void `set` (int id, Robot *obj)
- void `set` (std::string name)
- void `setdao` (DAO *_dao)
- void `sethadByEndEffector_Robot` (EndEffector *_hadByEndEffector_Robot)
- void `sethasRobot_Description` (std::string _hasRobot_Description)
- void `sethasRobot_Id` (std::string _hasRobot_Id)
- void `sethasRobot_MaximumLoadWeight` (double _hasRobot_MaximumLoadWeight)
- void `sethasRobot_WorkVolume` (std::vector< BoxVolume * > _hasRobot_WorkVolume)
- void `sethasWorkstation_Robot` (KittingWorkstation *_hasWorkstation_Robot)
- ~Robot ()

Private Attributes

- DAO * dao
- EndEffector * hadByEndEffector_Robot
- std::string hasRobot_Description
- std::string hasRobot_Id
- double hasRobot_MaximumLoadWeight
- std::vector< BoxVolume * > hasRobot_WorkVolume
- KittingWorkstation * hasWorkstation_Robot
- int RobotID

Additional Inherited Members

8.53.1 Detailed Description

Definition at line 30 of file Robot.h.

8.53.2 Constructor & Destructor Documentation

8.53.2.1 Robot::Robot (std::string name)

Definition at line 22 of file Robot.cpp.

8.53.2.2 Robot::~Robot ()

Definition at line 27 of file Robot.cpp.

8.53.3 Member Function Documentation

8.53.3.1 void Robot::copy (std::map< std::string, std::string > object)

Definition at line 118 of file Robot.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



8.53.3.2 std::vector< std::string > Robot::Explode (const std::string & str, char separator)

Definition at line 142 of file Robot.cpp.

Here is the caller graph for this function:

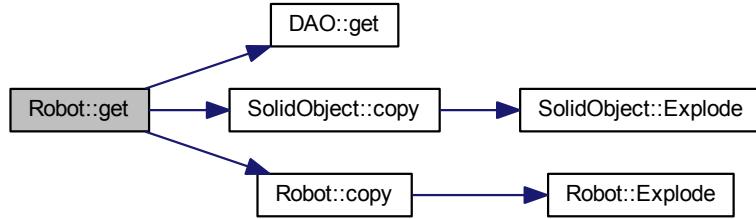


8.53.3.3 void Robot::get (int id)

8.53.3.4 void Robot::get (std::string name)

Definition at line 79 of file Robot.cpp.

Here is the call graph for this function:



8.53.3.5 `DAO * Robot::getdao()`

Definition at line 46 of file `Robot.cpp`.

8.53.3.6 `EndEffector * Robot::gethadByEndEffector_Robot()`

Definition at line 49 of file `Robot.cpp`.

8.53.3.7 `std::string Robot::gethasRobot_Description()`

Definition at line 34 of file `Robot.cpp`.

8.53.3.8 `std::string Robot::gethasRobot_Id()`

Definition at line 37 of file `Robot.cpp`.

8.53.3.9 `double Robot::gethasRobot_MaximumLoadWeight()`

Definition at line 40 of file `Robot.cpp`.

8.53.3.10 `std::vector< BoxVolume * > Robot::gethasRobot_WorkVolume()`

Definition at line 52 of file `Robot.cpp`.

8.53.3.11 `KittingWorkstation * Robot::gethasWorkstation_Robot()`

Definition at line 55 of file `Robot.cpp`.

8.53.3.12 `int Robot::getRobotID()`

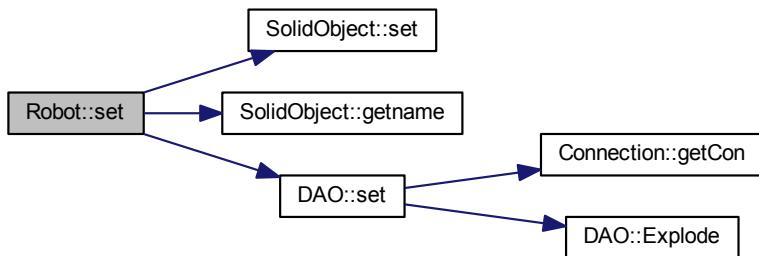
Definition at line 43 of file `Robot.cpp`.

8.53.3.13 void Robot::set (int *id*, Robot * *obj*)

8.53.3.14 void Robot::set (std::string *name*)

Definition at line 89 of file Robot.cpp.

Here is the call graph for this function:



8.53.3.15 void Robot::setdao (DAO * *_dao*)

Definition at line 67 of file Robot.cpp.

8.53.3.16 void Robot::sethadByEndEffector_Robot (EndEffector * *_hadByEndEffector_Robot*)

Definition at line 70 of file Robot.cpp.

8.53.3.17 void Robot::sethasRobot_Description (std::string *_hasRobot_Description*)

Definition at line 58 of file Robot.cpp.

8.53.3.18 void Robot::sethasRobot_Id (std::string *_hasRobot_Id*)

Definition at line 61 of file Robot.cpp.

8.53.3.19 void Robot::sethasRobot_MaximumLoadWeight (double *_hasRobot_MaximumLoadWeight*)

Definition at line 64 of file Robot.cpp.

8.53.3.20 void Robot::sethasRobot_WorkVolume (std::vector< BoxVolume * > *_hasRobot_WorkVolume*)

Definition at line 73 of file Robot.cpp.

8.53.3.21 void Robot::sethasWorkstation_Robot (KittingWorkstation * *_hasWorkstation_Robot*)

Definition at line 76 of file Robot.cpp.

8.53.4 Member Data Documentation

8.53.4.1 DAO* Robot::dao [private]

Definition at line 35 of file Robot.h.

8.53.4.2 EndEffector* Robot::hadByEndEffector_Robot [private]

Definition at line 36 of file Robot.h.

8.53.4.3 std::string Robot::hasRobot_Description [private]

Definition at line 31 of file Robot.h.

8.53.4.4 std::string Robot::hasRobot_Id [private]

Definition at line 32 of file Robot.h.

8.53.4.5 double Robot::hasRobot_MaximumLoadWeight [private]

Definition at line 33 of file Robot.h.

8.53.4.6 std::vector<BoxVolume*> Robot::hasRobot_WorkVolume [private]

Definition at line 37 of file Robot.h.

8.53.4.7 KittingWorkstation* Robot::hasWorkstation_Robot [private]

Definition at line 38 of file Robot.h.

8.53.4.8 int Robot::RobotID [private]

Definition at line 34 of file Robot.h.

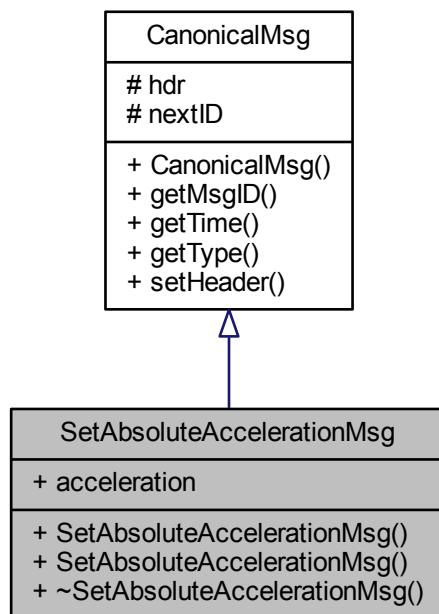
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[Robot.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[Robot.cpp](#)

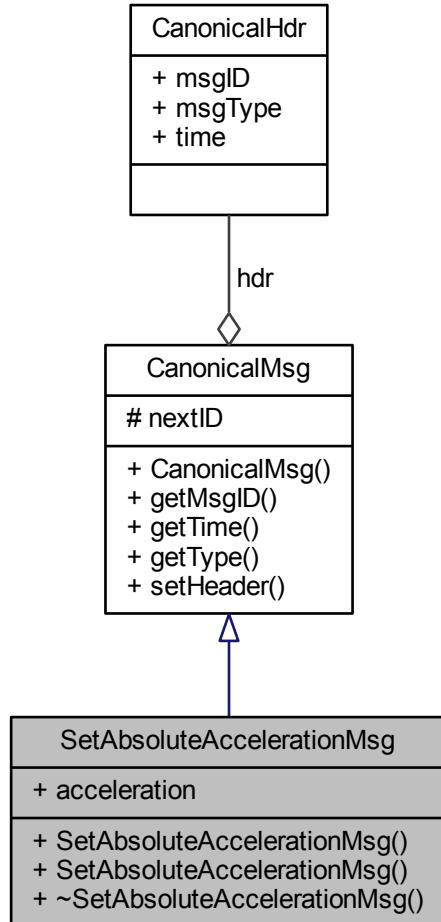
8.54 SetAbsoluteAccelerationMsg Class Reference

```
#include <canonicalMsg.hh>
```

Inheritance diagram for SetAbsoluteAccelerationMsg:



Collaboration diagram for SetAbsoluteAccelerationMsg:



Public Member Functions

- [SetAbsoluteAccelerationMsg \(\)](#)
- [SetAbsoluteAccelerationMsg \(double accelerationIn\)](#)
- [~SetAbsoluteAccelerationMsg \(\)](#)

Public Attributes

- double [acceleration](#)

Additional Inherited Members

8.54.1 Detailed Description

Definition at line 132 of file canonicalMsg.hh.

8.54.2 Constructor & Destructor Documentation

8.54.2.1 SetAbsoluteAccelerationMsg::SetAbsoluteAccelerationMsg () [inline]

Definition at line 134 of file canonicalMsg.hh.

8.54.2.2 SetAbsoluteAccelerationMsg::SetAbsoluteAccelerationMsg (double accelerationIn) [inline]

Definition at line 135 of file canonicalMsg.hh.

8.54.2.3 SetAbsoluteAccelerationMsg::~SetAbsoluteAccelerationMsg () [inline]

Definition at line 136 of file canonicalMsg.hh.

8.54.3 Member Data Documentation

8.54.3.1 double SetAbsoluteAccelerationMsg::acceleration

Definition at line 136 of file canonicalMsg.hh.

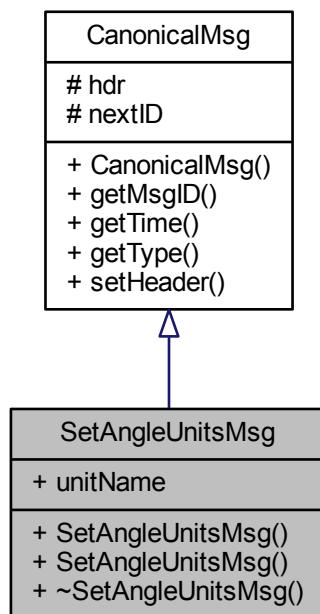
The documentation for this class was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[canonicalMsg.hh](#)

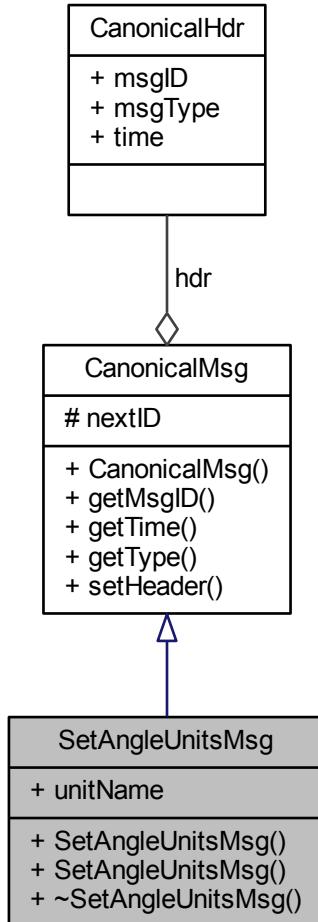
8.55 SetAngleUnitsMsg Class Reference

```
#include <canonicalMsg.hh>
```

Inheritance diagram for SetAngleUnitsMsg:



Collaboration diagram for SetAngleUnitsMsg:



Public Member Functions

- `SetAngleUnitsMsg ()`
- `SetAngleUnitsMsg (std::string unitNameIn)`
- `~SetAngleUnitsMsg ()`

Public Attributes

- `std::string unitName`

Additional Inherited Members

8.55.1 Detailed Description

Definition at line 140 of file canonicalMsg.hh.

8.55.2 Constructor & Destructor Documentation

8.55.2.1 SetAngleUnitsMsg::SetAngleUnitsMsg () [inline]

Definition at line 142 of file canonicalMsg.hh.

8.55.2.2 SetAngleUnitsMsg::SetAngleUnitsMsg (std::string *unitNameIn*) [inline]

Definition at line 143 of file canonicalMsg.hh.

8.55.2.3 SetAngleUnitsMsg::~SetAngleUnitsMsg () [inline]

Definition at line 144 of file canonicalMsg.hh.

8.55.3 Member Data Documentation

8.55.3.1 std::string SetAngleUnitsMsg::unitName

Definition at line 144 of file canonicalMsg.hh.

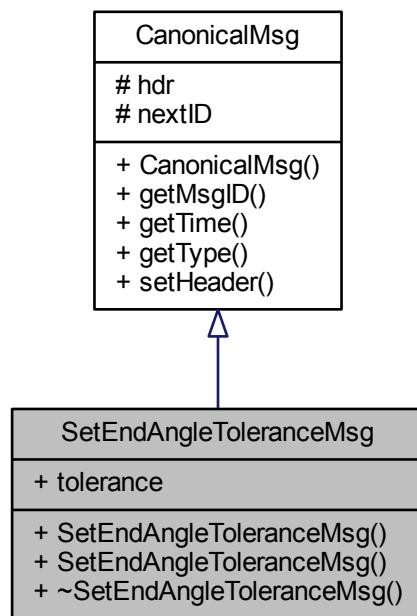
The documentation for this class was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[canonicalMsg.hh](#)

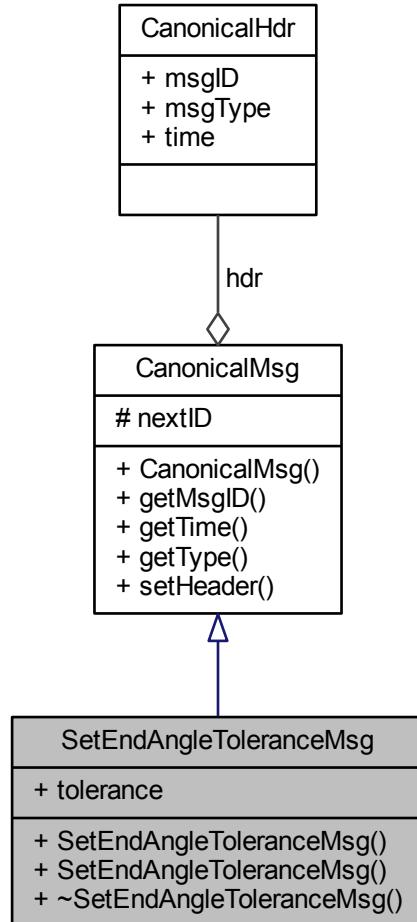
8.56 SetEndAngleToleranceMsg Class Reference

```
#include <canonicalMsg.hh>
```

Inheritance diagram for SetEndAngleToleranceMsg:



Collaboration diagram for SetEndAngleToleranceMsg:



Public Member Functions

- `SetEndAngleToleranceMsg ()`
- `SetEndAngleToleranceMsg (double toleranceIn)`
- `~SetEndAngleToleranceMsg ()`

Public Attributes

- double `tolerance`

Additional Inherited Members

8.56.1 Detailed Description

Definition at line 148 of file canonicalMsg.hh.

8.56.2 Constructor & Destructor Documentation

8.56.2.1 SetEndAngleToleranceMsg::SetEndAngleToleranceMsg() [inline]

Definition at line 150 of file canonicalMsg.hh.

8.56.2.2 SetEndAngleToleranceMsg::SetEndAngleToleranceMsg(double toleranceIn) [inline]

Definition at line 151 of file canonicalMsg.hh.

8.56.2.3 SetEndAngleToleranceMsg::~SetEndAngleToleranceMsg() [inline]

Definition at line 152 of file canonicalMsg.hh.

8.56.3 Member Data Documentation

8.56.3.1 double SetEndAngleToleranceMsg::tolerance

Definition at line 152 of file canonicalMsg.hh.

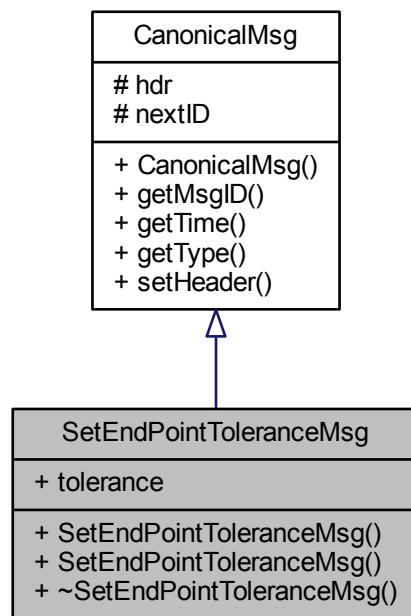
The documentation for this class was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[canonicalMsg.hh](#)

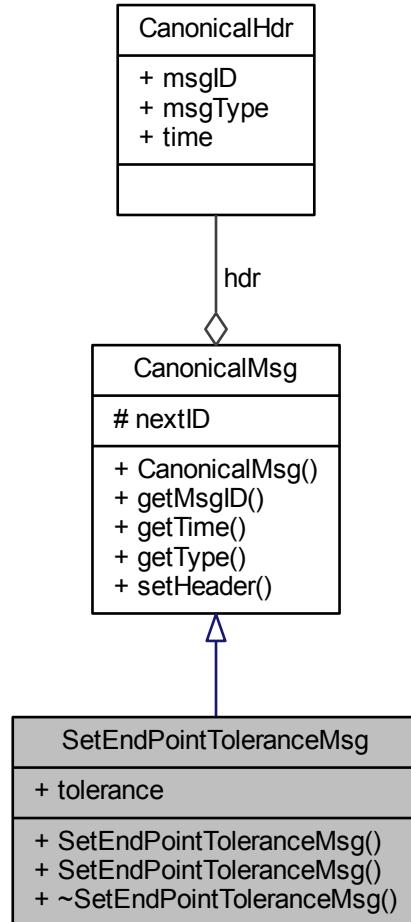
8.57 SetEndPointToleranceMsg Class Reference

```
#include <canonicalMsg.hh>
```

Inheritance diagram for SetEndPointToleranceMsg:



Collaboration diagram for SetEndPointToleranceMsg:



Public Member Functions

- `SetEndPointToleranceMsg ()`
- `SetEndPointToleranceMsg (double toleranceIn)`
- `~SetEndPointToleranceMsg ()`

Public Attributes

- double `tolerance`

Additional Inherited Members

8.57.1 Detailed Description

Definition at line 156 of file canonicalMsg.hh.

8.57.2 Constructor & Destructor Documentation

8.57.2.1 SetEndPointToleranceMsg::SetEndPointToleranceMsg () [inline]

Definition at line 158 of file canonicalMsg.hh.

8.57.2.2 SetEndPointToleranceMsg::SetEndPointToleranceMsg (double toleranceIn) [inline]

Definition at line 159 of file canonicalMsg.hh.

8.57.2.3 SetEndPointToleranceMsg::~SetEndPointToleranceMsg () [inline]

Definition at line 160 of file canonicalMsg.hh.

8.57.3 Member Data Documentation

8.57.3.1 double SetEndPointToleranceMsg::tolerance

Definition at line 160 of file canonicalMsg.hh.

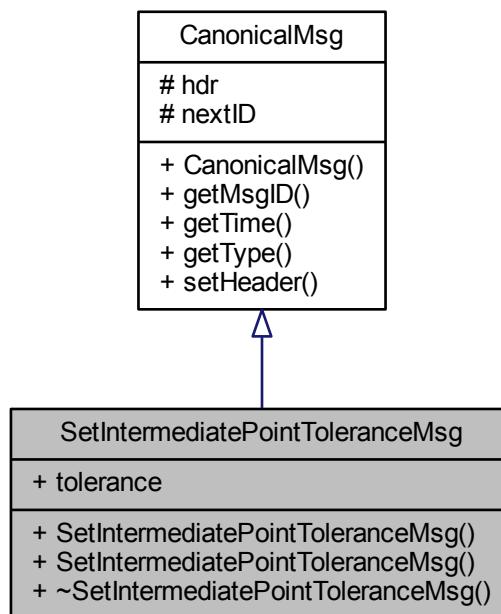
The documentation for this class was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[canonicalMsg.hh](#)

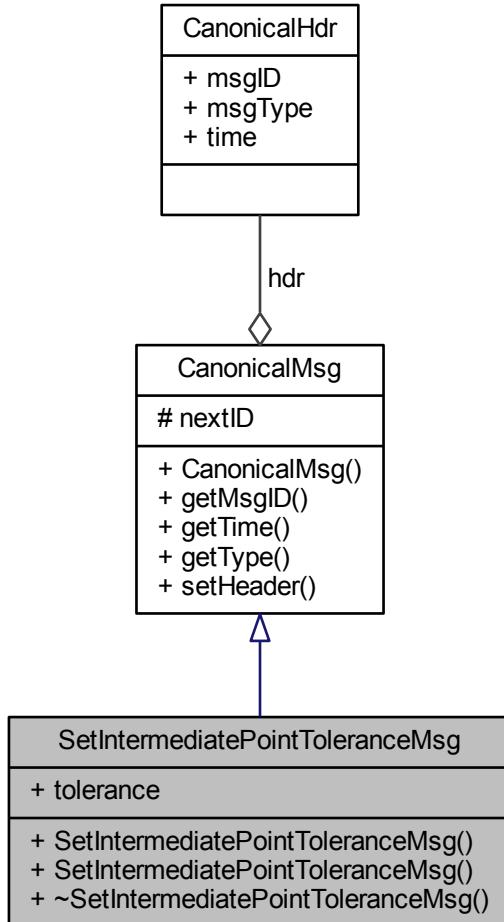
8.58 SetIntermediatePointToleranceMsg Class Reference

```
#include <canonicalMsg.hh>
```

Inheritance diagram for SetIntermediatePointToleranceMsg:



Collaboration diagram for SetIntermediatePointToleranceMsg:



Public Member Functions

- [SetIntermediatePointToleranceMsg \(\)](#)
- [SetIntermediatePointToleranceMsg \(double toleranceIn\)](#)
- [~SetIntermediatePointToleranceMsg \(\)](#)

Public Attributes

- double [tolerance](#)

Additional Inherited Members

8.58.1 Detailed Description

Definition at line 164 of file canonicalMsg.hh.

8.58.2 Constructor & Destructor Documentation

8.58.2.1 SetIntermediatePointToleranceMsg::SetIntermediatePointToleranceMsg () [inline]

Definition at line 166 of file canonicalMsg.hh.

8.58.2.2 SetIntermediatePointToleranceMsg::SetIntermediatePointToleranceMsg (double toleranceIn) [inline]

Definition at line 167 of file canonicalMsg.hh.

8.58.2.3 SetIntermediatePointToleranceMsg::~SetIntermediatePointToleranceMsg () [inline]

Definition at line 168 of file canonicalMsg.hh.

8.58.3 Member Data Documentation

8.58.3.1 double SetIntermediatePointToleranceMsg::tolerance

Definition at line 168 of file canonicalMsg.hh.

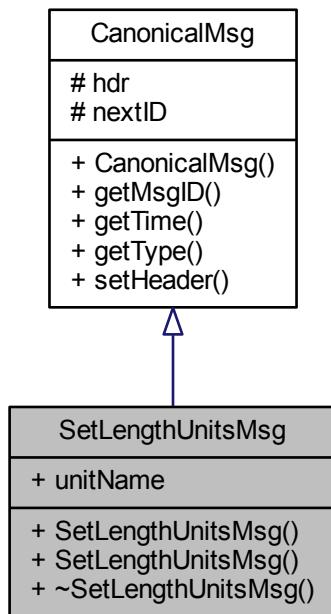
The documentation for this class was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[canonicalMsg.hh](#)

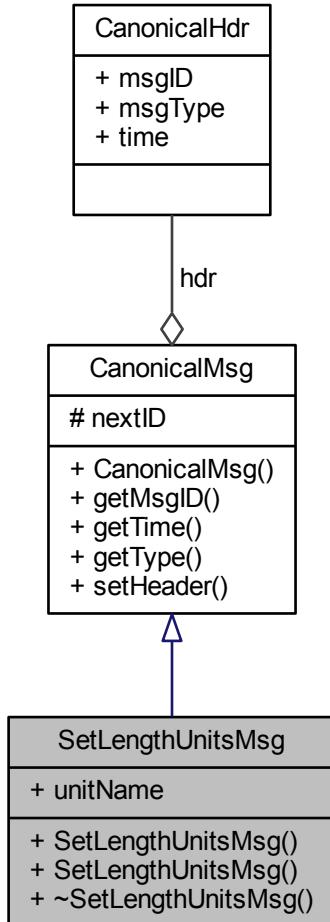
8.59 SetLengthUnitsMsg Class Reference

```
#include <canonicalMsg.hh>
```

Inheritance diagram for SetLengthUnitsMsg:



Collaboration diagram for SetLengthUnitsMsg:



Public Member Functions

- [SetLengthUnitsMsg \(\)](#)
- [SetLengthUnitsMsg \(std::string unitNameIn\)](#)
- [~SetLengthUnitsMsg \(\)](#)

Public Attributes

- std::string [unitName](#)

Additional Inherited Members

8.59.1 Detailed Description

Definition at line 172 of file canonicalMsg.hh.

8.59.2 Constructor & Destructor Documentation

8.59.2.1 SetLengthUnitsMsg::SetLengthUnitsMsg () [inline]

Definition at line 174 of file canonicalMsg.hh.

8.59.2.2 SetLengthUnitsMsg::SetLengthUnitsMsg (std::string *unitNameIn*) [inline]

Definition at line 175 of file canonicalMsg.hh.

8.59.2.3 SetLengthUnitsMsg::~SetLengthUnitsMsg () [inline]

Definition at line 176 of file canonicalMsg.hh.

8.59.3 Member Data Documentation

8.59.3.1 std::string SetLengthUnitsMsg::unitName

Definition at line 176 of file canonicalMsg.hh.

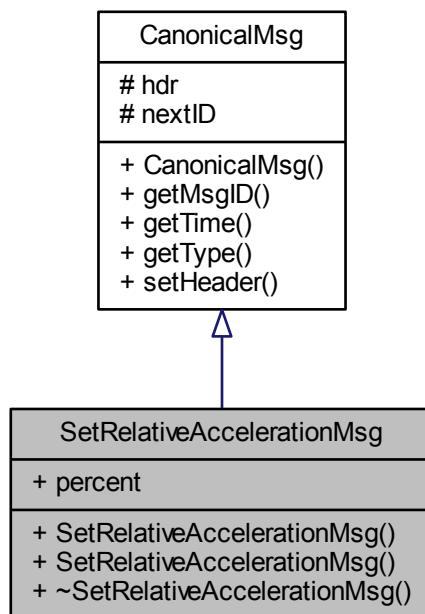
The documentation for this class was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[canonicalMsg.hh](#)

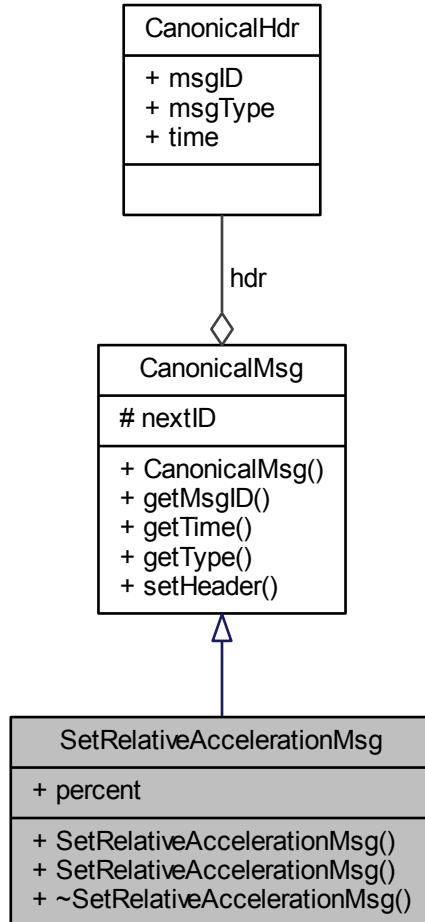
8.60 SetRelativeAccelerationMsg Class Reference

```
#include <canonicalMsg.hh>
```

Inheritance diagram for SetRelativeAccelerationMsg:



Collaboration diagram for SetRelativeAccelerationMsg:



Public Member Functions

- [SetRelativeAccelerationMsg \(\)](#)
- [SetRelativeAccelerationMsg \(double percentIn\)](#)
- [~SetRelativeAccelerationMsg \(\)](#)

Public Attributes

- double [percent](#)

Additional Inherited Members

8.60.1 Detailed Description

Definition at line 180 of file canonicalMsg.hh.

8.60.2 Constructor & Destructor Documentation

8.60.2.1 SetRelativeAccelerationMsg::SetRelativeAccelerationMsg() [inline]

Definition at line 182 of file canonicalMsg.hh.

8.60.2.2 SetRelativeAccelerationMsg::SetRelativeAccelerationMsg(double percentIn) [inline]

Definition at line 183 of file canonicalMsg.hh.

8.60.2.3 SetRelativeAccelerationMsg::~SetRelativeAccelerationMsg() [inline]

Definition at line 184 of file canonicalMsg.hh.

8.60.3 Member Data Documentation

8.60.3.1 double SetRelativeAccelerationMsg::percent

Definition at line 184 of file canonicalMsg.hh.

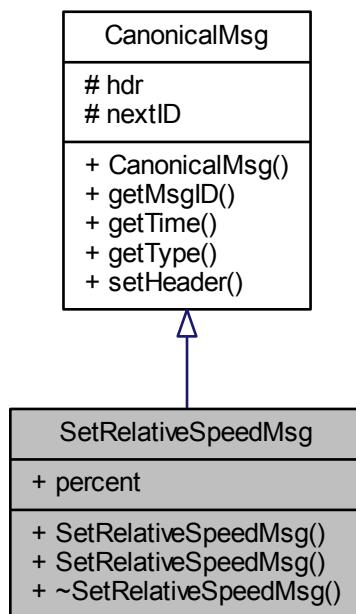
The documentation for this class was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[canonicalMsg.hh](#)

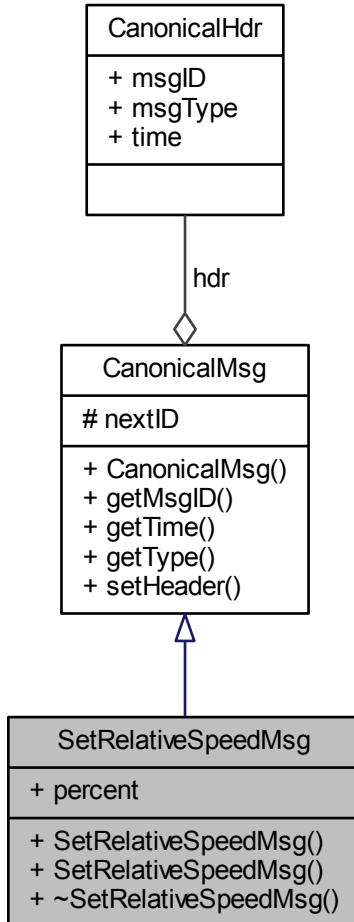
8.61 SetRelativeSpeedMsg Class Reference

```
#include <canonicalMsg.hh>
```

Inheritance diagram for SetRelativeSpeedMsg:



Collaboration diagram for SetRelativeSpeedMsg:



Public Member Functions

- [SetRelativeSpeedMsg \(\)](#)
- [SetRelativeSpeedMsg \(double percentIn\)](#)
- [~SetRelativeSpeedMsg \(\)](#)

Public Attributes

- double [percent](#)

Additional Inherited Members

8.61.1 Detailed Description

Definition at line 188 of file canonicalMsg.hh.

8.61.2 Constructor & Destructor Documentation

8.61.2.1 SetRelativeSpeedMsg::SetRelativeSpeedMsg () [inline]

Definition at line 190 of file canonicalMsg.hh.

8.61.2.2 SetRelativeSpeedMsg::SetRelativeSpeedMsg (double percentIn) [inline]

Definition at line 191 of file canonicalMsg.hh.

8.61.2.3 SetRelativeSpeedMsg::~SetRelativeSpeedMsg () [inline]

Definition at line 192 of file canonicalMsg.hh.

8.61.3 Member Data Documentation

8.61.3.1 double SetRelativeSpeedMsg::percent

Definition at line 192 of file canonicalMsg.hh.

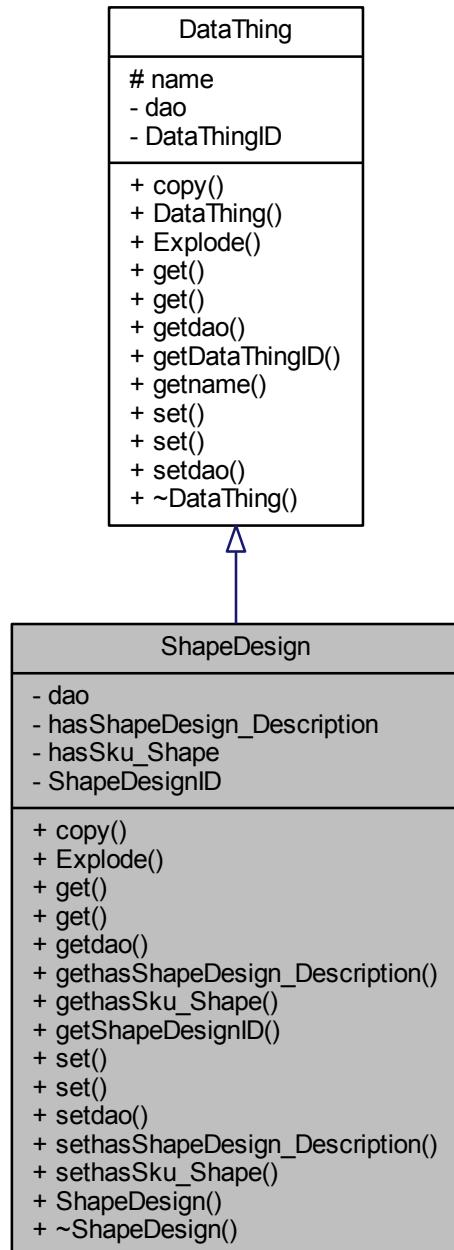
The documentation for this class was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[canonicalMsg.hh](#)

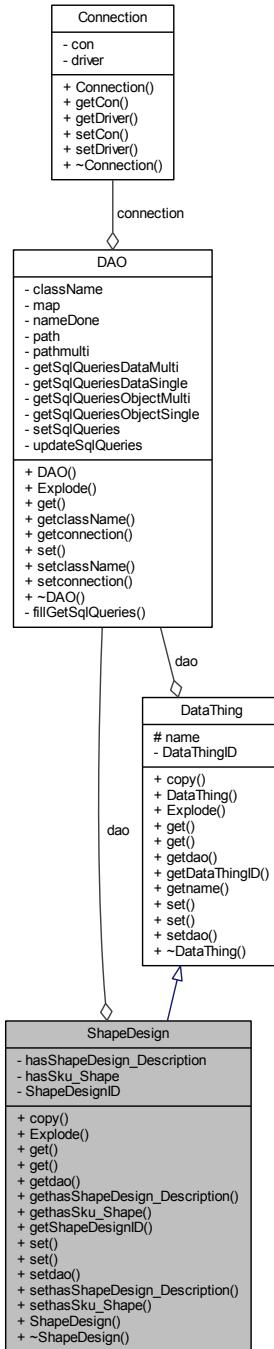
8.62 ShapeDesign Class Reference

```
#include <ShapeDesign.h>
```

Inheritance diagram for ShapeDesign:



Collaboration diagram for ShapeDesign:



Public Member Functions

- void `copy` (std::map< std::string, std::string > object)
- std::vector< std::string > `Explode` (const std::string &str, char separator)

- void [get](#) (int id)
- void [get](#) (std::string name)
- DAO * [getdao](#) ()
- std::string [gethasShapeDesign_Description](#) ()
- std::vector< StockKeepingUnit * > [gethasSku_Shape](#) ()
- int [getShapeDesignID](#) ()
- void [set](#) (int id, ShapeDesign *obj)
- void [set](#) (std::string name)
- void [setdao](#) (DAO *_dao)
- void [sethasShapeDesign_Description](#) (std::string _hasShapeDesign_Description)
- void [sethasSku_Shape](#) (std::vector< StockKeepingUnit * > _hasSku_Shape)
- ShapeDesign (std::string name)
- ~ShapeDesign ()

Private Attributes

- DAO * dao
- std::string hasShapeDesign_Description
- std::vector< StockKeepingUnit * > hasSku_Shape
- int ShapeDesignID

Additional Inherited Members

8.62.1 Detailed Description

Definition at line 28 of file ShapeDesign.h.

8.62.2 Constructor & Destructor Documentation

8.62.2.1 ShapeDesign::ShapeDesign (std::string name)

Definition at line 20 of file ShapeDesign.cpp.

8.62.2.2 ShapeDesign::~ShapeDesign ()

Definition at line 23 of file ShapeDesign.cpp.

8.62.3 Member Function Documentation

8.62.3.1 void ShapeDesign::copy (std::map< std::string, std::string > object)

Definition at line 80 of file ShapeDesign.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



8.62.3.2 std::vector< std::string > ShapeDesign::Explode (const std::string & str, char separator)

Definition at line 96 of file ShapeDesign.cpp.

Here is the caller graph for this function:

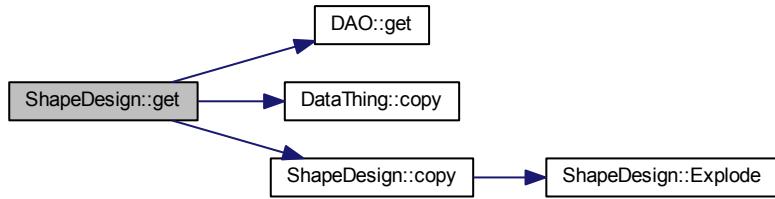


8.62.3.3 void ShapeDesign::get (int id)

8.62.3.4 void ShapeDesign::get (std::string name)

Definition at line 49 of file ShapeDesign.cpp.

Here is the call graph for this function:



8.62.3.5 `DAO * ShapeDesign::getdao()`

Definition at line 34 of file `ShapeDesign.cpp`.

8.62.3.6 `std::string ShapeDesign::gethasShapeDesign_Description()`

Definition at line 28 of file `ShapeDesign.cpp`.

8.62.3.7 `std::vector< StockKeepingUnit * > ShapeDesign::gethasSku_Shape()`

Definition at line 37 of file `ShapeDesign.cpp`.

8.62.3.8 `int ShapeDesign::getShapeDesignID()`

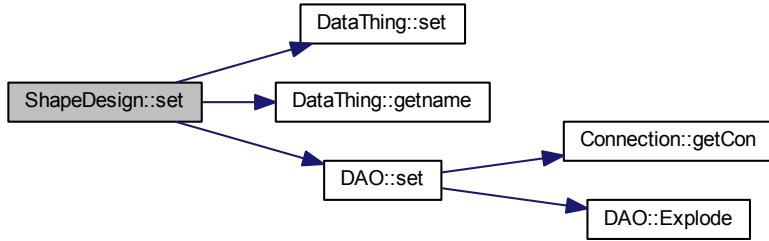
Definition at line 31 of file `ShapeDesign.cpp`.

8.62.3.9 `void ShapeDesign::set(int id, ShapeDesign * obj)`

8.62.3.10 `void ShapeDesign::set(std::string name)`

Definition at line 59 of file `ShapeDesign.cpp`.

Here is the call graph for this function:



8.62.3.11 void ShapeDesign::setdao (DAO * _dao)

Definition at line 43 of file ShapeDesign.cpp.

8.62.3.12 void ShapeDesign::sethasShapeDesign_Description (std::string _hasShapeDesign_Description)

Definition at line 40 of file ShapeDesign.cpp.

8.62.3.13 void ShapeDesign::sethasSku_Shape (std::vector< StockKeepingUnit * > _hasSku_Shape)

Definition at line 46 of file ShapeDesign.cpp.

8.62.4 Member Data Documentation

8.62.4.1 DAO* ShapeDesign::dao [private]

Definition at line 31 of file ShapeDesign.h.

8.62.4.2 std::string ShapeDesign::hasShapeDesign_Description [private]

Definition at line 29 of file ShapeDesign.h.

8.62.4.3 std::vector<StockKeepingUnit*> ShapeDesign::hasSku_Shape [private]

Definition at line 32 of file ShapeDesign.h.

8.62.4.4 int ShapeDesign::ShapeDesignID [private]

Definition at line 30 of file ShapeDesign.h.

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

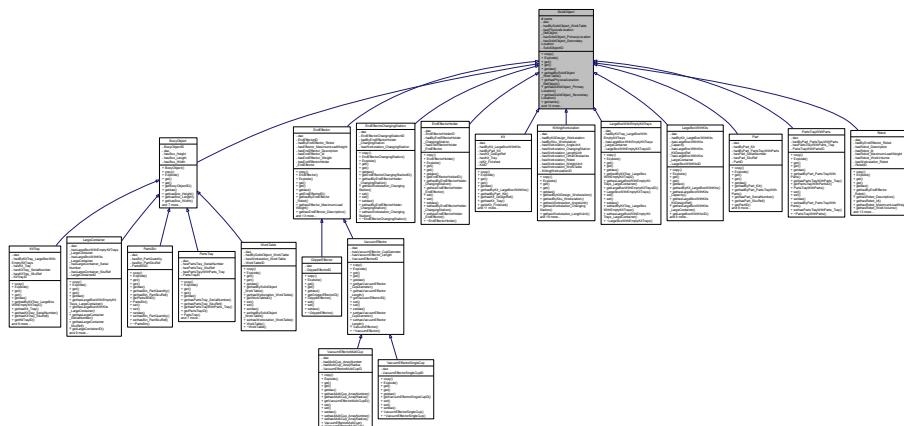
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[ShapeDesign.h](#)

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[ShapeDesign.cpp](#)

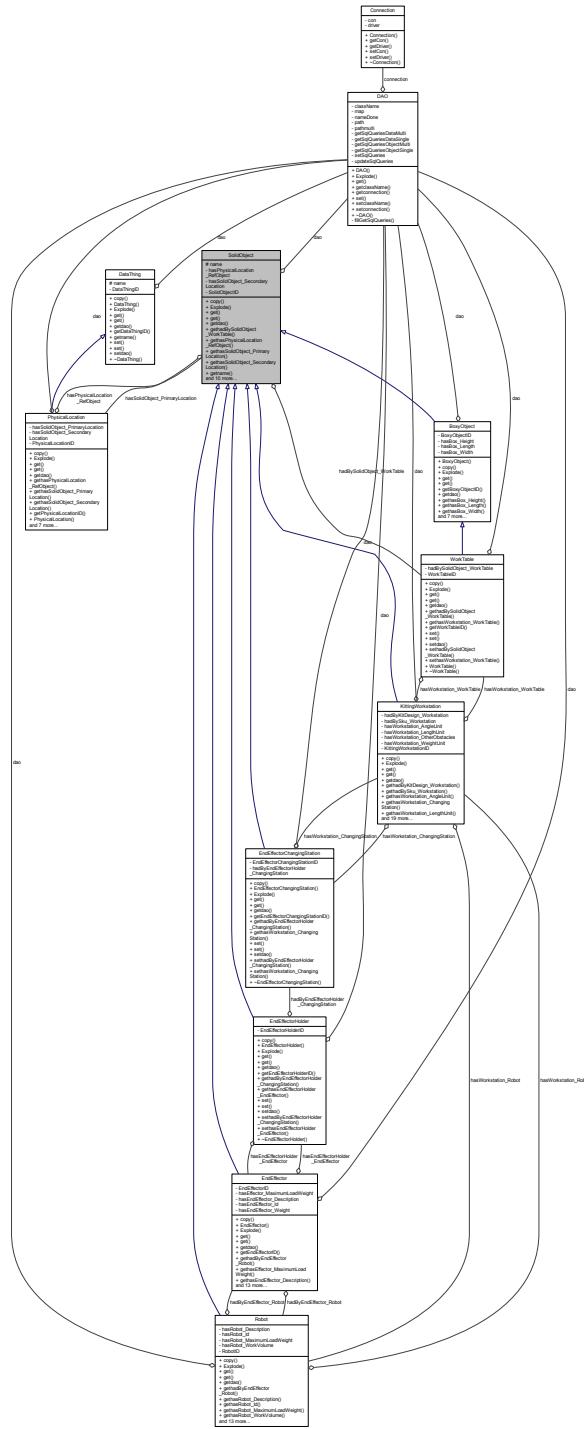
8.63 SolidObject Class Reference

```
#include <SolidObject.h>
```

Inheritance diagram for SolidObject:



Collaboration diagram for SolidObject:



Public Member Functions

- void `copy` (std::map< std::string, std::string > object)
 - std::vector< std::string > `Explode` (const std::string &str, char separator)

- void `get` (int id)
- void `get` (std::string name)
- DAO * `getdao` ()
- WorkTable * `gethadBySolidObject_WorkTable` ()
- std::vector< PhysicalLocation * > `gethasPhysicalLocation_RefObject` ()
- PhysicalLocation * `gethasSolidObject_PrimaryLocation` ()
- std::vector< PhysicalLocation * > `gethasSolidObject_SecondaryLocation` ()
- std::string `getname` ()
- int `getSolidObjectID` ()
- void `set` (int id, SolidObject *obj)
- void `set` (std::string name)
- void `setdao` (DAO *_dao)
- void `sethadBySolidObject_WorkTable` (WorkTable *_hadBySolidObject_WorkTable)
- void `sethasPhysicalLocation_RefObject` (std::vector< PhysicalLocation * > _hasPhysicalLocation_RefObject)
- void `sethasSolidObject_PrimaryLocation` (PhysicalLocation *_hasSolidObject_PrimaryLocation)
- void `sethasSolidObject_SecondaryLocation` (std::vector< PhysicalLocation * > _hasSolidObject_SecondaryLocation)
- SolidObject (std::string name)
- ~SolidObject ()

Protected Attributes

- std::string name

Private Attributes

- DAO * dao
- WorkTable * hadBySolidObject_WorkTable
- std::vector< PhysicalLocation * > hasPhysicalLocation_RefObject
- PhysicalLocation * hasSolidObject_PrimaryLocation
- std::vector< PhysicalLocation * > hasSolidObject_SecondaryLocation
- int SolidObjectID

8.63.1 Detailed Description

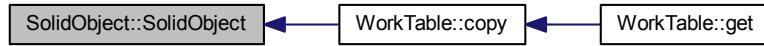
Definition at line 28 of file SolidObject.h.

8.63.2 Constructor & Destructor Documentation

8.63.2.1 SolidObject::SolidObject (std::string name)

Definition at line 21 of file SolidObject.cpp.

Here is the caller graph for this function:



8.63.2.2 SolidObject::~SolidObject ()

Definition at line 26 of file SolidObject.cpp.

8.63.3 Member Function Documentation

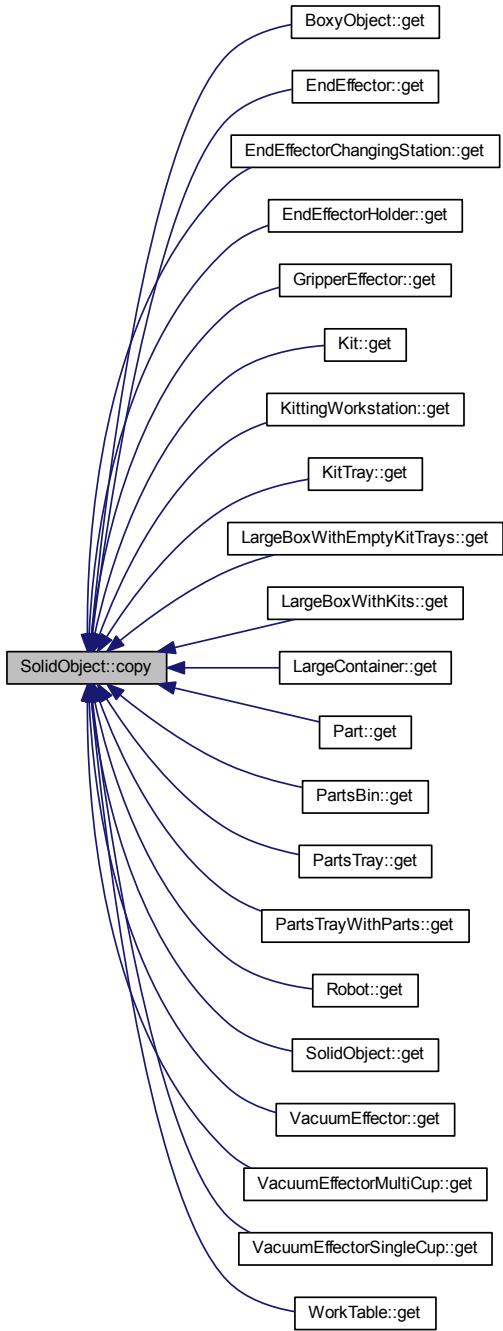
8.63.3.1 void SolidObject::copy (std::map< std::string, std::string > object)

Definition at line 106 of file SolidObject.cpp.

Here is the call graph for this function:



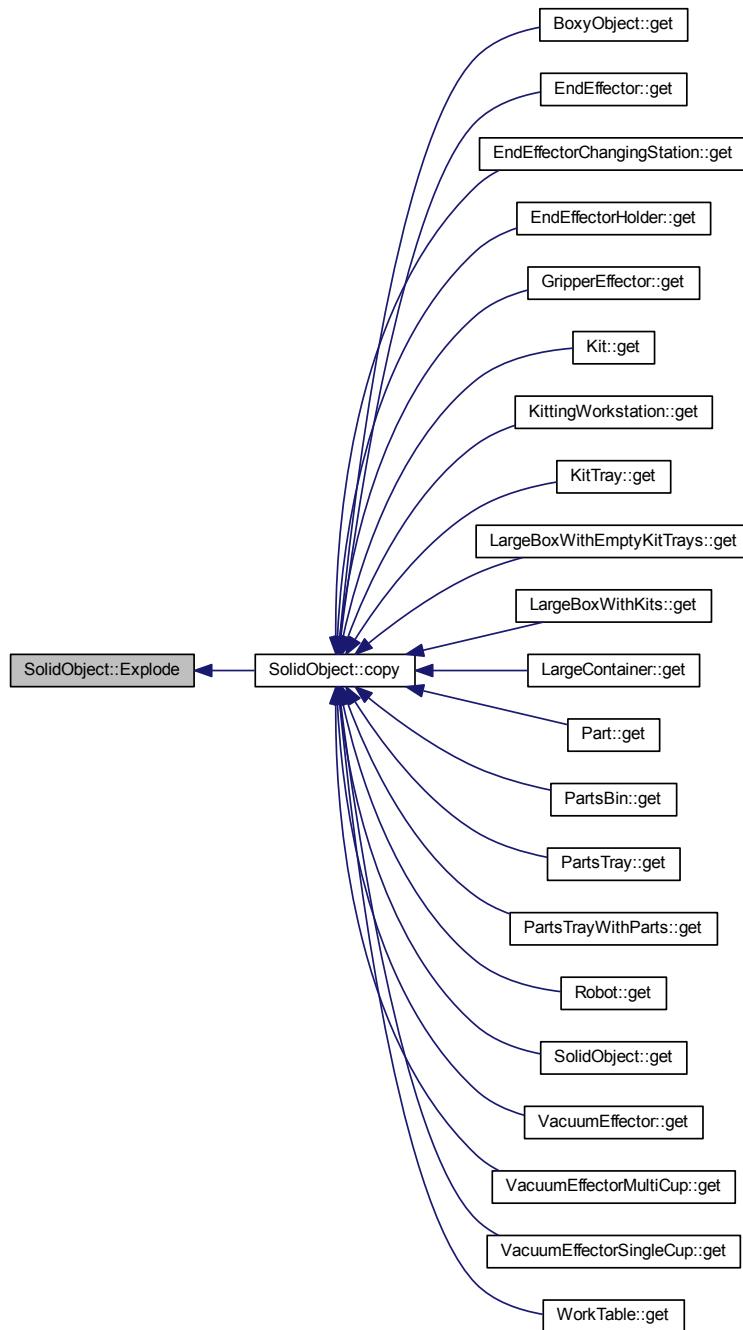
Here is the caller graph for this function:



8.63.3.2 `std::vector< std::string > SolidObject::Explode (const std::string & str, char separator)`

Definition at line 133 of file SolidObject.cpp.

Here is the caller graph for this function:

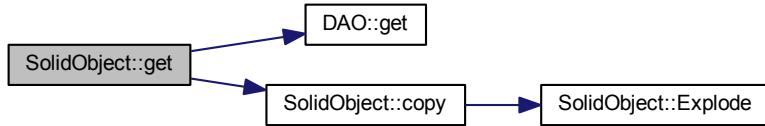


8.63.3.3 void SolidObject::get (int id)

8.63.3.4 void SolidObject::get (std::string name)

Definition at line 71 of file SolidObject.cpp.

Here is the call graph for this function:



8.63.3.5 DAO * SolidObject::getdao ()

Definition at line 38 of file SolidObject.cpp.

8.63.3.6 WorkTable * SolidObject::gethadBySolidObject_WorkTable ()

Definition at line 47 of file SolidObject.cpp.

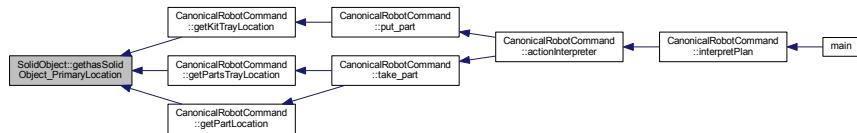
8.63.3.7 std::vector< PhysicalLocation * > SolidObject::gethasPhysicalLocation_RefObject ()

Definition at line 50 of file SolidObject.cpp.

8.63.3.8 PhysicalLocation * SolidObject::gethasSolidObject_PrimaryLocation ()

Definition at line 44 of file SolidObject.cpp.

Here is the caller graph for this function:



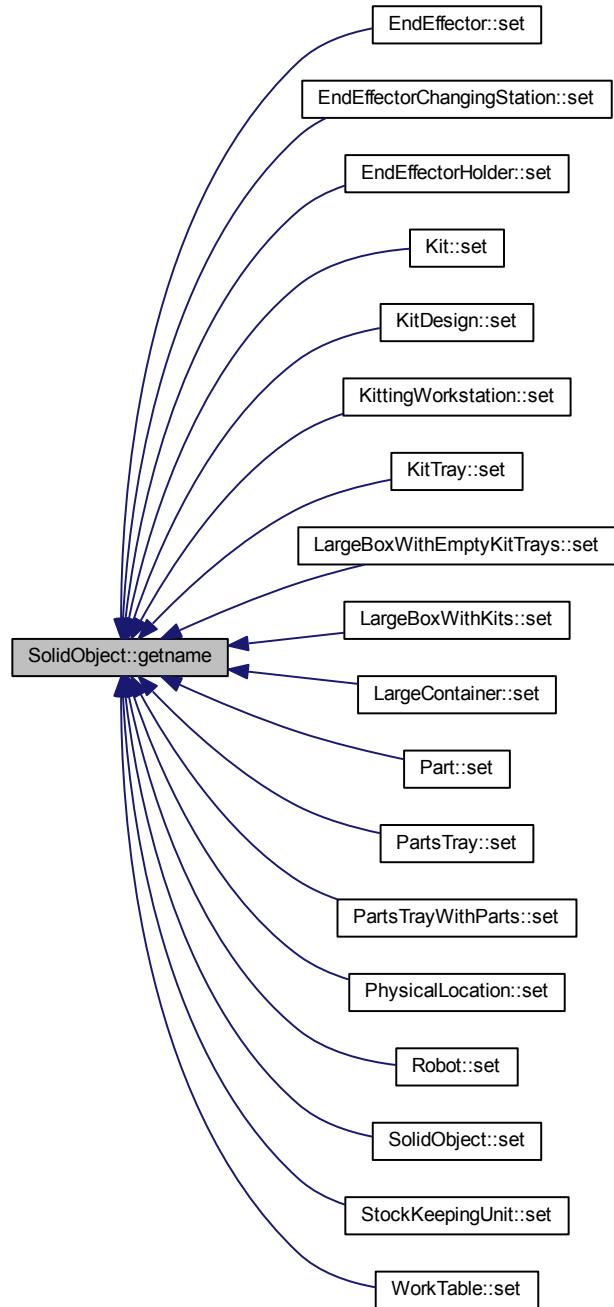
8.63.3.9 std::vector< PhysicalLocation * > SolidObject::gethasSolidObject_SecondaryLocation ()

Definition at line 41 of file SolidObject.cpp.

8.63.3.10 std::string SolidObject::getname()

Definition at line 53 of file SolidObject.cpp.

Here is the caller graph for this function:

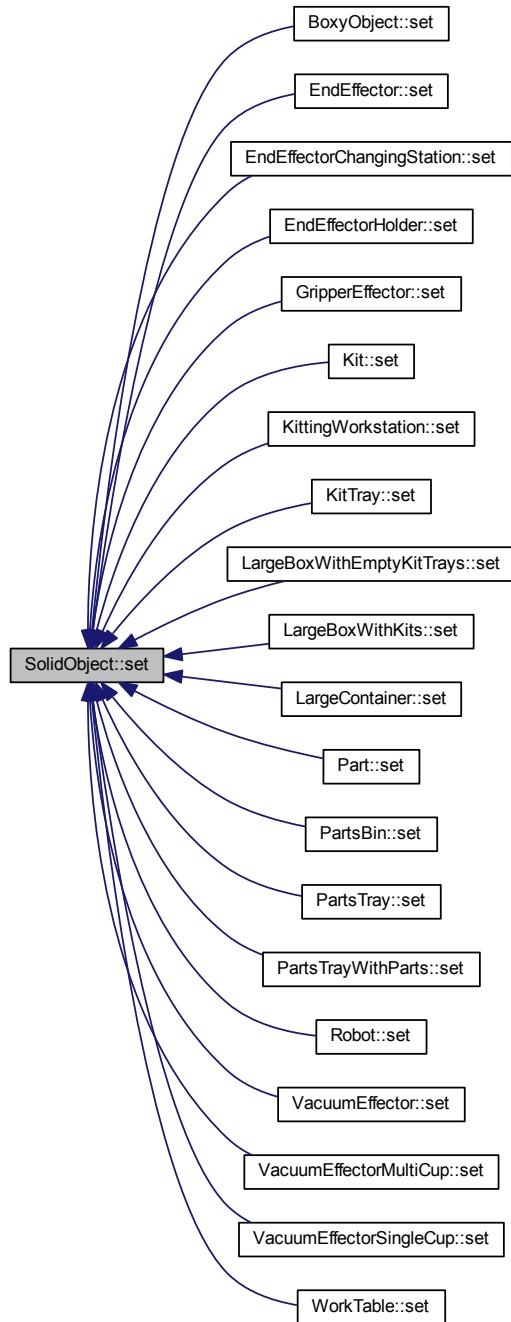


8.63.3.11 int SolidObject::getSolidObjectID()

Definition at line 35 of file SolidObject.cpp.

8.63.3.12 void SolidObject::set (int *id*, SolidObject * *obj*)

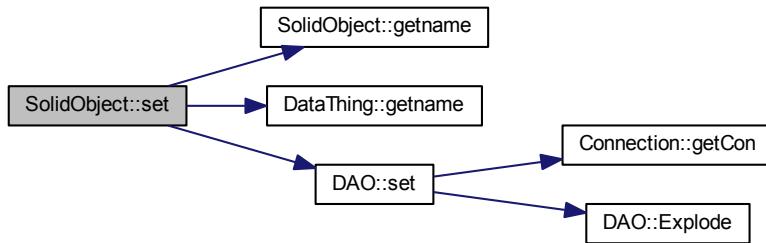
Here is the caller graph for this function:



8.63.3.13 void SolidObject::set (std::string name)

Definition at line 78 of file SolidObject.cpp.

Here is the call graph for this function:



8.63.3.14 void SolidObject::setdao (DAO * _dao)

Definition at line 56 of file SolidObject.cpp.

8.63.3.15 void SolidObject::sethadBySolidObject_WorkTable (WorkTable * _hadBySolidObject_WorkTable)

Definition at line 65 of file SolidObject.cpp.

8.63.3.16 void SolidObject::sethasPhysicalLocation_RefObject (std::vector< PhysicalLocation * > _hasPhysicalLocation_RefObject)

Definition at line 68 of file SolidObject.cpp.

8.63.3.17 void SolidObject::sethasSolidObject_PrimaryLocation (PhysicalLocation * _hasSolidObject_PrimaryLocation)

Definition at line 62 of file SolidObject.cpp.

8.63.3.18 void SolidObject::sethasSolidObject_SecondaryLocation (std::vector< PhysicalLocation * > _hasSolidObject_SecondaryLocation)

Definition at line 59 of file SolidObject.cpp.

8.63.4 Member Data Documentation

8.63.4.1 DAO* SolidObject::dao [private]

Definition at line 30 of file SolidObject.h.

8.63.4.2 `WorkTable* SolidObject::hadBySolidObject_WorkTable [private]`

Definition at line 33 of file SolidObject.h.

8.63.4.3 `std::vector<PhysicalLocation*> SolidObject::hasPhysicalLocation_RefObject [private]`

Definition at line 34 of file SolidObject.h.

8.63.4.4 `PhysicalLocation* SolidObject::hasSolidObject_PrimaryLocation [private]`

Definition at line 32 of file SolidObject.h.

8.63.4.5 `std::vector<PhysicalLocation*> SolidObject::hasSolidObject_SecondaryLocation [private]`

Definition at line 31 of file SolidObject.h.

8.63.4.6 `std::string SolidObject::name [protected]`

Definition at line 36 of file SolidObject.h.

8.63.4.7 `int SolidObject::SolidObjectID [private]`

Definition at line 29 of file SolidObject.h.

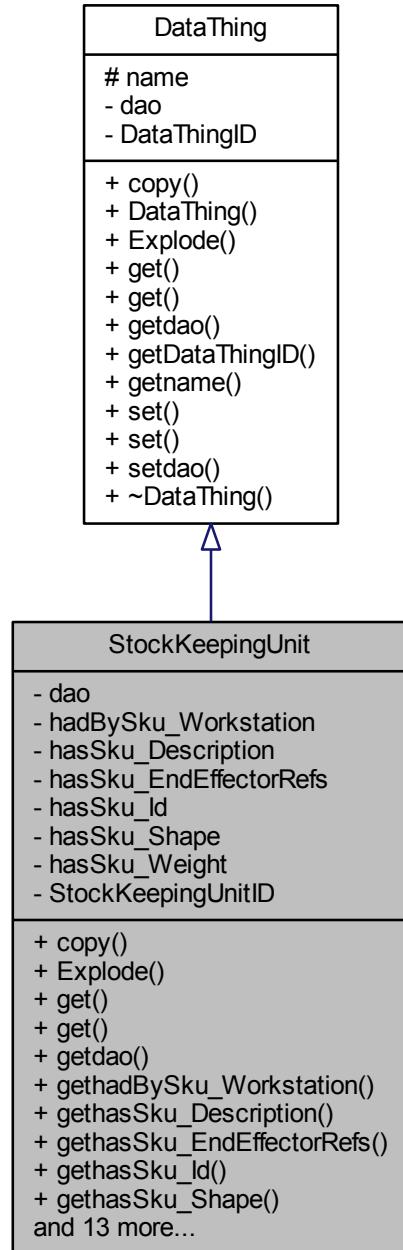
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[SolidObject.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[SolidObject.cpp](#)

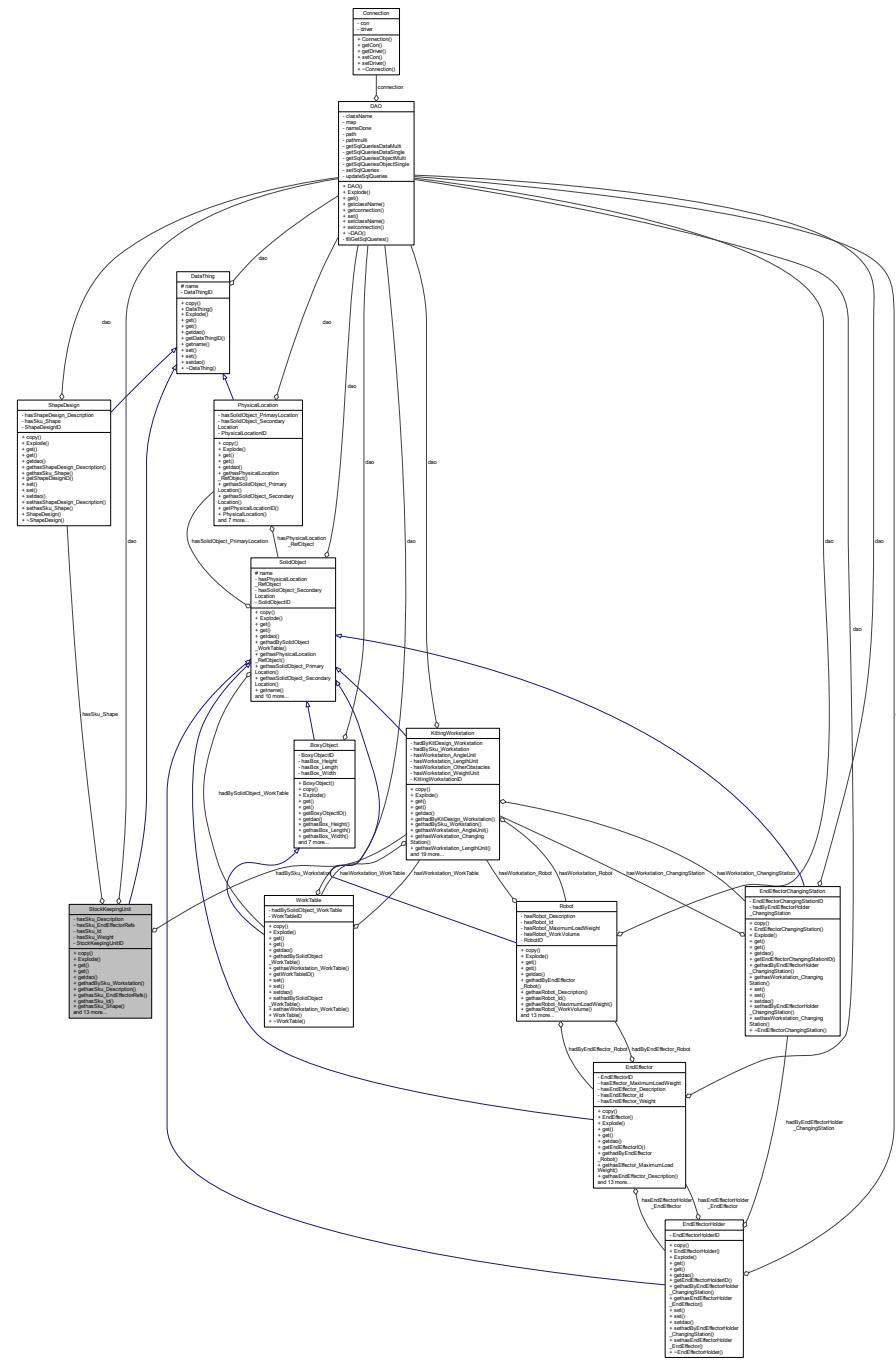
8.64 StockKeepingUnit Class Reference

```
#include <StockKeepingUnit.h>
```

Inheritance diagram for StockKeepingUnit:



Collaboration diagram for StockKeepingUnit:



Public Member Functions

- void `copy` (std::map< std::string, std::string > object)
 - std::vector< std::string > `Explode` (const std::string &str, char separator)
 - void `get` (int id)
 - void `get` (std::string name)

- `DAO * getdao ()`
- `KittingWorkstation * gethadBySku_Workstation ()`
- `std::string gethasSku_Description ()`
- `std::vector< std::string > gethasSku_EndEffectorRefs ()`
- `std::string gethasSku_Id ()`
- `ShapeDesign * gethasSku_Shape ()`
- `double gethasSku_Weight ()`
- `int getStockKeepingUnitID ()`
- `void set (int id, StockKeepingUnit *obj)`
- `void set (std::string name)`
- `void setdao (DAO *_dao)`
- `void sethadBySku_Workstation (KittingWorkstation *_hadBySku_Workstation)`
- `void sethasSku_Description (std::string _hasSku_Description)`
- `void sethasSku_EndEffectorRefs (std::vector< std::string > _hasSku_EndEffectorRefs)`
- `void sethasSku_Id (std::string _hasSku_Id)`
- `void sethasSku_Shape (ShapeDesign *_hasSku_Shape)`
- `void sethasSku_Weight (double _hasSku_Weight)`
- `StockKeepingUnit (std::string name)`
- `~StockKeepingUnit ()`

Private Attributes

- `DAO * dao`
- `KittingWorkstation * hadBySku_Workstation`
- `std::string hasSku_Description`
- `std::vector< std::string > hasSku_EndEffectorRefs`
- `std::string hasSku_Id`
- `ShapeDesign * hasSku_Shape`
- `double hasSku_Weight`
- `int StockKeepingUnitID`

Additional Inherited Members

8.64.1 Detailed Description

Definition at line 29 of file StockKeepingUnit.h.

8.64.2 Constructor & Destructor Documentation

8.64.2.1 StockKeepingUnit::StockKeepingUnit (`std::string name`)

Definition at line 21 of file StockKeepingUnit.cpp.

8.64.2.2 StockKeepingUnit::~StockKeepingUnit ()

Definition at line 26 of file StockKeepingUnit.cpp.

8.64.3 Member Function Documentation

8.64.3.1 void StockKeepingUnit::copy (std::map< std::string, std::string > object)

Definition at line 112 of file StockKeepingUnit.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



8.64.3.2 std::vector< std::string > StockKeepingUnit::Explode (const std::string & str, char separator)

Definition at line 134 of file StockKeepingUnit.cpp.

Here is the caller graph for this function:

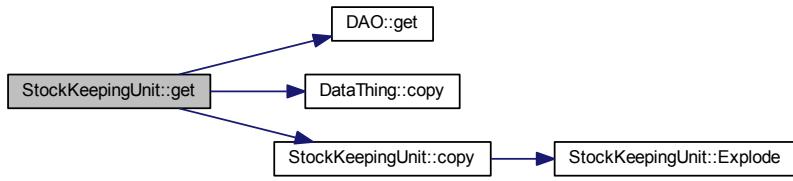


8.64.3.3 void StockKeepingUnit::get (int id)

8.64.3.4 void StockKeepingUnit::get (std::string name)

Definition at line 76 of file StockKeepingUnit.cpp.

Here is the call graph for this function:



8.64.3.5 `DAO * StockKeepingUnit::getdao()`

Definition at line 46 of file StockKeepingUnit.cpp.

8.64.3.6 `KittingWorkstation * StockKeepingUnit::gethadBySku_Workstation()`

Definition at line 49 of file StockKeepingUnit.cpp.

8.64.3.7 `std::string StockKeepingUnit::gethasSku_Description()`

Definition at line 34 of file StockKeepingUnit.cpp.

8.64.3.8 `std::vector< std::string > StockKeepingUnit::gethasSku_EndEffectorRefs()`

Definition at line 31 of file StockKeepingUnit.cpp.

8.64.3.9 `std::string StockKeepingUnit::gethasSku_Id()`

Definition at line 37 of file StockKeepingUnit.cpp.

8.64.3.10 `ShapeDesign * StockKeepingUnit::gethasSku_Shape()`

Definition at line 52 of file StockKeepingUnit.cpp.

8.64.3.11 `double StockKeepingUnit::gethasSku_Weight()`

Definition at line 40 of file StockKeepingUnit.cpp.

8.64.3.12 `int StockKeepingUnit::getStockKeepingUnitID()`

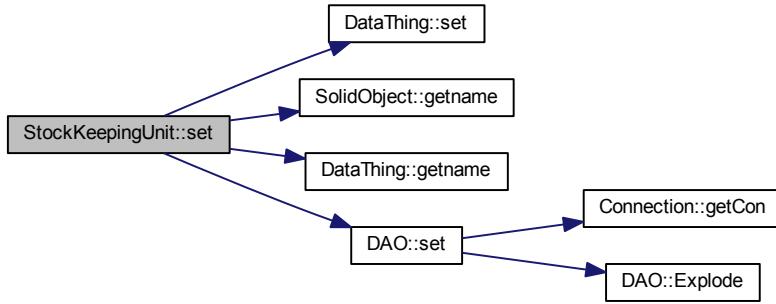
Definition at line 43 of file StockKeepingUnit.cpp.

8.64.3.13 void StockKeepingUnit::set (int *id*, StockKeepingUnit * *obj*)

8.64.3.14 void StockKeepingUnit::set (std::string *name*)

Definition at line 86 of file StockKeepingUnit.cpp.

Here is the call graph for this function:



8.64.3.15 void StockKeepingUnit::setdao (DAO * *_dao*)

Definition at line 67 of file StockKeepingUnit.cpp.

8.64.3.16 void StockKeepingUnit::sethadBySku_Workstation (KittingWorkstation * *_hadBySku_Workstation*)

Definition at line 70 of file StockKeepingUnit.cpp.

8.64.3.17 void StockKeepingUnit::sethasSku_Description (std::string *_hasSku_Description*)

Definition at line 58 of file StockKeepingUnit.cpp.

8.64.3.18 void StockKeepingUnit::sethasSku_EndEffectorRefs (std::vector< std::string > *_hasSku_EndEffectorRefs*)

Definition at line 55 of file StockKeepingUnit.cpp.

8.64.3.19 void StockKeepingUnit::sethasSku_Id (std::string *_hasSku_Id*)

Definition at line 61 of file StockKeepingUnit.cpp.

8.64.3.20 void StockKeepingUnit::sethasSku_Shape (ShapeDesign * *_hasSku_Shape*)

Definition at line 73 of file StockKeepingUnit.cpp.

8.64.3.21 void StockKeepingUnit::sethasSku_Weight (double _hasSku_Weight)

Definition at line 64 of file StockKeepingUnit.cpp.

8.64.4 Member Data Documentation

8.64.4.1 DAO* StockKeepingUnit::dao [private]

Definition at line 35 of file StockKeepingUnit.h.

8.64.4.2 KittingWorkstation* StockKeepingUnit::hadBySku_Workstation [private]

Definition at line 36 of file StockKeepingUnit.h.

8.64.4.3 std::string StockKeepingUnit::hasSku_Description [private]

Definition at line 31 of file StockKeepingUnit.h.

8.64.4.4 std::vector<std::string> StockKeepingUnit::hasSku_EndEffectorRefs [private]

Definition at line 30 of file StockKeepingUnit.h.

8.64.4.5 std::string StockKeepingUnit::hasSku_Id [private]

Definition at line 32 of file StockKeepingUnit.h.

8.64.4.6 ShapeDesign* StockKeepingUnit::hasSku_Shape [private]

Definition at line 37 of file StockKeepingUnit.h.

8.64.4.7 double StockKeepingUnit::hasSku_Weight [private]

Definition at line 33 of file StockKeepingUnit.h.

8.64.4.8 int StockKeepingUnit::StockKeepingUnitID [private]

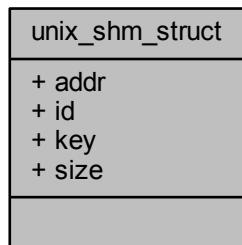
Definition at line 34 of file StockKeepingUnit.h.

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

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[StockKeepingUnit.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[StockKeepingUnit.cpp](#)

8.65 unix_shm_struct Struct Reference

Collaboration diagram for unix_shm_struct:



Public Attributes

- void * `addr`
- `ulapi_id` `id`
- `ulapi_id` `key`
- `ulapi_integer` `size`

8.65.1 Detailed Description

Definition at line 58 of file ulapi.cpp.

8.65.2 Member Data Documentation

8.65.2.1 `void* unix_shm_struct::addr`

Definition at line 63 of file ulapi.cpp.

8.65.2.2 `ulapi_id unix_shm_struct::id`

Definition at line 62 of file ulapi.cpp.

8.65.2.3 `ulapi_id unix_shm_struct::key`

Definition at line 60 of file ulapi.cpp.

8.65.2.4 `ulapi_integer unix_shm_struct::size`

Definition at line 61 of file ulapi.cpp.

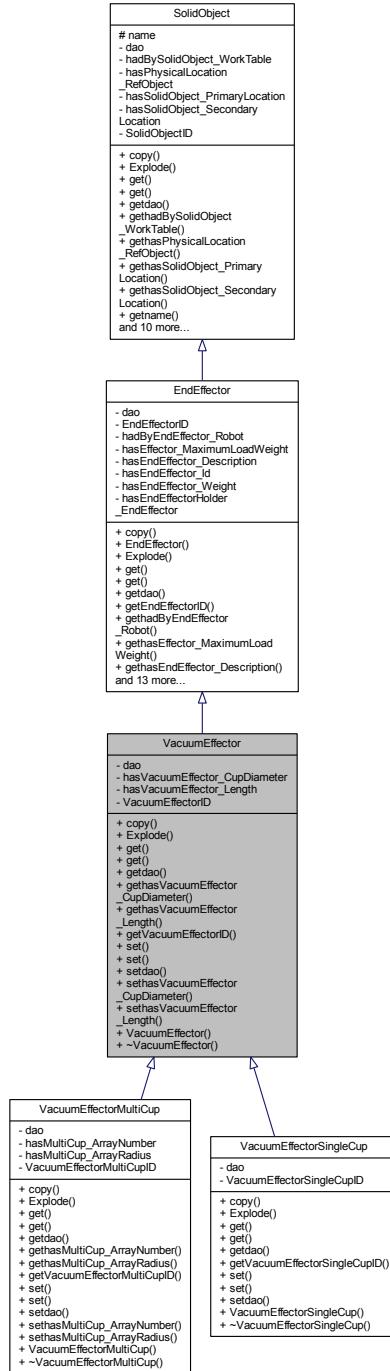
The documentation for this struct was generated from the following file:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/[ulapi.cpp](#)

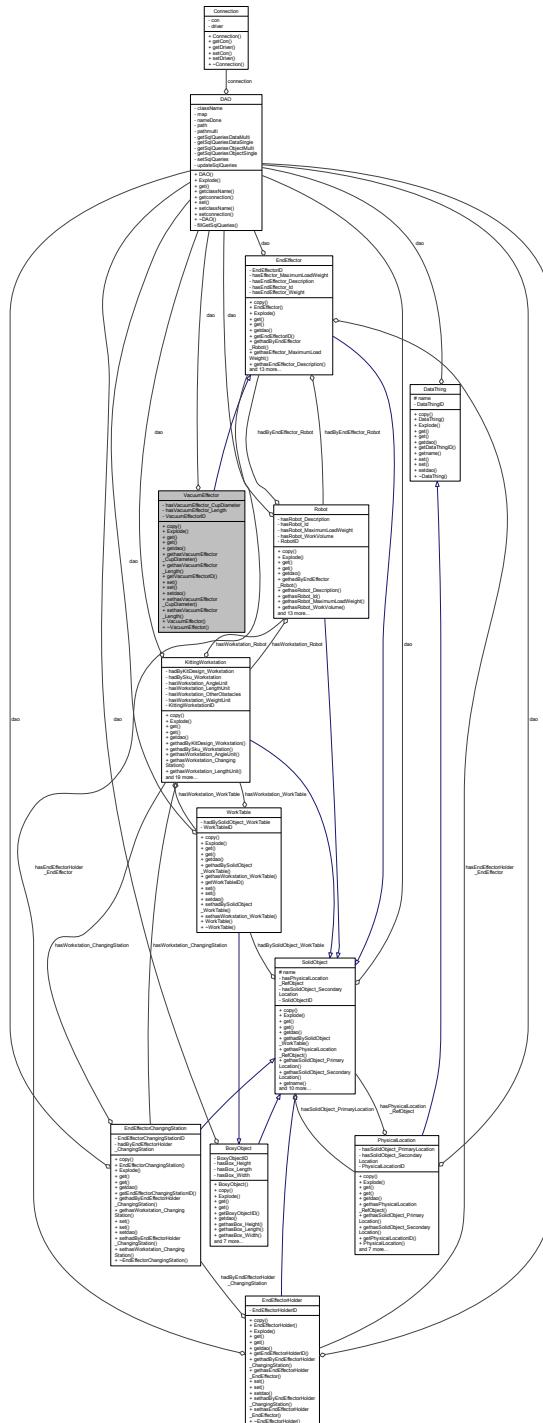
8.66 VacuumEffector Class Reference

```
#include <VacuumEffector.h>
```

Inheritance diagram for VacuumEffector:



Collaboration diagram for VacuumEffector:



Public Member Functions

- void **copy** (std::map< std::string, std::string > object)
 - std::vector< std::string > **Explode** (const std::string &str, char separator)

- void `get` (int id)
- void `get` (std::string name)
- DAO * `getdao` ()
- double `gethasVacuumEffector_CupDiameter` ()
- double `gethasVacuumEffector_Length` ()
- int `getVacuumEffectorID` ()
- void `set` (int id, VacuumEffector *obj)
- void `set` (std::string name)
- void `setdao` (DAO *_dao)
- void `sethasVacuumEffector_CupDiameter` (double _hasVacuumEffector_CupDiameter)
- void `sethasVacuumEffector_Length` (double _hasVacuumEffector_Length)
- VacuumEffector (std::string name)
- ~VacuumEffector ()

Private Attributes

- DAO * dao
- double hasVacuumEffector_CupDiameter
- double hasVacuumEffector_Length
- int VacuumEffectorID

Additional Inherited Members

8.66.1 Detailed Description

Definition at line 27 of file VacuumEffector.h.

8.66.2 Constructor & Destructor Documentation

8.66.2.1 VacuumEffector::VacuumEffector (std::string name)

Definition at line 19 of file VacuumEffector.cpp.

8.66.2.2 VacuumEffector::~VacuumEffector ()

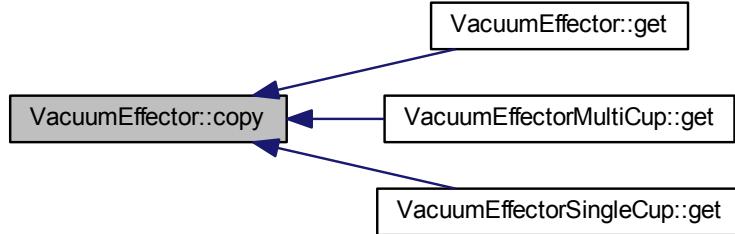
Definition at line 22 of file VacuumEffector.cpp.

8.66.3 Member Function Documentation

8.66.3.1 void VacuumEffector::copy (std::map< std::string, std::string > object)

Definition at line 81 of file VacuumEffector.cpp.

Here is the caller graph for this function:



8.66.3.2 std::vector< std::string > VacuumEffector::Explode (const std::string & str, char separator)

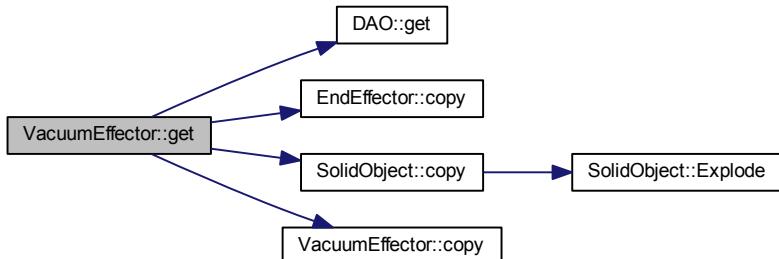
Definition at line 92 of file VacuumEffector.cpp.

8.66.3.3 void VacuumEffector::get (int id)

8.66.3.4 void VacuumEffector::get (std::string name)

Definition at line 46 of file VacuumEffector.cpp.

Here is the call graph for this function:



8.66.3.5 DAO * VacuumEffector::getdao ()

Definition at line 34 of file VacuumEffector.cpp.

8.66.3.6 double VacuumEffector::getHasVacuumEffector_CupDiameter()

Definition at line 25 of file VacuumEffector.cpp.

8.66.3.7 double VacuumEffector::getHasVacuumEffector_Length()

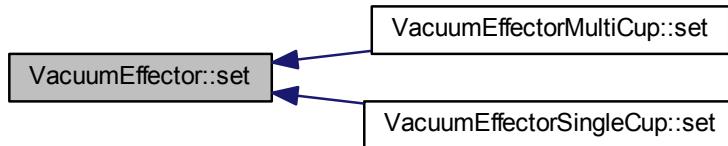
Definition at line 28 of file VacuumEffector.cpp.

8.66.3.8 int VacuumEffector::getVacuumEffectorID()

Definition at line 31 of file VacuumEffector.cpp.

8.66.3.9 void VacuumEffector::set(int *id*, VacuumEffector * *obj*)

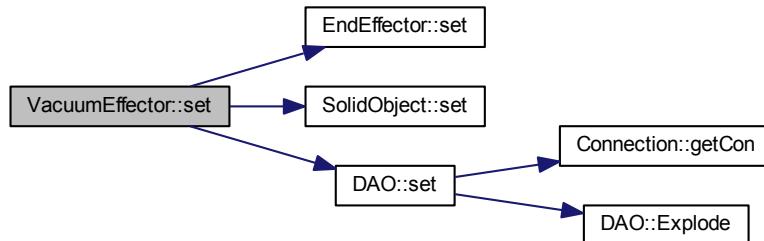
Here is the caller graph for this function:



8.66.3.10 void VacuumEffector::set(std::string *name*)

Definition at line 59 of file VacuumEffector.cpp.

Here is the call graph for this function:



8.66.3.11 void VacuumEffector::setdao (DAO * _dao)

Definition at line 43 of file VacuumEffector.cpp.

8.66.3.12 void VacuumEffector::sethasVacuumEffector_CupDiameter (double _hasVacuumEffector_CupDiameter)

Definition at line 37 of file VacuumEffector.cpp.

8.66.3.13 void VacuumEffector::sethasVacuumEffector_Length (double _hasVacuumEffector_Length)

Definition at line 40 of file VacuumEffector.cpp.

8.66.4 Member Data Documentation

8.66.4.1 DAO* VacuumEffector::dao [private]

Definition at line 31 of file VacuumEffector.h.

8.66.4.2 double VacuumEffector::hasVacuumEffector_CupDiameter [private]

Definition at line 28 of file VacuumEffector.h.

8.66.4.3 double VacuumEffector::hasVacuumEffector_Length [private]

Definition at line 29 of file VacuumEffector.h.

8.66.4.4 int VacuumEffector::VacuumEffectorID [private]

Definition at line 30 of file VacuumEffector.h.

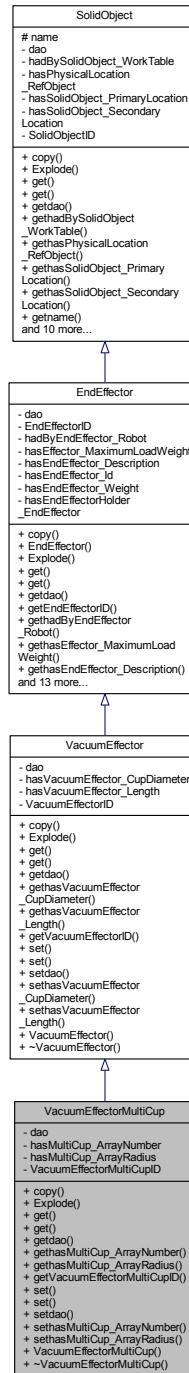
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[VacuumEffector.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[VacuumEffector.cpp](#)

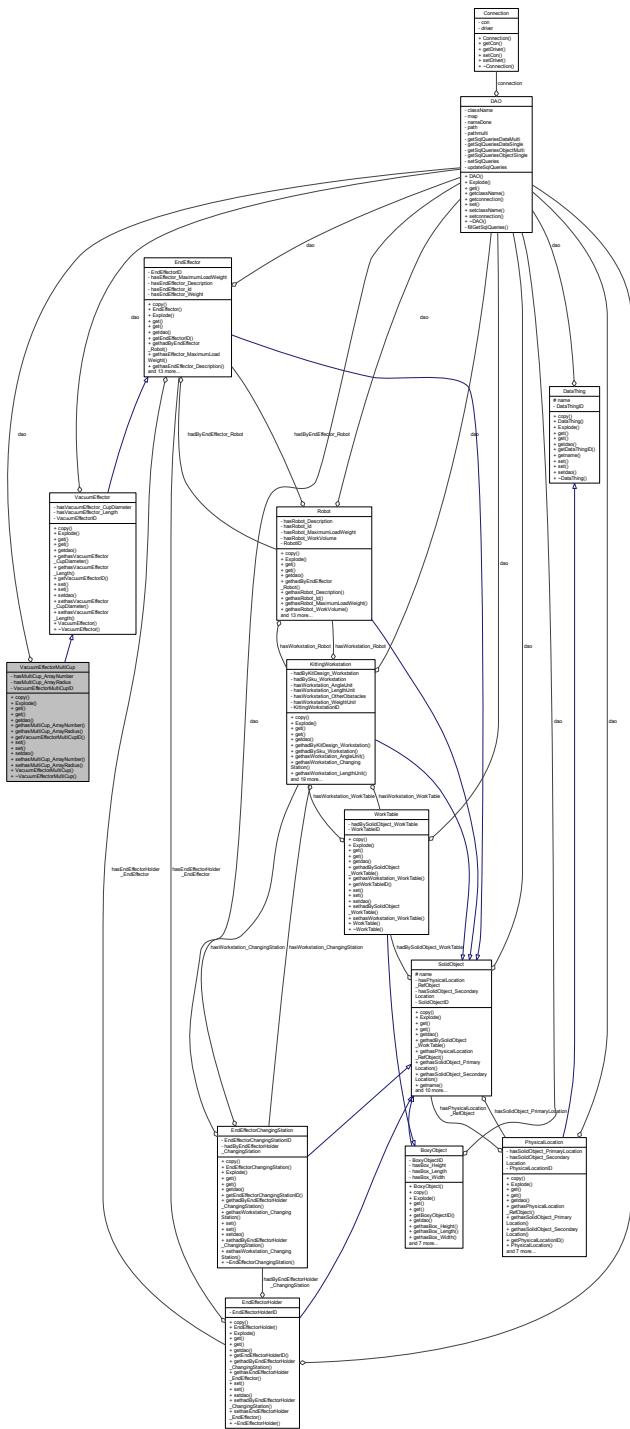
8.67 VacuumEffectorMultiCup Class Reference

```
#include <VacuumEffectorMultiCup.h>
```

Inheritance diagram for VacuumEffectorMultiCup:



Collaboration diagram for VacuumEffectorMultiCup:



Public Member Functions

- void `copy` (std::map< std::string, std::string > object)
- std::vector< std::string > `Explode` (const std::string &str, char separator)

- void [get](#) (int id)
- void [get](#) (std::string name)
- DAO * [getdao](#) ()
- std::string [gethasMultiCup_ArrayNumber](#) ()
- double [gethasMultiCup_ArrayRadius](#) ()
- int [getVacuumEffectorMultiCupID](#) ()
- void [set](#) (int id, VacuumEffectorMultiCup *obj)
- void [set](#) (std::string name)
- void [setdao](#) (DAO *_dao)
- void [sethasMultiCup_ArrayNumber](#) (std::string _hasMultiCup_ArrayNumber)
- void [sethasMultiCup_ArrayRadius](#) (double _hasMultiCup_ArrayRadius)
- VacuumEffectorMultiCup (std::string name)
- ~VacuumEffectorMultiCup ()

Private Attributes

- DAO * dao
- std::string hasMultiCup_ArrayNumber
- double hasMultiCup_ArrayRadius
- int VacuumEffectorMultiCupID

Additional Inherited Members

8.67.1 Detailed Description

Definition at line 27 of file VacuumEffectorMultiCup.h.

8.67.2 Constructor & Destructor Documentation

8.67.2.1 VacuumEffectorMultiCup::VacuumEffectorMultiCup (std::string name)

Definition at line 19 of file VacuumEffectorMultiCup.cpp.

8.67.2.2 VacuumEffectorMultiCup::~VacuumEffectorMultiCup ()

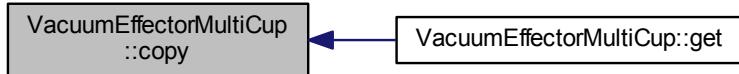
Definition at line 22 of file VacuumEffectorMultiCup.cpp.

8.67.3 Member Function Documentation

8.67.3.1 void VacuumEffectorMultiCup::copy (std::map< std::string, std::string > object)

Definition at line 84 of file VacuumEffectorMultiCup.cpp.

Here is the caller graph for this function:



8.67.3.2 `std::vector< std::string > VacuumEffectorMultiCup::Explode (const std::string & str, char separator)`

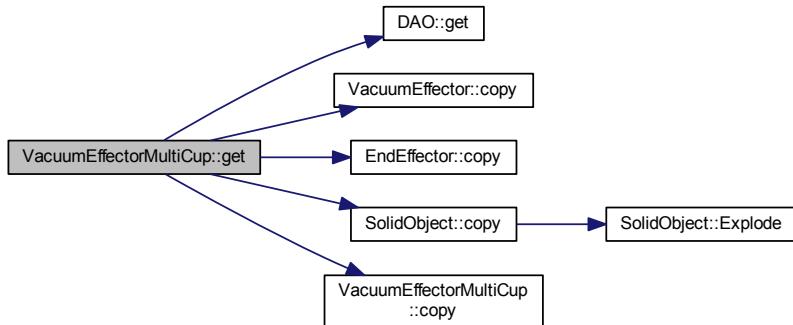
Definition at line 95 of file VacuumEffectorMultiCup.cpp.

8.67.3.3 `void VacuumEffectorMultiCup::get (int id)`

8.67.3.4 `void VacuumEffectorMultiCup::get (std::string name)`

Definition at line 46 of file VacuumEffectorMultiCup.cpp.

Here is the call graph for this function:



8.67.3.5 `DAO * VacuumEffectorMultiCup::getdao ()`

Definition at line 34 of file VacuumEffectorMultiCup.cpp.

8.67.3.6 `std::string VacuumEffectorMultiCup::gethasMultiCup_ArrayNumber ()`

Definition at line 25 of file VacuumEffectorMultiCup.cpp.

8.67.3.7 double VacuumEffectorMultiCup::getHasMultiCup_ArrayRadius()

Definition at line 28 of file VacuumEffectorMultiCup.cpp.

8.67.3.8 int VacuumEffectorMultiCup::getVacuumEffectorMultiCupID()

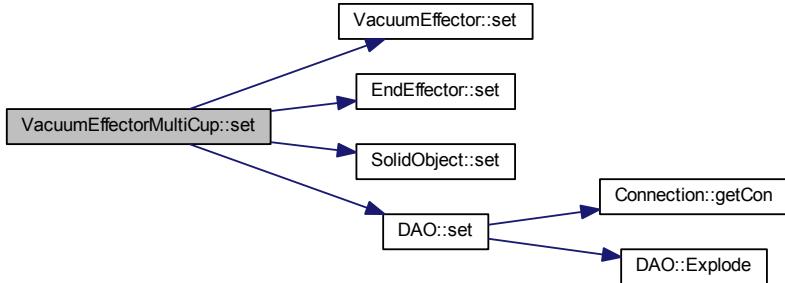
Definition at line 31 of file VacuumEffectorMultiCup.cpp.

8.67.3.9 void VacuumEffectorMultiCup::set(int *id*, VacuumEffectorMultiCup * *obj*)

8.67.3.10 void VacuumEffectorMultiCup::set(std::string *name*)

Definition at line 62 of file VacuumEffectorMultiCup.cpp.

Here is the call graph for this function:



8.67.3.11 void VacuumEffectorMultiCup::setdao(DAO * *dao*)

Definition at line 43 of file VacuumEffectorMultiCup.cpp.

8.67.3.12 void VacuumEffectorMultiCup::setHasMultiCup_ArrayNumber(std::string *_hasMultiCup_ArrayNumber*)

Definition at line 37 of file VacuumEffectorMultiCup.cpp.

8.67.3.13 void VacuumEffectorMultiCup::setHasMultiCup_ArrayRadius(double *_hasMultiCup_ArrayRadius*)

Definition at line 40 of file VacuumEffectorMultiCup.cpp.

8.67.4 Member Data Documentation

8.67.4.1 DAO* VacuumEffectorMultiCup::*dao* [private]

Definition at line 31 of file VacuumEffectorMultiCup.h.

8.67.4.2 `std::string VacuumEffectorMultiCup::hasMultiCup_ArrayNumber [private]`

Definition at line 28 of file VacuumEffectorMultiCup.h.

8.67.4.3 `double VacuumEffectorMultiCup::hasMultiCup_ArrayRadius [private]`

Definition at line 29 of file VacuumEffectorMultiCup.h.

8.67.4.4 `int VacuumEffectorMultiCup::VacuumEffectorMultiCupID [private]`

Definition at line 30 of file VacuumEffectorMultiCup.h.

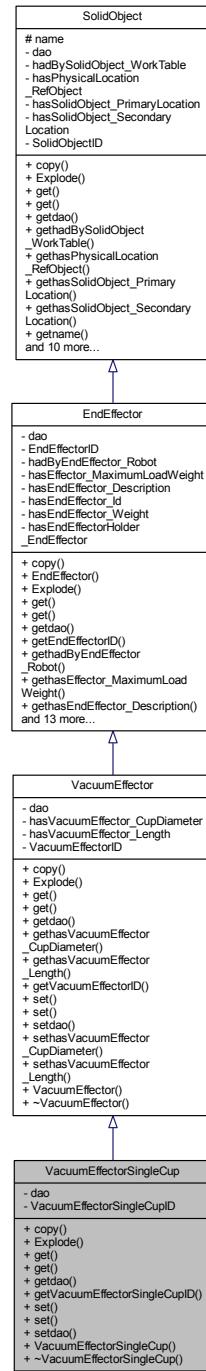
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[VacuumEffectorMultiCup.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[VacuumEffectorMultiCup.cpp](#)

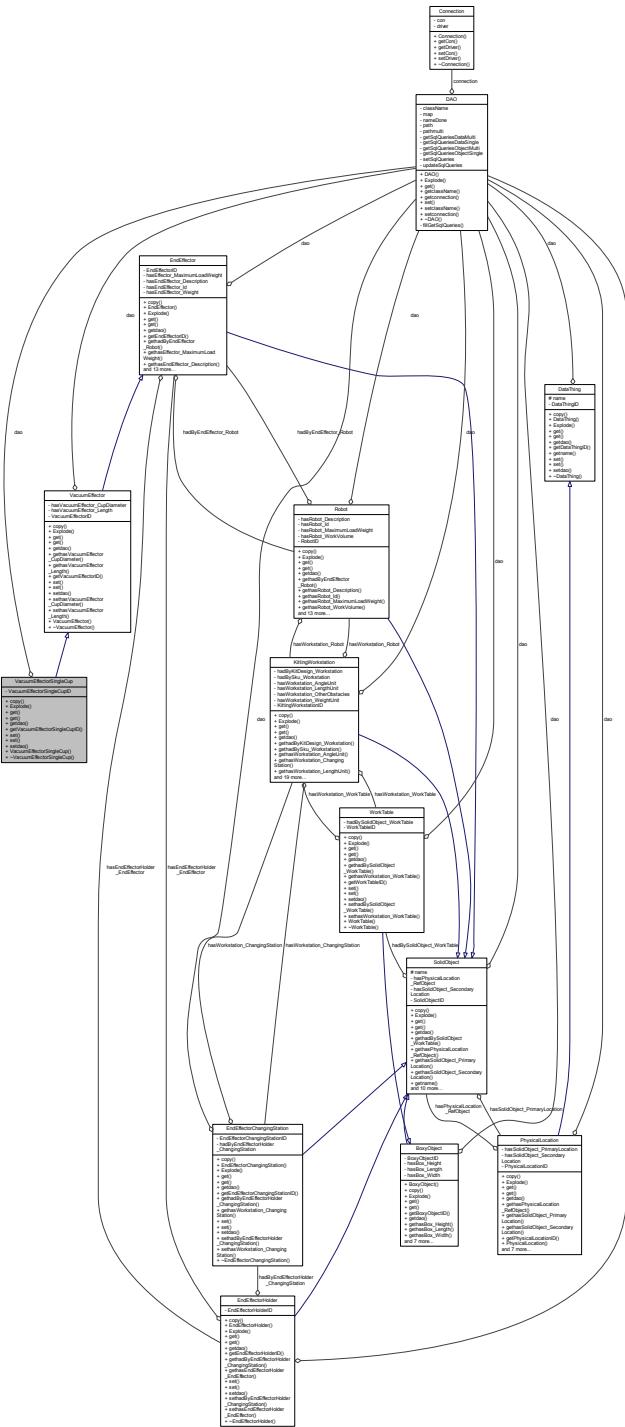
8.68 VacuumEffectorSingleCup Class Reference

```
#include <VacuumEffectorSingleCup.h>
```

Inheritance diagram for VacuumEffectorSingleCup:



Collaboration diagram for VacuumEffectorSingleCup:



Public Member Functions

- void `copy` (std::map< std::string, std::string > object)
 - std::vector< std::string > `Explode` (const std::string &str, char separator)

- void [get](#) (int id)
- void [get](#) (std::string name)
- DAO * [getdao](#) ()
- int [getVacuumEffectorSingleCupID](#) ()
- void [set](#) (int id, VacuumEffectorSingleCup *obj)
- void [set](#) (std::string name)
- void [setdao](#) (DAO *_dao)
- VacuumEffectorSingleCup (std::string name)
- ~VacuumEffectorSingleCup ()

Private Attributes

- DAO * dao
- int VacuumEffectorSingleCupID

Additional Inherited Members

8.68.1 Detailed Description

Definition at line 27 of file VacuumEffectorSingleCup.h.

8.68.2 Constructor & Destructor Documentation

8.68.2.1 VacuumEffectorSingleCup::VacuumEffectorSingleCup (std::string name)

Definition at line 19 of file VacuumEffectorSingleCup.cpp.

8.68.2.2 VacuumEffectorSingleCup::~VacuumEffectorSingleCup ()

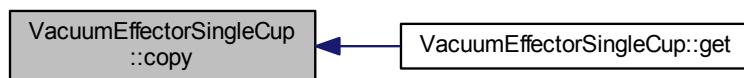
Definition at line 22 of file VacuumEffectorSingleCup.cpp.

8.68.3 Member Function Documentation

8.68.3.1 void VacuumEffectorSingleCup::copy (std::map< std::string, std::string > object)

Definition at line 68 of file VacuumEffectorSingleCup.cpp.

Here is the caller graph for this function:



8.68.3.2 `std::vector< std::string > VacuumEffectorSingleCup::Explode (const std::string & str, char separator)`

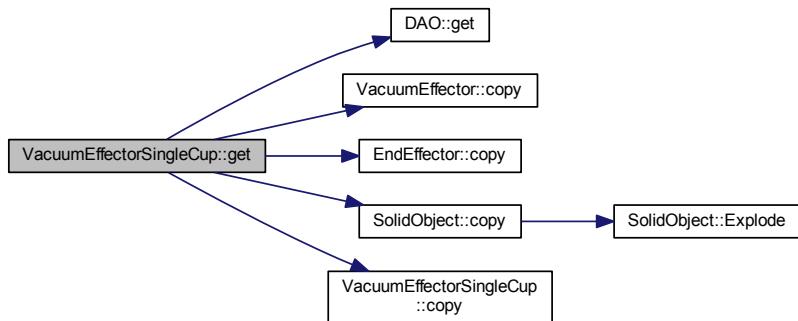
Definition at line 77 of file VacuumEffectorSingleCup.cpp.

8.68.3.3 `void VacuumEffectorSingleCup::get (int id)`

8.68.3.4 `void VacuumEffectorSingleCup::get (std::string name)`

Definition at line 34 of file VacuumEffectorSingleCup.cpp.

Here is the call graph for this function:



8.68.3.5 `DAO * VacuumEffectorSingleCup::getdao ()`

Definition at line 28 of file VacuumEffectorSingleCup.cpp.

8.68.3.6 `int VacuumEffectorSingleCup::getVacuumEffectorSingleCupID ()`

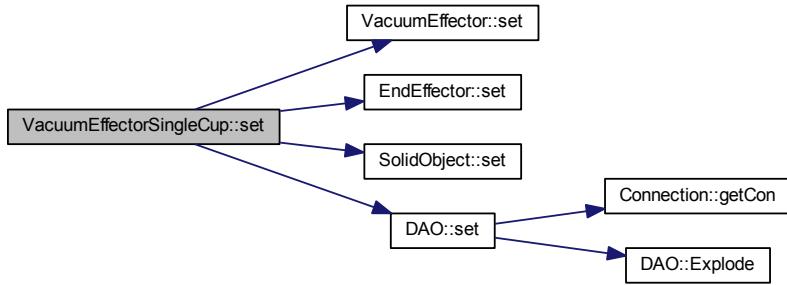
Definition at line 25 of file VacuumEffectorSingleCup.cpp.

8.68.3.7 `void VacuumEffectorSingleCup::set (int id, VacuumEffectorSingleCup * obj)`

8.68.3.8 `void VacuumEffectorSingleCup::set (std::string name)`

Definition at line 50 of file VacuumEffectorSingleCup.cpp.

Here is the call graph for this function:



8.68.3.9 void VacuumEffectorSingleCup::setdao (DAO * _dao)

Definition at line 31 of file VacuumEffectorSingleCup.cpp.

8.68.4 Member Data Documentation

8.68.4.1 DAO* VacuumEffectorSingleCup::dao [private]

Definition at line 29 of file VacuumEffectorSingleCup.h.

8.68.4.2 int VacuumEffectorSingleCup::VacuumEffectorSingleCupID [private]

Definition at line 28 of file VacuumEffectorSingleCup.h.

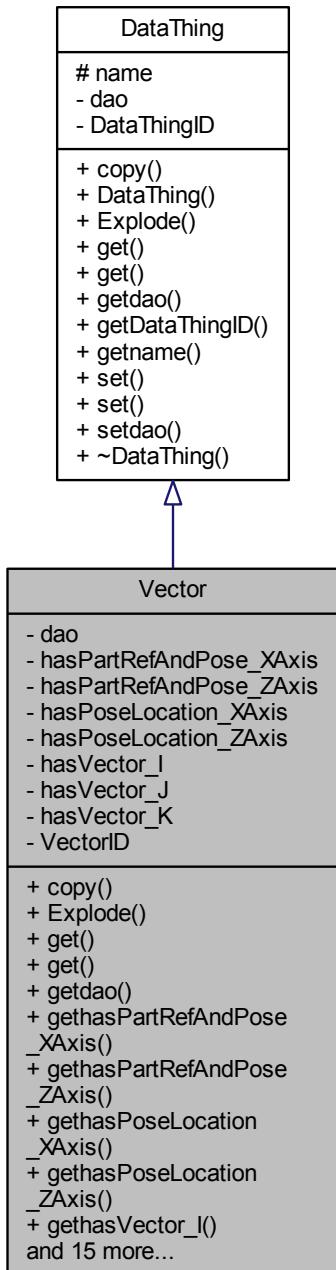
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[VacuumEffectorSingleCup.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[VacuumEffectorSingleCup.cpp](#)

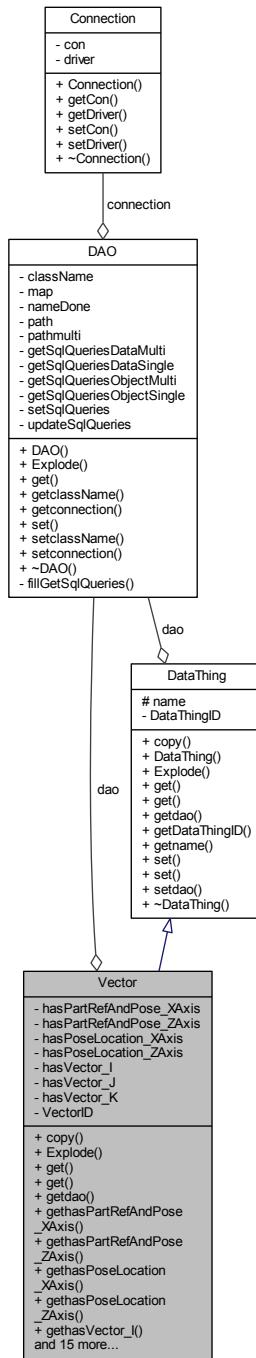
8.69 Vector Class Reference

```
#include <Vector.h>
```

Inheritance diagram for Vector:



Collaboration diagram for Vector:



Public Member Functions

- void `copy` (std::map< std::string, std::string > object)
 - std::vector< std::string > `Explode` (const std::string &str, char separator)

- void `get` (int id)
- void `get` (std::string name)
- DAO * `getdao` ()
- std::vector< PartRefAndPose * > `gethasPartRefAndPose_XAxis` ()
- std::vector< PartRefAndPose * > `gethasPartRefAndPose_ZAxis` ()
- std::vector< PoseLocation * > `gethasPoseLocation_XAxis` ()
- std::vector< PoseLocation * > `gethasPoseLocation_ZAxis` ()
- double `gethasVector_I` ()
- double `gethasVector_J` ()
- double `gethasVector_K` ()
- int `getVectorID` ()
- void `set` (int id, Vector *obj)
- void `set` (std::string name)
- void `setdao` (DAO *_dao)
- void `sethasPartRefAndPose_XAxis` (std::vector< PartRefAndPose * > _hasPartRefAndPose_XAxis)
- void `sethasPartRefAndPose_ZAxis` (std::vector< PartRefAndPose * > _hasPartRefAndPose_ZAxis)
- void `sethasPoseLocation_XAxis` (std::vector< PoseLocation * > _hasPoseLocation_XAxis)
- void `sethasPoseLocation_ZAxis` (std::vector< PoseLocation * > _hasPoseLocation_ZAxis)
- void `sethasVector_I` (double _hasVector_I)
- void `sethasVector_J` (double _hasVector_J)
- void `sethasVector_K` (double _hasVector_K)
- Vector (std::string name)
- ~Vector ()

Private Attributes

- DAO * dao
- std::vector< PartRefAndPose * > hasPartRefAndPose_XAxis
- std::vector< PartRefAndPose * > hasPartRefAndPose_ZAxis
- std::vector< PoseLocation * > hasPoseLocation_XAxis
- std::vector< PoseLocation * > hasPoseLocation_ZAxis
- double hasVector_I
- double hasVector_J
- double hasVector_K
- int VectorID

Additional Inherited Members

8.69.1 Detailed Description

Definition at line 29 of file Vector.h.

8.69.2 Constructor & Destructor Documentation

8.69.2.1 Vector::Vector (std::string name)

Definition at line 21 of file Vector.cpp.

8.69.2.2 Vector::~Vector()

Definition at line 24 of file Vector.cpp.

8.69.3 Member Function Documentation

8.69.3.1 void Vector::copy(std::map< std::string, std::string > object)

Definition at line 143 of file Vector.cpp.

Here is the call graph for this function:



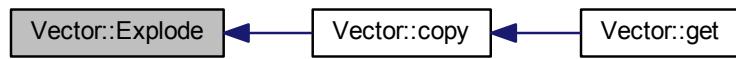
Here is the caller graph for this function:



8.69.3.2 std::vector< std::string > Vector::Explode(const std::string & str, char separator)

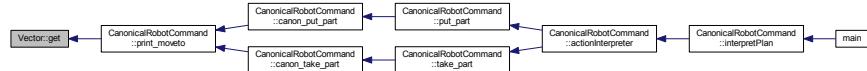
Definition at line 179 of file Vector.cpp.

Here is the caller graph for this function:



8.69.3.3 void Vector::get(int id)

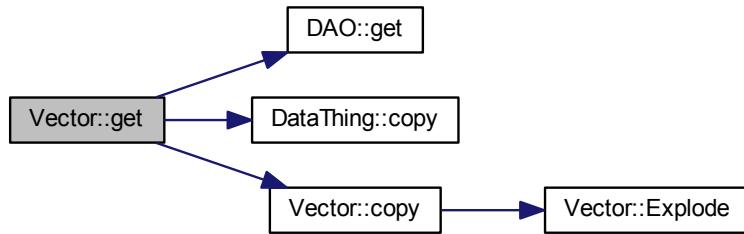
Here is the caller graph for this function:



8.69.3.4 void Vector::get(std::string name)

Definition at line 86 of file Vector.cpp.

Here is the call graph for this function:



8.69.3.5 DAO * Vector::getdao()

Definition at line 47 of file Vector.cpp.

8.69.3.6 std::vector< PartRefAndPose * > Vector::gethasPartRefAndPose_XAxis()

Definition at line 53 of file Vector.cpp.

8.69.3.7 std::vector< PartRefAndPose * > Vector::gethasPartRefAndPose_ZAxis()

Definition at line 50 of file Vector.cpp.

8.69.3.8 std::vector< PoseLocation * > Vector::gethasPoseLocation_XAxis()

Definition at line 59 of file Vector.cpp.

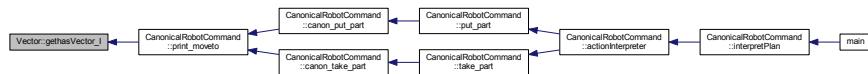
8.69.3.9 `std::vector< PoseLocation * > Vector::getHasPoseLocation_ZAxis()`

Definition at line 56 of file Vector.cpp.

8.69.3.10 `double Vector::getHasVector_I()`

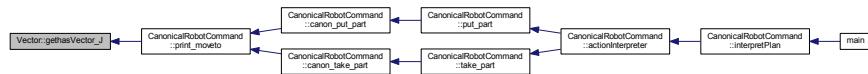
Definition at line 41 of file Vector.cpp.

Here is the caller graph for this function:

8.69.3.11 `double Vector::getHasVector_J()`

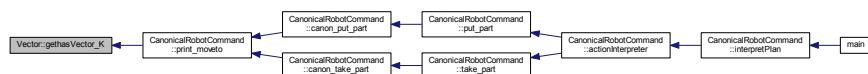
Definition at line 38 of file Vector.cpp.

Here is the caller graph for this function:

8.69.3.12 `double Vector::getHasVector_K()`

Definition at line 35 of file Vector.cpp.

Here is the caller graph for this function:

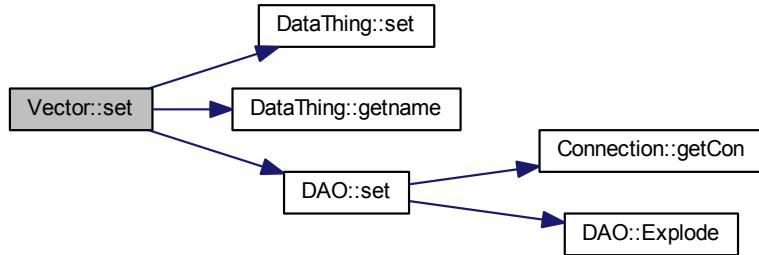
8.69.3.13 `int Vector::getVectorID()`

Definition at line 44 of file Vector.cpp.

8.69.3.14 `void Vector::set(int id, Vector * obj)`8.69.3.15 `void Vector::set(std::string name)`

Definition at line 96 of file Vector.cpp.

Here is the call graph for this function:



8.69.3.16 void Vector::setdao (DAO * _dao)

Definition at line 71 of file Vector.cpp.

8.69.3.17 void Vector::sethasPartRefAndPose_XAxis (std::vector< PartRefAndPose * > _hasPartRefAndPose_XAxis)

Definition at line 77 of file Vector.cpp.

8.69.3.18 void Vector::sethasPartRefAndPose_ZAxis (std::vector< PartRefAndPose * > _hasPartRefAndPose_ZAxis)

Definition at line 74 of file Vector.cpp.

8.69.3.19 void Vector::sethasPoseLocation_XAxis (std::vector< PoseLocation * > _hasPoseLocation_XAxis)

Definition at line 83 of file Vector.cpp.

8.69.3.20 void Vector::sethasPoseLocation_ZAxis (std::vector< PoseLocation * > _hasPoseLocation_ZAxis)

Definition at line 80 of file Vector.cpp.

8.69.3.21 void Vector::sethasVector_I (double _hasVector_I)

Definition at line 68 of file Vector.cpp.

8.69.3.22 void Vector::sethasVector_J (double _hasVector_J)

Definition at line 65 of file Vector.cpp.

8.69.3.23 void Vector::sethasVector_K (double _hasVector_K)

Definition at line 62 of file Vector.cpp.

8.69.4 Member Data Documentation

8.69.4.1 DAO* Vector::dao [private]

Definition at line 34 of file Vector.h.

8.69.4.2 std::vector<PartRefAndPose*> Vector::hasPartRefAndPose_XAxis [private]

Definition at line 36 of file Vector.h.

8.69.4.3 std::vector<PartRefAndPose*> Vector::hasPartRefAndPose_ZAxis [private]

Definition at line 35 of file Vector.h.

8.69.4.4 std::vector<PoseLocation*> Vector::hasPoseLocation_XAxis [private]

Definition at line 38 of file Vector.h.

8.69.4.5 std::vector<PoseLocation*> Vector::hasPoseLocation_ZAxis [private]

Definition at line 37 of file Vector.h.

8.69.4.6 double Vector::hasVector_I [private]

Definition at line 32 of file Vector.h.

8.69.4.7 double Vector::hasVector_J [private]

Definition at line 31 of file Vector.h.

8.69.4.8 double Vector::hasVector_K [private]

Definition at line 30 of file Vector.h.

8.69.4.9 int Vector::VectorID [private]

Definition at line 33 of file Vector.h.

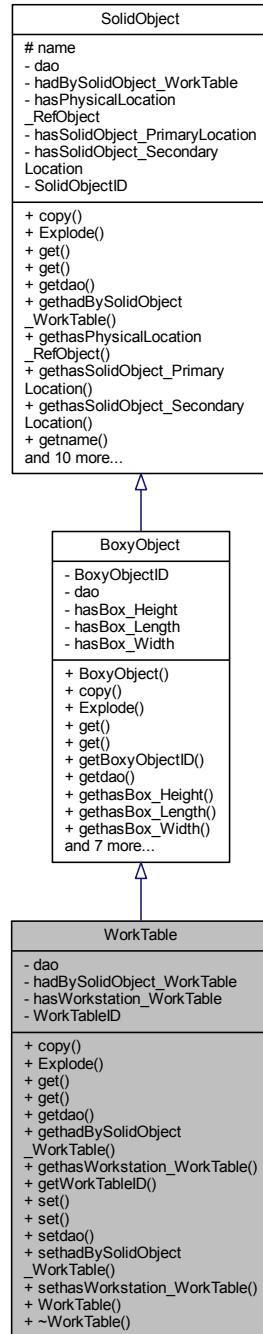
The documentation for this class was generated from the following files:

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[Vector.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[Vector.cpp](#)

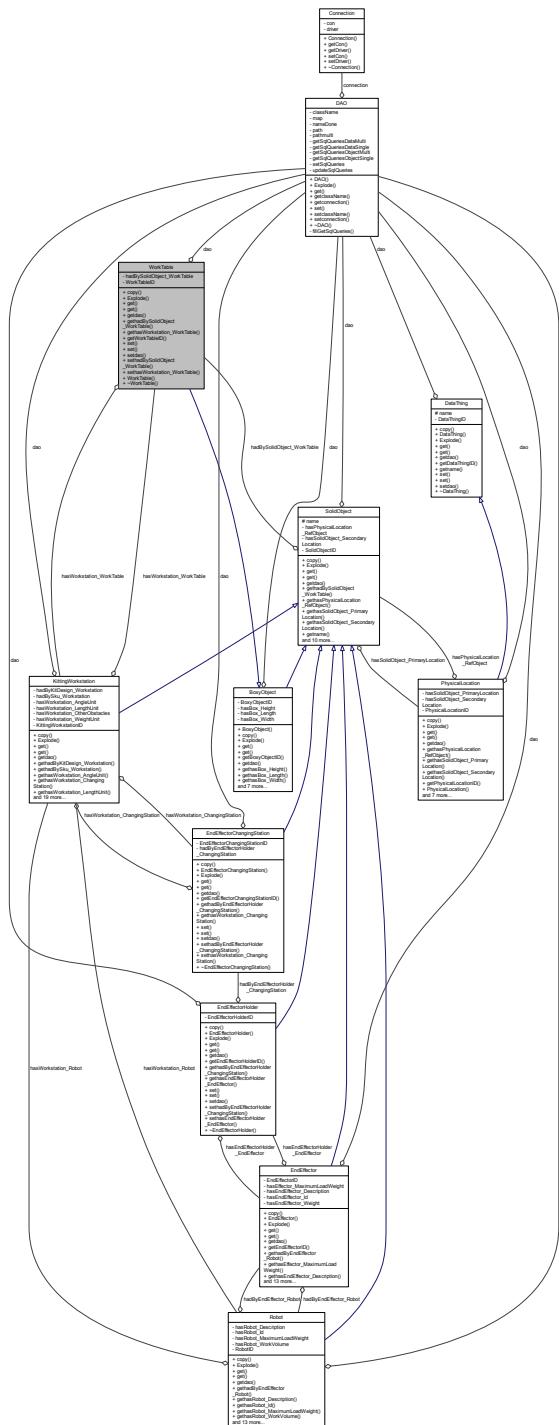
8.70 WorkTable Class Reference

```
#include <WorkTable.h>
```

Inheritance diagram for WorkTable:



Collaboration diagram for WorkTable:



Public Member Functions

- void `copy` (std::map< std::string, std::string > object)
 - std::vector< std::string > `Explode` (const std::string &str, char separator)

- void [get](#) (int id)
- void [get](#) (std::string name)
- DAO * [getdao](#) ()
- std::vector< [SolidObject](#) * > [gethadBySolidObject_WorkTable](#) ()
- KittingWorkstation * [gethasWorkstation_WorkTable](#) ()
- int [getWorkTableID](#) ()
- void [set](#) (int id, [WorkTable](#) *obj)
- void [set](#) (std::string name)
- void [setdao](#) (DAO *_dao)
- void [sethadBySolidObject_WorkTable](#) (std::vector< [SolidObject](#) * > _hadBySolidObject_WorkTable)
- void [sethasWorkstation_WorkTable](#) (KittingWorkstation *_hasWorkstation_WorkTable)
- [WorkTable](#) (std::string name)
- [~WorkTable](#) ()

Private Attributes

- DAO * dao
- std::vector< [SolidObject](#) * > hadBySolidObject_WorkTable
- KittingWorkstation * hasWorkstation_WorkTable
- int WorkTableID

Additional Inherited Members

8.70.1 Detailed Description

Definition at line 29 of file [WorkTable.h](#).

8.70.2 Constructor & Destructor Documentation

8.70.2.1 [WorkTable::WorkTable](#) (std::string name)

Definition at line 21 of file [WorkTable.cpp](#).

8.70.2.2 [WorkTable::~WorkTable](#) ()

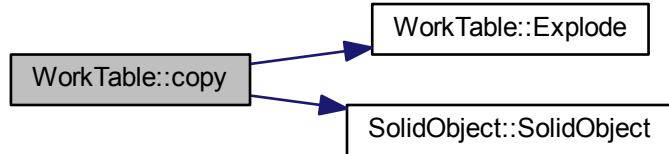
Definition at line 25 of file [WorkTable.cpp](#).

8.70.3 Member Function Documentation

8.70.3.1 [void WorkTable::copy](#) (std::map< std::string, std::string > object)

Definition at line 89 of file [WorkTable.cpp](#).

Here is the call graph for this function:



Here is the caller graph for this function:



8.70.3.2 std::vector< std::string > WorkTable::Explode (const std::string & str, char separator)

Definition at line 107 of file WorkTable.cpp.

Here is the caller graph for this function:

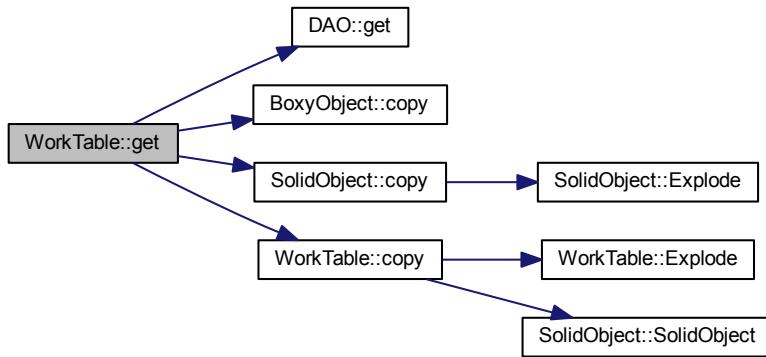


8.70.3.3 void WorkTable::get (int id)

8.70.3.4 void WorkTable::get (std::string name)

Definition at line 52 of file WorkTable.cpp.

Here is the call graph for this function:



8.70.3.5 `DAO * WorkTable::getdao ()`

Definition at line 34 of file `WorkTable.cpp`.

8.70.3.6 `std::vector< SolidObject * > WorkTable::gethadBySolidObject_WorkTable ()`

Definition at line 37 of file `WorkTable.cpp`.

8.70.3.7 `KittingWorkstation * WorkTable::gethasWorkstation_WorkTable ()`

Definition at line 40 of file `WorkTable.cpp`.

8.70.3.8 `int WorkTable::getWorkTableID ()`

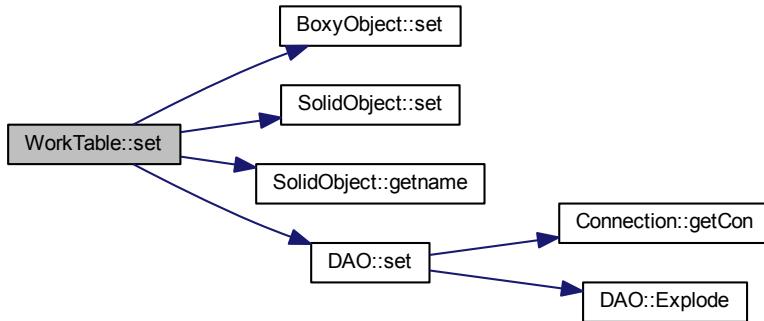
Definition at line 31 of file `WorkTable.cpp`.

8.70.3.9 `void WorkTable::set (int id, WorkTable * obj)`

8.70.3.10 `void WorkTable::set (std::string name)`

Definition at line 65 of file `WorkTable.cpp`.

Here is the call graph for this function:



8.70.3.11 void WorkTable::setdao (DAO * _dao)

Definition at line 43 of file WorkTable.cpp.

8.70.3.12 void WorkTable::sethadBySolidObject_WorkTable (std::vector< SolidObject * > _hadBySolidObject_WorkTable)

Definition at line 46 of file WorkTable.cpp.

8.70.3.13 void WorkTable::sethasWorkstation_WorkTable (KittingWorkstation * _hasWorkstation_WorkTable)

Definition at line 49 of file WorkTable.cpp.

8.70.4 Member Data Documentation

8.70.4.1 DAO* WorkTable::dao [private]

Definition at line 31 of file WorkTable.h.

8.70.4.2 std::vector<SolidObject*> WorkTable::hadBySolidObject_WorkTable [private]

Definition at line 32 of file WorkTable.h.

8.70.4.3 KittingWorkstation* WorkTable::hasWorkstation_WorkTable [private]

Definition at line 33 of file WorkTable.h.

8.70.4.4 int WorkTable::WorkTableID [private]

Definition at line 30 of file WorkTable.h.

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

- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[WorkTable.h](#)
- C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/[WorkTable.cpp](#)

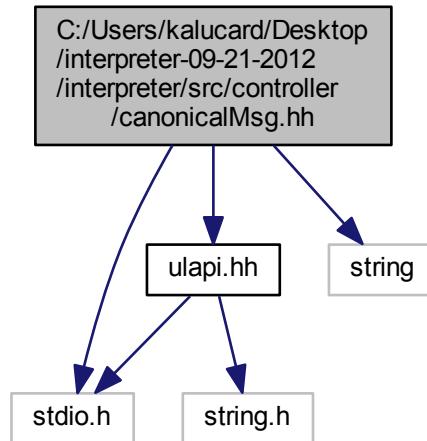
Chapter 9

File Documentation

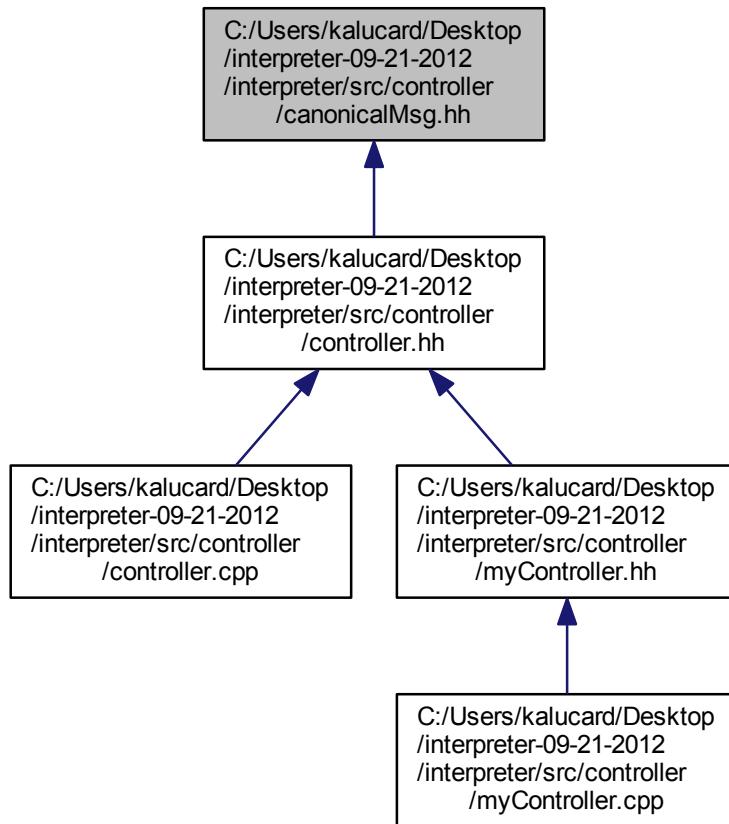
9.1 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/canonicalMsg.hh File Reference

Provide a generic class to base command processing routines for the canonical robot command language. This class will provide virtual routines that may be specialized for individual robot controllers.

```
#include <stdio.h>
#include "ulapi.hh"
#include <string>
Include dependency graph for canonicalMsg.hh:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct [CanonicalHdr](#)
- class [CanonicalMsg](#)
- class [CloseGripperMsg](#)
- class [DwellMsg](#)
- class [EndCanonMsg](#)
- class [InitCanonMsg](#)
- class [MessageMsg](#)
- class [MoveSmoothlyToMsg](#)
- class [MoveStraightToMsg](#)
- class [MoveToMsg](#)
- class [OpenGripperMsg](#)
- class [SetAbsoluteAccelerationMsg](#)
- class [SetAngleUnitsMsg](#)
- class [SetEndAngleToleranceMsg](#)

- class [SetEndPointToleranceMsg](#)
- class [SetIntermediatePointToleranceMsg](#)
- class [SetLengthUnitsMsg](#)
- class [SetRelativeAccelerationMsg](#)
- class [SetRelativeSpeedMsg](#)

Enumerations

- enum [CanonicalType](#) {
CloseGripper, *Dwell*, *EndCanon*, *InitCanon*,
Message, *MoveSmoothlyTo*, *MoveStraightTo*, *MoveTo*,
OpenGripper, *SetAbsoluteAcceleration*, *SetAngleUnits*, *SetEndAngleTolerance*,
SetEndPointTolerance, *SetIntermediatePointTolerance*, *SetLengthUnits*, *SetRelativeAcceleration*,
SetRelativeSpeed }

9.1.1 Detailed Description

Provide a generic class to base command processing routines for the canonical robot command language. This class will provide virtual routines that may be specialized for individual robot controllers.

Author

Stephen Balakirsky

Date

05/31/2012

Definition in file [canonicalMsg.hh](#).

9.1.2 Enumeration Type Documentation

9.1.2.1 enum CanonicalType

Enumerator:

CloseGripper
Dwell
EndCanon
InitCanon
Message
MoveSmoothlyTo
MoveStraightTo
MoveTo
OpenGripper
SetAbsoluteAcceleration
SetAngleUnits
SetEndAngleTolerance

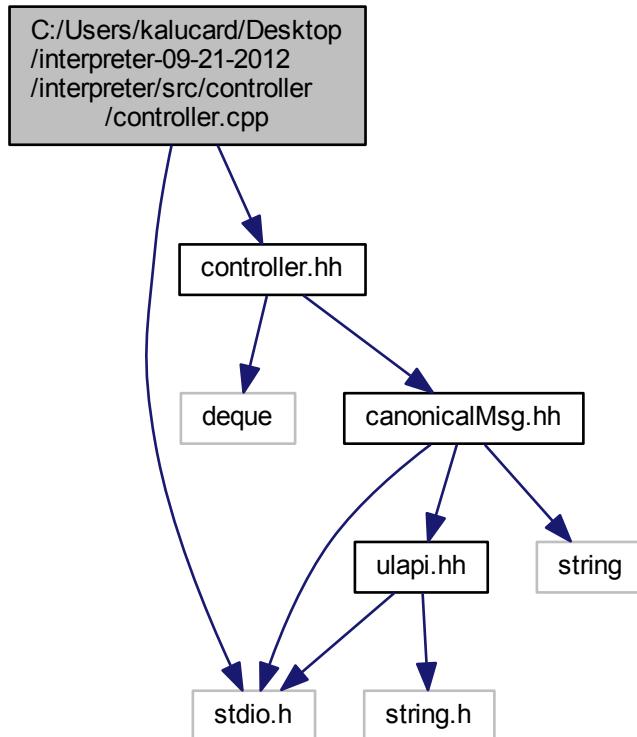
SetEndPointTolerance
SetIntermediatePointTolerance
SetLengthUnits
SetRelativeAcceleration
SetRelativeSpeed

Definition at line 26 of file canonicalMsg.hh.

9.2 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/controller.cpp

File Reference

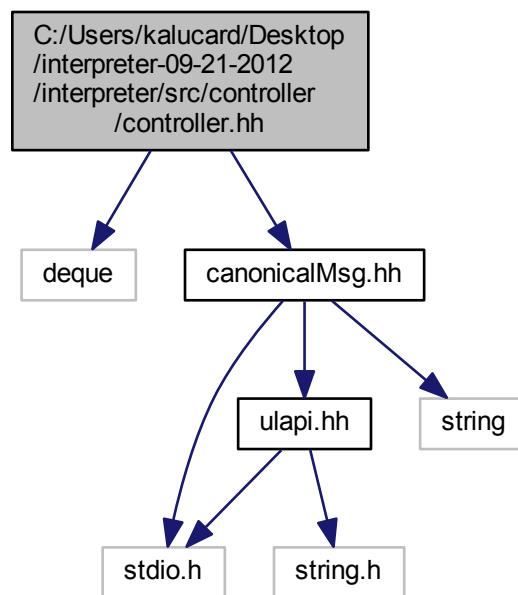
```
#include <stdio.h>
#include "controller.hh"
Include dependency graph for controller.cpp:
```



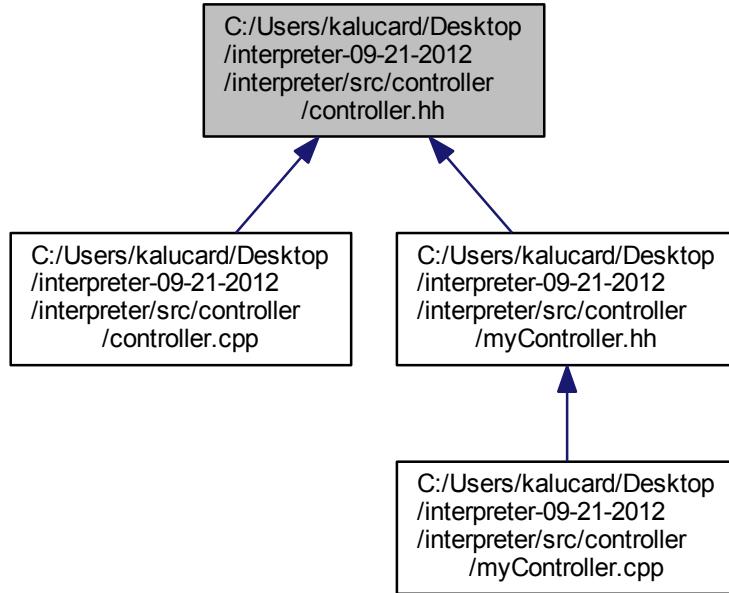
9.3 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/controller.hh File Reference

Provide a generic class to base executor controller modules off of. This class will provide routines to queue and dequeue commands as well as virtual functions for items such as emergency conditions and command handlers.

```
#include <deque>
#include "canonicalMsg.hh"
Include dependency graph for controller.hh:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [Controller](#)

9.3.1 Detailed Description

Provide a generic class to base executor controller modules off of. This class will provide routines to queue and dequeue commands as well as virtual functions for items such as emergency conditions and command handlers.

Author

Stephen Balakirsky

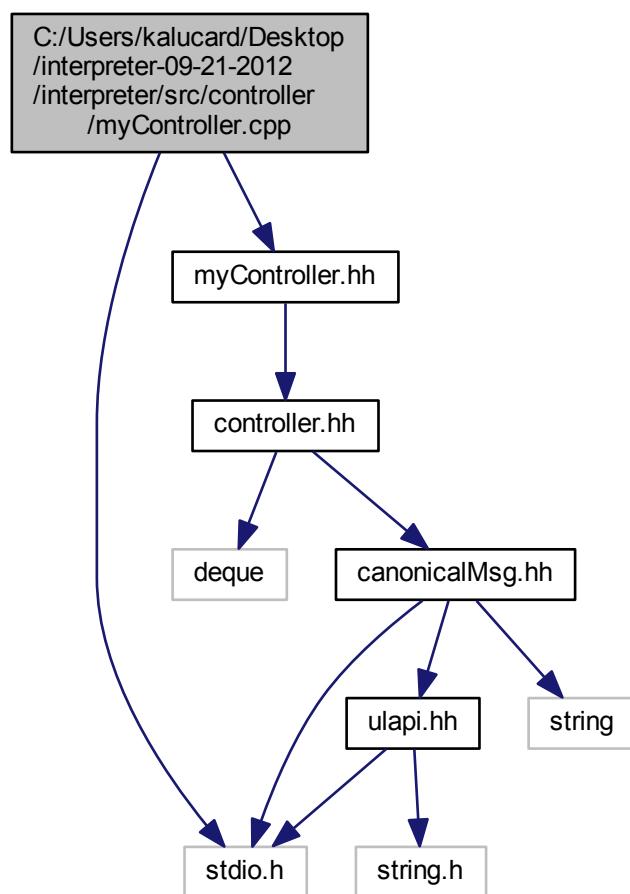
Date

05/31/2012

Definition in file [controller.hh](#).

9.4 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/myController.cpp File Reference

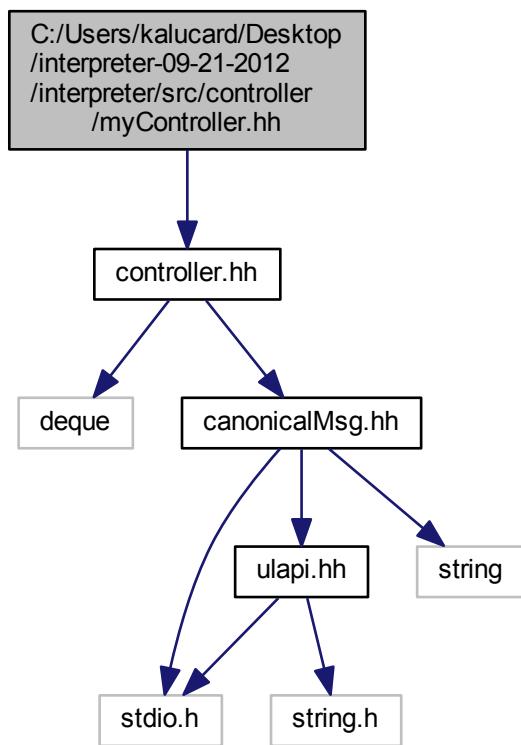
```
#include <stdio.h>
#include "myController.hh"
Include dependency graph for myController.cpp:
```



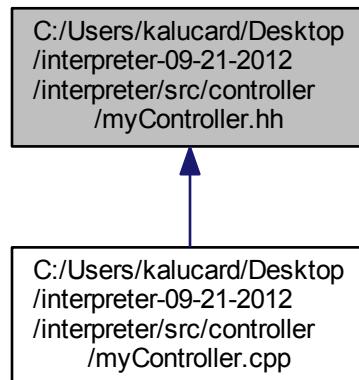
9.5 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/myController.hh File Reference

Include file for myController class.

```
#include "controller.hh"
Include dependency graph for myController.hh:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [MyController](#)

9.5.1 Detailed Description

Include file for myController class.

Author

Stephen Balakirsky

Date

05/31/2012

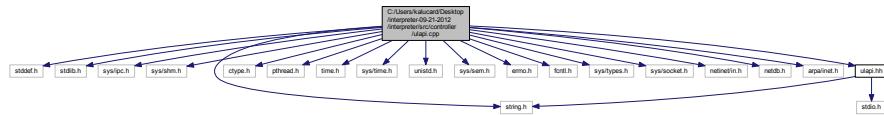
Definition in file [myController.hh](#).

9.6 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/ulapi.cpp

File Reference

```
#include <stddef.h>
#include <stdlib.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <string.h>
#include <ctype.h>
#include <pthread.h>
#include <time.h>
#include <sys/time.h>
#include <unistd.h>
#include <sys/sem.h>
#include <errno.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <arpa/inet.h>
#include "ulapi.hh"
```

Include dependency graph for ulapi.cpp:



Classes

- struct [unix_shm_struct](#)

TypeDefs

- typedef void *(*[pthread_task_code](#))(void *)

Functions

- [ulapi_result ulapi_exit \(void\)](#)
- [ulapi_result ulapi_init \(ulapi_integer sel\)](#)
- [ulapi_result ulapi_mutex_delete \(void *mutex\)](#)

- `ulapi_result ulapi_mutex_give (void *mutex)`
- `void * ulapi_mutex_new (ulapi_id key)`
- `ulapi_result ulapi_mutex_take (void *mutex)`
- `ulapi_prio ulapi_prio_highest (void)`
- `ulapi_prio ulapi_prio_lowest (void)`
- `ulapi_prio ulapi_prio_next_higher (ulapi_prio prio)`
- `ulapi_prio ulapi_prio_next_lower (ulapi_prio prio)`
- `ulapi_result ulapi_socket_close (ulapi_integer id)`
- `ulapi_integer ulapi_socket_get_client_id (ulapi_integer port, const char *hostname)`
- `ulapi_integer ulapi_socket_read (ulapi_integer id, char *buf, ulapi_integer len)`
- `ulapi_integer ulapi_socket_write (ulapi_integer id, const char *buf, ulapi_integer len)`
- `ulapi_result ulapi_task_delete (void *task)`
- `ulapi_result ulapi_task_init (void)`
- `void * ulapi_task_new (void)`
- `ulapi_result ulapi_task_pause (void *task)`
- `ulapi_result ulapi_task_resume (void *task)`
- `ulapi_result ulapi_task_set_period (void *task, ulapi_integer period_nsec)`
- `ulapi_result ulapi_task_start (void *task, void(*taskcode)(void *), void *taskarg, ulapi_prio prio, ulapi_integer period_nsec)`
- `ulapi_result ulapi_task_stop (void *task)`
- `ulapi_real ulapi_time (void)`
- `ulapi_result unix_ulapi_exit (void)`
- `ulapi_result unix_ulapi_fifo_delete (ulapi_integer key, ulapi_integer fd, ulapi_integer size)`
- `ulapi_result unix_ulapi_fifo_new (ulapi_integer key, ulapi_integer *fd, ulapi_integer size)`
- `ulapi_integer unix_ulapi_fifo_read (ulapi_integer fd, char *buf, ulapi_integer size)`
- `ulapi_integer unix_ulapi_fifo_write (ulapi_integer fd, const char *buf, ulapi_integer size)`
- `void * unix_ulapi_shm_addr (void *shm)`
- `ulapi_result unix_ulapi_shm_delete (void *shm)`
- `void * unix_ulapi_shm_new (ulapi_id key, ulapi_integer size)`

9.6.1 Typedef Documentation

9.6.1.1 `typedef void*(* pthread_task_code)(void *)`

Definition at line 299 of file ulapi.cpp.

9.6.2 Function Documentation

9.6.2.1 `ulapi_result ulapi_exit (void)`

Definition at line 249 of file ulapi.cpp.

9.6.2.2 `ulapi_result ulapi_init (ulapi_integer sel)`

Call this once before any other ULA API functions, passing `UL_USE_DEFAULT` for the default ULA API implementation, or one of the platform-specific values like `UL_USE_UNIX` or `UL_USE_RTAI` to force an implementation, if available. Returns `ULAPI_OK` if successful, otherwise `ULAPI_RESULT_IMPL_ERROR` if the implementation specified by `sel` is not available, or `ULAPI_RESULT_BAD_ARGS` if `sel` is not recognized.

Definition at line 46 of file ulapi.cpp.

9.6.2.3 `ulapi_result ulapi_mutex_delete(void * mutex)`

Deletes the mutex.

Definition at line 194 of file ulapi.cpp.

9.6.2.4 `ulapi_result ulapi_mutex_give(void * mutex)`

Definition at line 227 of file ulapi.cpp.

9.6.2.5 `void* ulapi_mutex_new(ulapi_id key)`

Returns a pointer to an implementation-defined structure that is passed to the other mutex functions, or NULL if no mutex can be created.

Definition at line 206 of file ulapi.cpp.

9.6.2.6 `ulapi_result ulapi_mutex_take(void * mutex)`

Takes the mutex, signifying that the associated shared resource will now be used by the task. If the mutex is already taken, this blocks the caller until the mutex is given.

Definition at line 235 of file ulapi.cpp.

9.6.2.7 `ulapi_prio ulapi_prio_highest(void)`

Definition at line 255 of file ulapi.cpp.

Here is the caller graph for this function:

**9.6.2.8 `ulapi_prio ulapi_prio_lowest(void)`**

Definition at line 261 of file ulapi.cpp.

Here is the caller graph for this function:



9.6.2.9 `ulapi_prio ulapi_prio_next_higher(ulapi_prio prio)`

Definition at line 267 of file ulapi.cpp.

Here is the call graph for this function:



9.6.2.10 `ulapi_prio ulapi_prio_next_lower(ulapi_prio prio)`

Definition at line 276 of file ulapi.cpp.

Here is the call graph for this function:



9.6.2.11 `ulapi_result ulapi_socket_close(ulapi_integer id)`

Closes the socket id, whether that for a client, for a server, or to a client, broadcast or otherwise.

Definition at line 243 of file ulapi.cpp.

9.6.2.12 `ulapi_integer ulapi_socket_get_client_id (ulapi_integer port, const char * host)`

Connects as a client to the socket server on *port* and *host*. Returns the integer socket descriptor for later sends and receives.

Definition at line 156 of file ulapi.cpp.

9.6.2.13 `ulapi_integer ulapi_socket_read (ulapi_integer id, char * buf, ulapi_integer len)`

Reads up to *len* bytes from socket *id* into *buf*. Returns the number of bytes read, or -1 on error.

Definition at line 144 of file ulapi.cpp.

9.6.2.14 `ulapi_integer ulapi_socket_write (ulapi_integer id, const char * buf, ulapi_integer len)`

Writes *len* bytes from *buf* to socket *id*. Returns the number of bytes written, or -1 on error.

Definition at line 150 of file ulapi.cpp.

9.6.2.15 `ulapi_result ulapi_task_delete (void * task)`

Definition at line 291 of file ulapi.cpp.

9.6.2.16 `ulapi_result ulapi_task_init (void)`

Definition at line 347 of file ulapi.cpp.

9.6.2.17 `void* ulapi_task_new (void)`

Allocates space for a platform-specific data structure that holds the task information. Pass this to the *ulapi_task_* functions.

Definition at line 285 of file ulapi.cpp.

9.6.2.18 `ulapi_result ulapi_task_pause (void * task)`

Definition at line 329 of file ulapi.cpp.

9.6.2.19 `ulapi_result ulapi_task_resume (void * task)`

Definition at line 335 of file ulapi.cpp.

9.6.2.20 `ulapi_result ulapi_task_set_period (void * task, ulapi_integer period_nsec)`

Definition at line 341 of file ulapi.cpp.

9.6.2.21 `ulapi_result ulapi_task_start (void * task, void(*)(void *) taskcode, void * taskarg, ulapi_prio prio, ulapi_integer period_nsec)`

Definition at line 302 of file ulapi.cpp.

9.6.2.22 `ulapi_result ulapi_task_stop (void * task)`

Definition at line 322 of file ulapi.cpp.

9.6.2.23 `ulapi_real ulapi_time (void)`

Definition at line 382 of file ulapi.cpp.

Here is the caller graph for this function:



9.6.2.24 `ulapi_result unix_ulapi_exit (void)`

Definition at line 53 of file ulapi.cpp.

9.6.2.25 `ulapi_result unix_ulapi_fifo_delete (ulapi_integer key, ulapi_integer fd, ulapi_integer size)`

Definition at line 125 of file ulapi.cpp.

9.6.2.26 `ulapi_result unix_ulapi_fifo_new (ulapi_integer key, ulapi_integer * fd, ulapi_integer size)`

Definition at line 118 of file ulapi.cpp.

9.6.2.27 `ulapi_integer unix_ulapi_fifo_read (ulapi_integer fd, char * buf, ulapi_integer size)`

Definition at line 138 of file ulapi.cpp.

9.6.2.28 `ulapi_integer unix_ulapi_fifo_write (ulapi_integer fd, const char * buf, ulapi_integer size)`

Definition at line 132 of file ulapi.cpp.

9.6.2.29 `void* unix_ulapi_shm_addr (void * shm)`

Definition at line 95 of file ulapi.cpp.

9.6.2.30 `ulapi_result unix_ulapi_shm_delete (void * shm)`

Definition at line 101 of file ulapi.cpp.

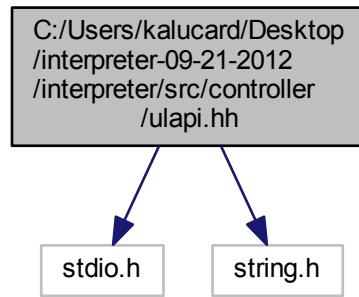
9.6.2.31 `void* unix_ulapi_shm_new(ulapi_id key, ulapi_integer size)`

Definition at line 67 of file ulapi.cpp.

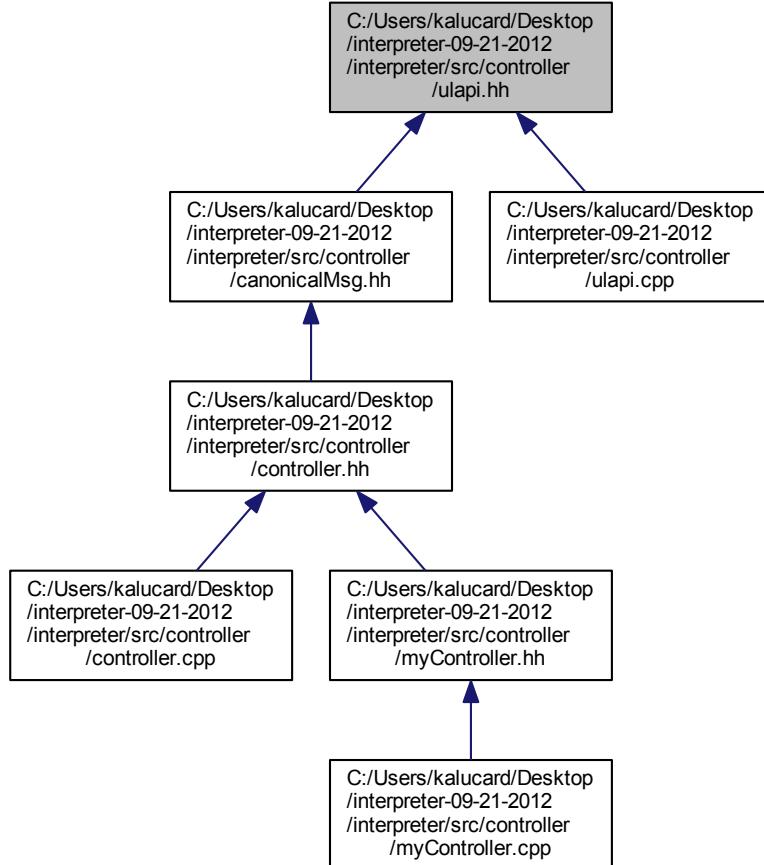
9.7 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/ulapi.hh File Reference

Declarations of the user-level application programming interface, ULA API, for portable communication between user-level processes and real-time tasks conforming to the RTAPI real-time API.

```
#include <stdio.h>
#include <string.h>
Include dependency graph for ulapi.hh:
```



This graph shows which files directly or indirectly include this file:



Macros

- #define DIGITS_IN(x) (sizeof(x) * 3 + 1)
- #define ULAPIO_DECL_SHARED
- #define ulapi_print printf
- #define ulapi_snprintf snprintf
- #define ulapi_sscanf sscanf
- #define ulapi_strcasecmp strcasecmp
- #define ulapi_strdup strdup
- #define ulapi_strncpy strncpy

Typedefs

- typedef char ulapi_flag
- typedef int ulapi_id

- `typedef int ulapi_integer`
- `typedef int ulapi_prio`
- `typedef double ulapi_real`
- `typedef int ulapi_result`
- `typedef void(* ulapi_task_code)(void *)`

Enumerations

- `enum { UL_USE_DEFAULT = 0, UL_USE_UNIX, UL_USE_RTAI }`
- `enum { ULAPIOK = 0, ULAPI_ERROR, ULAPI_IMPL_ERROR, ULAPI_BAD_ARGS }`
- `enum { ULAPI_DEBUG_WARN = 0x01, ULAPI_DEBUG_ERROR = 0x02, ULAPI_DEBUG_ALL = 0xFF }`
- `enum ulapi_stdio { ULAPI_STDIN, ULAPI_STDOUT, ULAPI_STDERR }`

Functions

- `char * ulapi_address_to_hostname (ulapi_integer address)`
- `ulapi_result ulapi_cond_broadcast (void *cond)`
- `ulapi_result ulapi_cond_delete (void *cond)`
- `void * ulapi_cond_new (ulapi_id key)`
- `ulapi_result ulapi_cond_signal (void *cond)`
- `ulapi_result ulapi_cond_wait (void *cond, void *mutex)`
- `ulapi_result ulapi_exit (void)`
- `ulapi_result ulapi_fd_close (void *id)`
- `ulapi_result ulapi_fd_delete (void *id)`
- `void * ulapi_fd_new (void)`
- `ulapi_result ulapi_fd_open (const char *port, void *id)`
- `ulapi_integer ulapi_fd_read (void *id, char *buf, ulapi_integer len)`
- `ulapi_result ulapi_fd_set_blocking (void *id)`
- `ulapi_result ulapi_fd_set_nonblocking (void *id)`
- `ulapi_integer ulapi_fd_write (void *id, const char *buf, ulapi_integer len)`
- `ulapi_result ulapi_fifo_delete (ulapi_integer key, ulapi_integer fd, ulapi_integer size)`
- `ulapi_result ulapi_fifo_new (ulapi_integer key, ulapi_integer *fd, ulapi_integer size)`
- `ulapi_integer ulapi_fifo_read (ulapi_integer fd, char *buf, ulapi_integer size)`
- `ulapi_integer ulapi_fifo_write (ulapi_integer fd, const char *buf, ulapi_integer size)`
- `void ulapi_free_argv (ulapi_integer argc, char **argv)`
- `ulapi_integer ulapi_get_host_address (void)`
- `ulapi_integer ulapi_hostname_to_address (const char *hostname)`
- `ulapi_result ulapi_init (ulapi_integer sel)`
- `ulapi_result ulapi_mutex_delete (void *mutex)`
- `ulapi_result ulapi_mutex_give (void *mutex)`
- `void * ulapi_mutex_new (ulapi_id key)`
- `ulapi_result ulapi_mutex_take (void *mutex)`
- `ulapi_prio ulapi_prio_highest (void)`
- `ulapi_prio ulapi_prio_lowest (void)`
- `ulapi_prio ulapi_prio_next_higher (ulapi_prio prio)`
- `ulapi_prio ulapi_prio_next_lower (ulapi_prio prio)`
- `ulapi_result ulapi_self_set_period (ulapi_integer period_nsec)`
- `ulapi_result ulapi_sem_delete (void *sem)`
- `ulapi_result ulapi_sem_give (void *sem)`
- `void * ulapi_sem_new (ulapi_id key)`

- `ulapi_result ulapi_sem_take (void *sem)`
- `ulapi_result ulapi_serial_close (void *id)`
- `ulapi_result ulapi_serial_delete (void *id)`
- `void * ulapi_serial_new (void)`
- `ulapi_result ulapi_serial_open (const char *port, void *id)`
- `ulapi_integer ulapi_serial_read (void *id, char *buf, ulapi_integer len)`
- `ulapi_result ulapi_serial_set_blocking (void *id)`
- `ulapi_result ulapi_serial_set_nonblocking (void *id)`
- `ulapi_integer ulapi_serial_write (void *id, const char *buf, ulapi_integer len)`
- `void ulapi_set_debug (ulapi_integer mask)`
- `void * ulapi_shm_addr (void *shm)`
- `ulapi_result ulapi_shm_delete (void *shm)`
- `void * ulapi_shm_new (ulapi_id key, ulapi_integer size)`
- `void ulapi_sleep (ulapi_real secs)`
- `ulapi_integer ulapi_socket_broadcast (ulapi_integer id, ulapi_integer port, const char *buf, ulapi_integer len)`
- `ulapi_result ulapi_socket_close (ulapi_integer id)`
- `ulapi_integer ulapi_socket_get_broadcastee_id (ulapi_integer port)`
- `ulapi_integer ulapi_socket_get_broadcaster_id (void)`
- `ulapi_integer ulapi_socket_get_client_id (ulapi_integer port, const char *host)`
- `ulapi_integer ulapi_socket_get_connection_id (ulapi_integer id)`
- `ulapi_integer ulapi_socket_get_server_id (ulapi_integer port)`
- `ulapi_integer ulapi_socket_read (ulapi_integer id, char *buf, ulapi_integer len)`
- `ulapi_result ulapi_socket_set_blocking (ulapi_integer id)`
- `ulapi_result ulapi_socket_set_nonblocking (ulapi_integer id)`
- `ulapi_integer ulapi_socket_write (ulapi_integer id, const char *buf, ulapi_integer len)`
- `ulapi_result ulapi_std_open (ulapi_stdio io, void *id)`
- `ulapi_result ulapi_system (const char *prog, int *result)`
- `ulapi_result ulapi_task_delete (void *task)`
- `ulapi_result ulapi_task_exit (void)`
- `ulapi_integer ulapi_task_id (void)`
- `ulapi_result ulapi_task_init (void)`
- `ulapi_result ulapi_task_join (void *task)`
- `void * ulapi_task_new (void)`
- `ulapi_result ulapi_task_pause (void *task)`
- `ulapi_result ulapi_task_resume (void *task)`
- `ulapi_result ulapi_task_set_period (void *task, ulapi_integer period_nsec)`
- `ulapi_result ulapi_task_start (void *task, void(*taskcode)(void *), void *taskarg, ulapi_prio prio, ulapi_integer period_nsec)`
- `ulapi_result ulapi_task_stop (void *task)`
- `ulapi_real ulapi_time (void)`
- `ulapi_integer ulapi_to_argv (const char *str, char ***argv)`
- `ulapi_result ulapi_wait (ulapi_integer period_nsec)`

9.7.1 Detailed Description

Declarations of the user-level application programming interface, ULAPIO, for portable communication between user-level processes and real-time tasks conforming to the RTAPI real-time API.

Definition in file [ulapi.hh](#).

9.7.2 Macro Definition Documentation

9.7.2.1 `#define DIGITS_IN(x) (sizeof(x) * 3 + 1)`

Definition at line 36 of file ulapi.hh.

9.7.2.2 `#define ULAPI_DECL_SHARED`

Definition at line 403 of file ulapi.hh.

9.7.2.3 `#define ulapi_print printf`

Definition at line 416 of file ulapi.hh.

9.7.2.4 `#define ulapi_snprintf snprintf`

Definition at line 424 of file ulapi.hh.

9.7.2.5 `#define ulapi_sscanf sscanf`

Definition at line 427 of file ulapi.hh.

9.7.2.6 `#define ulapi_strcasecmp strcasecmp`

Definition at line 425 of file ulapi.hh.

9.7.2.7 `#define ulapi_strdup strdup`

Definition at line 426 of file ulapi.hh.

9.7.2.8 `#define ulapi_strncpy strncpy`

Definition at line 428 of file ulapi.hh.

9.7.3 Typedef Documentation

9.7.3.1 `typedef char ulapi_flag`

Definition at line 60 of file ulapi.hh.

9.7.3.2 `typedef int ulapi_id`

Definition at line 54 of file ulapi.hh.

9.7.3.3 `typedef int ulapi_integer`

Definition at line 48 of file ulapi.hh.

9.7.3.4 `typedef int ulapi_prio`

As with RTAPI, priorities range from `ulapi_prio_lowest()` to `ulapi_prio_highest()`, inclusive. To use this API, use one of two methods:

Set your lowest priority task to `ulapi_prio_lowest()`, and for each task of the next lowest priority, set their priorities to `ulapi_prio_next_higher(previous)`.

Or,

Set your highest priority task to `ulapi_prio_highest()`, and for each task of the next highest priority, set their priorities to `ulapi_prio_next_lower(previous)`.

Definition at line 135 of file ulapi.hh.

9.7.3.5 `typedef double ulapi_real`

Definition at line 68 of file ulapi.hh.

9.7.3.6 `typedef int ulapi_result`

Definition at line 42 of file ulapi.hh.

9.7.3.7 `typedef void(* ulapi_task_code)(void *)`

Definition at line 150 of file ulapi.hh.

9.7.4 Enumeration Type Documentation

9.7.4.1 anonymous enum

Enumerator:

```
UL_USE_DEFAULT
UL_USE_UNIX
UL_USE_RTAI
```

Definition at line 71 of file ulapi.hh.

9.7.4.2 anonymous enum

Enumerator:

```
ULAPI_OK
ULAPI_ERROR
ULAPI_IMPL_ERROR
ULAPI_BAD_ARGS
```

Definition at line 82 of file ulapi.hh.

9.7.4.3 anonymous enum

Set and get the debug level.

Enumerator:

ULAPI_DEBUG_WARN
ULAPI_DEBUG_ERROR
ULAPI_DEBUG_ALL

Definition at line 108 of file ulapi.hh.

9.7.4.4 enum ulapi_stdio

Enumerator:

ULAPI_STDIN
ULAPI_STDOUT
ULAPI_STDERR

Definition at line 312 of file ulapi.hh.

9.7.5 Function Documentation

9.7.5.1 `char* ulapi_address_to_hostname (ulapi_integer address)`

9.7.5.2 `ulapi_result ulapi_cond_broadcast (void * cond)`

9.7.5.3 `ulapi_result ulapi_cond_delete (void * cond)`

Deletes the condition variable.

9.7.5.4 `void* ulapi_cond_new (ulapi_id key)`

Returns a pointer to an implementation-defined structure that is passed to the other condition variable functions, or NULL if no condition variable can be created.

9.7.5.5 `ulapi_result ulapi_cond_signal (void * cond)`

9.7.5.6 `ulapi_result ulapi_cond_wait (void * cond, void * mutex)`

Waits until the condition variable has reached its release value

9.7.5.7 `ulapi_result ulapi_exit (void)`

Definition at line 249 of file ulapi.cpp.

9.7.5.8 `ulapi_result ulapi_fd_close (void * id)`

Closes the file descriptor.

9.7.5.9 `ulapi_result ulapi_fd_delete (void * id)`

Deallocates a previously allocated file identifier.

9.7.5.10 `void* ulapi_fd_new (void)`

Allocates space for a file identifier, and returns its pointer.

9.7.5.11 `ulapi_result ulapi_fd_open (const char * port, void * id)`

Opens a connection to a file. Fills in the identifier and returns ULAPIO_OK if it worked, otherwise returns ULAPIO_ERROR.

9.7.5.12 `ulapi_integer ulapi_fd_read (void * id, char * buf, ulapi_integer len)`

Reads up to *len* bytes from file descriptor *id* into *buf*. Returns the number of bytes read, or -1 on error.

9.7.5.13 `ulapi_result ulapi_fd_set_blocking (void * id)`**9.7.5.14 `ulapi_result ulapi_fd_set_nonblocking (void * id)`**

Sets the file descriptor *id* to be blocking or nonblocking.

9.7.5.15 `ulapi_integer ulapi_fd_write (void * id, const char * buf, ulapi_integer len)`

Writes *len* bytes from *buf* to file descriptor *id*. Returns the number of bytes written, or -1 on error.

9.7.5.16 `ulapi_result ulapi_fifo_delete (ulapi_integer key, ulapi_integer fd, ulapi_integer size)`**9.7.5.17 `ulapi_result ulapi_fifo_new (ulapi_integer key, ulapi_integer * fd, ulapi_integer size)`****9.7.5.18 `ulapi_integer ulapi_fifo_read (ulapi_integer fd, char * buf, ulapi_integer size)`****9.7.5.19 `ulapi_integer ulapi_fifo_write (ulapi_integer fd, const char * buf, ulapi_integer size)`****9.7.5.20 `void ulapi_free_argv (ulapi_integer argc, char ** argv)`****9.7.5.21 `ulapi_integer ulapi_get_host_address (void)`****9.7.5.22 `ulapi_integer ulapi_hostname_to_address (const char * hostname)`**

9.7.5.23 `ulapi_result ulapi_init(ulapi_integer sel)`

Call this once before any other ULA API functions, passing UL_USE_DEFAULT for the default ULA API implementation, or one of the platform-specific values like UL_USE_UNIX or UL_USE_RTAI to force an implementation, if available. Returns ULA API_OK if successful, otherwise ULA API_RESULT_IMPL_ERROR if the implementation specified by *sel* is not available, or ULA API_RESULT_BAD_ARGS if *sel* is not recognized.

Definition at line 46 of file ulapi.cpp.

9.7.5.24 `ulapi_result ulapi_mutex_delete(void * mutex)`

Deletes the mutex.

Definition at line 194 of file ulapi.cpp.

9.7.5.25 `ulapi_result ulapi_mutex_give(void * mutex)`

Definition at line 227 of file ulapi.cpp.

9.7.5.26 `void* ulapi_mutex_new(ulapi_id key)`

Returns a pointer to an implementation-defined structure that is passed to the other mutex functions, or NULL if no mutex can be created.

Definition at line 206 of file ulapi.cpp.

9.7.5.27 `ulapi_result ulapi_mutex_take(void * mutex)`

Takes the mutex, signifying that the associated shared resource will now be used by the task. If the mutex is already taken, this blocks the caller until the mutex is given.

Definition at line 235 of file ulapi.cpp.

9.7.5.28 `ulapi_prio ulapi_prio_highest(void)`

Definition at line 255 of file ulapi.cpp.

Here is the caller graph for this function:



9.7.5.29 ulapi_prio ulapi_prio_lowest (void)

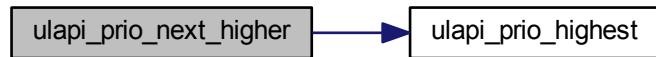
Definition at line 261 of file ulapi.cpp.

Here is the caller graph for this function:

**9.7.5.30 ulapi_prio ulapi_prio_next_higher (ulapi_prio prio)**

Definition at line 267 of file ulapi.cpp.

Here is the call graph for this function:

**9.7.5.31 ulapi_prio ulapi_prio_next_lower (ulapi_prio prio)**

Definition at line 276 of file ulapi.cpp.

Here is the call graph for this function:

**9.7.5.32 ulapi_result ulapi_self_set_period (ulapi_integer period_nsec)**

9.7.5.33 **ulapi_result ulapi_sem_delete (void * sem)**

9.7.5.34 **ulapi_result ulapi_sem_give (void * sem)**

9.7.5.35 **void* ulapi_sem_new (ulapi_id key)**

9.7.5.36 **ulapi_result ulapi_sem_take (void * sem)**

9.7.5.37 **ulapi_result ulapi_serial_close (void * id)**

Closes the serial port descriptor.

9.7.5.38 **ulapi_result ulapi_serial_delete (void * id)**

Deallocates a previously allocated serial port identifier.

9.7.5.39 **void* ulapi_serial_new (void)**

Allocates space for a serial port identifier, and returns its pointer.

9.7.5.40 **ulapi_result ulapi_serial_open (const char * port, void * id)**

Opens a connection to a serial port. Fills in the identifier and returns ULAPIOK if it worked, otherwise returns ULAPI_ERROR.

9.7.5.41 **ulapi_integer ulapi_serial_read (void * id, char * buf, ulapi_integer len)**

Reads up to *len* bytes from serial port descriptor *id* into *buf*. Returns the number of bytes read, or -1 on error.

9.7.5.42 **ulapi_result ulapi_serial_set_blocking (void * id)**

9.7.5.43 **ulapi_result ulapi_serial_set_nonblocking (void * id)**

Sets the serial port descriptor *id* to be blocking or nonblocking.

9.7.5.44 **ulapi_integer ulapi_serial_write (void * id, const char * buf, ulapi_integer len)**

Writes *len* bytes from *buf* to serial port descriptor *id*. Returns the number of bytes written, or -1 on error.

9.7.5.45 **void ulapi_set_debug (ulapi_integer mask)**

9.7.5.46 **void* ulapi_shm_addr (void * shm)**

Returns a pointer to the actual shared memory, given a shared memory data structure previously created with *ulapi_shm_new*.

9.7.5.47 ulapi_result ulapi_shm_delete(void *shm)

Deletes shared memory previously allocated with *ulapi_shm_new*.

9.7.5.48 void* ulapi_shm_new(ulapi_id key, ulapi_integer size)

Allocates space for a platform-specific data structure that holds the shared memory configuration. Pass this to *ulapi_shm_addr* to get a pointer to the actual shared memory.

9.7.5.49 void ulapi_sleep(ulapi_real secs)**9.7.5.50 ulapi_integer ulapi_socket_broadcast(ulapi_integer id, ulapi_integer port, const char *buf, ulapi_integer len)**

Broadcasts *len* bytes from *buf* to socket *id* using port -a port. Returns the number of bytes written, or -1 on error.

9.7.5.51 ulapi_result ulapi_socket_close(ulapi_integer id)

Closes the socket id, whether that for a client, for a server, or to a client, broadcast or otherwise.

Definition at line 243 of file ulapi.cpp.

9.7.5.52 ulapi_integer ulapi_socket_get_broadcastee_id(ulapi_integer port)

Gets an fd for broadcast reading.

9.7.5.53 ulapi_integer ulapi_socket_get_broadcaster_id(void)

Gets an fd for broadcast writing. No port is necessary; the port is provided when writing via *ulapi_socket_broadcast*.

9.7.5.54 ulapi_integer ulapi_socket_get_client_id(ulapi_integer port, const char *host)

Connects as a client to the socket server on *port* and *host*. Returns the integer socket descriptor for later sends and receives.

Definition at line 156 of file ulapi.cpp.

9.7.5.55 ulapi_integer ulapi_socket_get_connection_id(ulapi_integer id)

Called by a server to gets a connection from a client. Returns the integer socket descriptor for later sends and receives.

9.7.5.56 ulapi_integer ulapi_socket_get_server_id(ulapi_integer port)

Creates a server connection to the *port*. Returns the integer socket descriptor for later use in *ulapi_socket_get_client_connection*.

9.7.5.57 `ulapi_integer ulapi_socket_read (ulapi_integer id, char * buf, ulapi_integer len)`

Reads up to *len* bytes from socket *id* into *buf*. Returns the number of bytes read, or -1 on error.

Definition at line 144 of file ulapi.cpp.

9.7.5.58 `ulapi_result ulapi_socket_set_blocking (ulapi_integer id)`

9.7.5.59 `ulapi_result ulapi_socket_set_nonblocking (ulapi_integer id)`

Sets the socket to be blocking or nonblocking.

9.7.5.60 `ulapi_integer ulapi_socket_write (ulapi_integer id, const char * buf, ulapi_integer len)`

Writes *len* bytes from *buf* to socket *id*. Returns the number of bytes written, or -1 on error.

Definition at line 150 of file ulapi.cpp.

9.7.5.61 `ulapi_result ulapi_std_open (ulapi_stdio io, void * id)`

Copies the stdio file descriptor into *id*.

Parameters

<i>io</i>	one of ULAPIO_STDIN,STDOUT,STDERR
<i>id</i>	pointer to fd where it will be copied

9.7.5.62 `ulapi_result ulapi_system (const char * prog, int * result)`

Executes 'prog'. Returns ULAPIO_OK if it was able to execute, placing the program return value in 'result'. Otherwise, returns ULAPIO_ERROR and leaves 'result' alone.

9.7.5.63 `ulapi_result ulapi_task_delete (void * task)`

Definition at line 291 of file ulapi.cpp.

9.7.5.64 `ulapi_result ulapi_task_exit (void)`

9.7.5.65 `ulapi_integer ulapi_task_id (void)`

9.7.5.66 `ulapi_result ulapi_task_init (void)`

Definition at line 347 of file ulapi.cpp.

9.7.5.67 `ulapi_result ulapi_task_join (void * task)`

9.7.5.68 `void* ulapi_task_new (void)`

Allocates space for a platform-specific data structure that holds the task information. Pass this to the `ulapi_task_` functions.

Definition at line 285 of file ulapi.cpp.

9.7.5.69 `ulapi_result ulapi_task_pause (void * task)`

Definition at line 329 of file ulapi.cpp.

9.7.5.70 `ulapi_result ulapi_task_resume (void * task)`

Definition at line 335 of file ulapi.cpp.

9.7.5.71 `ulapi_result ulapi_task_set_period (void * task, ulapi_integer period_nsec)`

Definition at line 341 of file ulapi.cpp.

9.7.5.72 `ulapi_result ulapi_task_start (void * task, void(*)(void *) taskcode, void * taskarg, ulapi_prio prio, ulapi_integer period_nsec)`

Definition at line 302 of file ulapi.cpp.

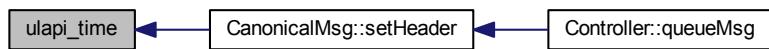
9.7.5.73 `ulapi_result ulapi_task_stop (void * task)`

Definition at line 322 of file ulapi.cpp.

9.7.5.74 `ulapi_real ulapi_time (void)`

Definition at line 382 of file ulapi.cpp.

Here is the caller graph for this function:



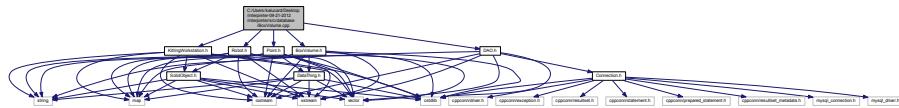
9.7.5.75 `ulapi_integer ulapi_to_argv (const char * str, char *** argv)`

9.7.5.76 `ulapi_result ulapi_wait (ulapi_integer period_nsec)`

9.8 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/BoxVolume.cpp

```
#include "BoxVolume.h"
#include "KittingWorkstation.h"
#include "Robot.h"
#include "DAO.h"
#include "Point.h"
Include dependency graph for BoxVolume.cpp:
```

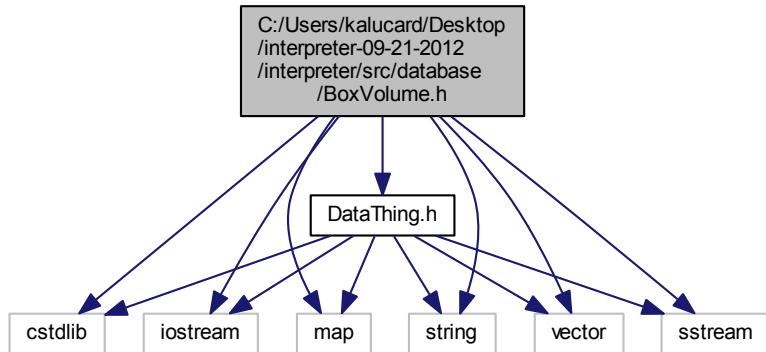
Include dependency graph for BoxVolume.cpp:



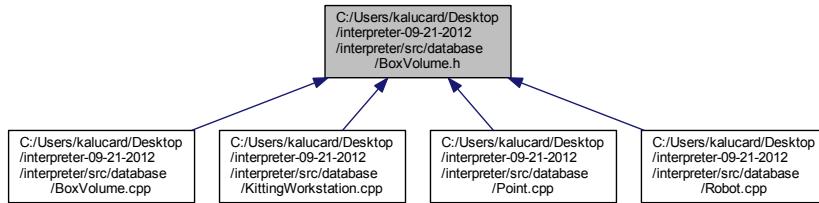
9.9 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/BoxVolume.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "DataThing
```

Include dependency graph for BoxVolume.h



This graph shows which files directly or indirectly include this file:



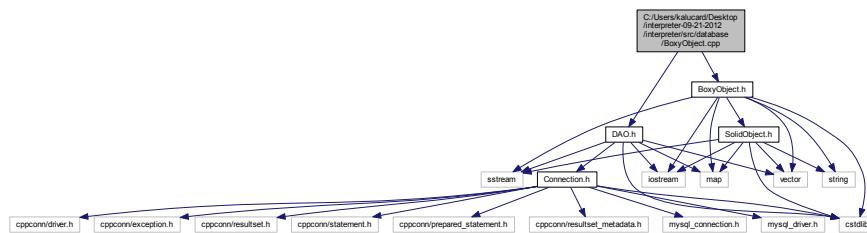
Classes

- class [BoxVolume](#)

9.10 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/BoxyObject.cpp File Reference

```
#include "BoxyObject.h"
#include "DAO.h"
```

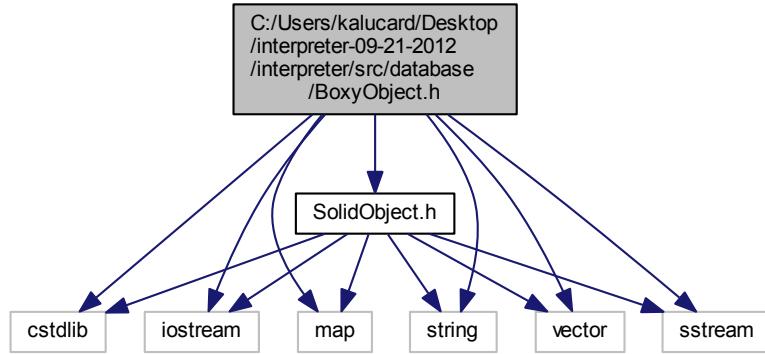
Include dependency graph for BoxyObject.cpp:



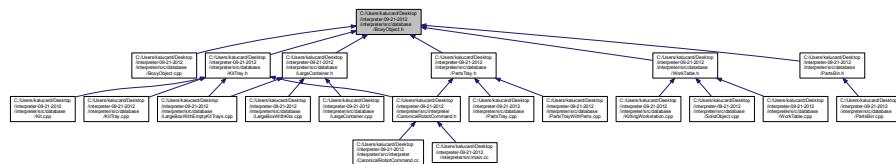
9.11 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/BoxyObject.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "SolidObject.h"
```

Include dependency graph for BoxyObject.h:



This graph shows which files directly or indirectly include this file:



Classes

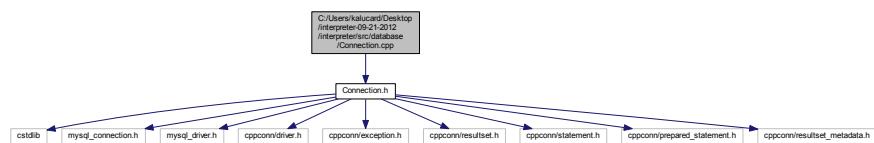
- class [BoxyObject](#)

9.12 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Connection.cpp

File Reference

```
#include "Connection.h"
```

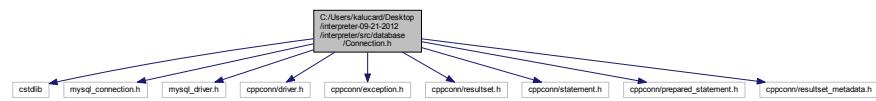
Include dependency graph for Connection.cpp:



9.13 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Connection.h File Reference

```
#include <cstdlib>
#include "mysql_connection.h"
#include "mysql_driver.h"
#include <cppconn/driver.h>
#include <cppconn/exception.h>
#include <cppconn/resultset.h>
#include <cppconn/statement.h>
#include <cppconn/prepared_statement.h>
#include <cppconn/resultset_metadata.h>
```

Include dependency graph for Connection.h:



This graph shows which files directly or indirectly include this file:



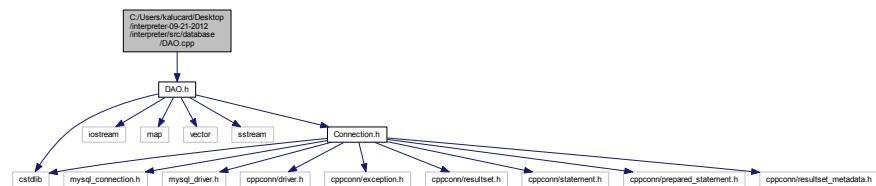
Classes

- class [Connection](#)

9.14 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/DAO.cpp File Reference

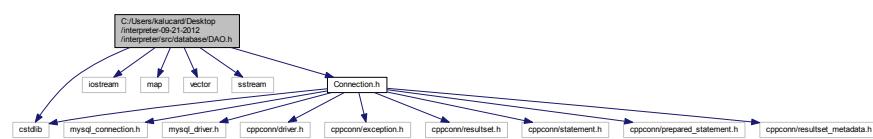
```
#include "DAO.h"
```

Include dependency graph for DAO.cpp:



9.15 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/DAO.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <vector>
#include <sstream>
#include "Connection.h"
Include dependency graph for DAO.h:
```



This graph shows which files directly or indirectly include this file:

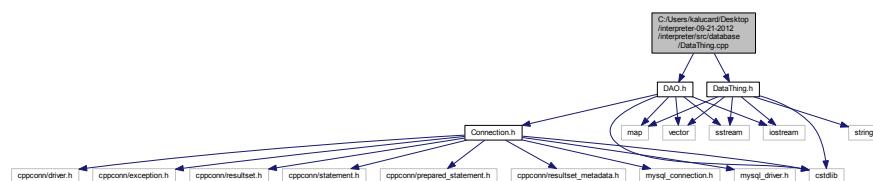


Classes

- class [DAO](#)

9.16 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/DataThing.cpp File Reference

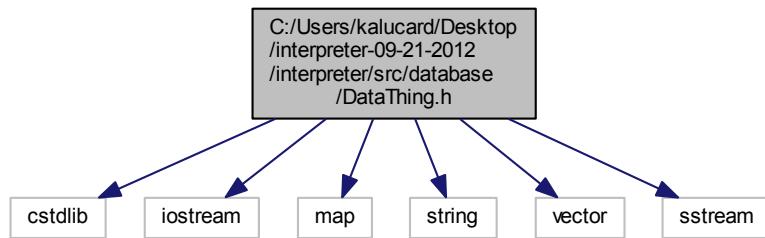
```
#include "DataThing.h"
#include "DAO.h"
Include dependency graph for DataThing.cpp:
```



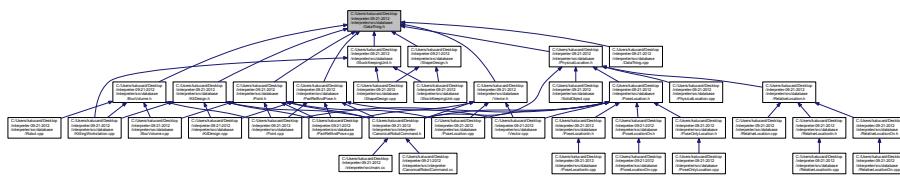
9.17 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/DataThing.h File Reference

```
#include <cstdlib>
```

```
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
Include dependency graph for DataThing.h:
```



This graph shows which files directly or indirectly include this file:

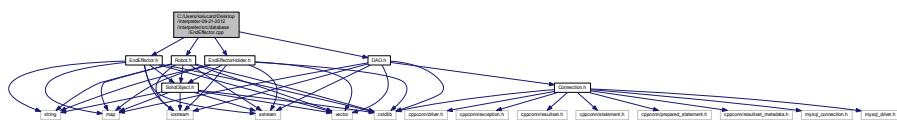


Classes

- class [DataThing](#)

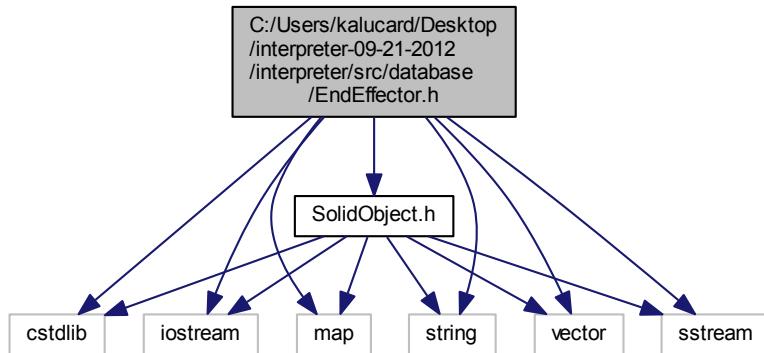
9.18 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffector.cpp File Reference

```
#include "EndEffector.h"
#include "Robot.h"
#include "DAO.h"
#include "EndEffectorHolder.h"
Include dependency graph for EndEffector.cpp:
```

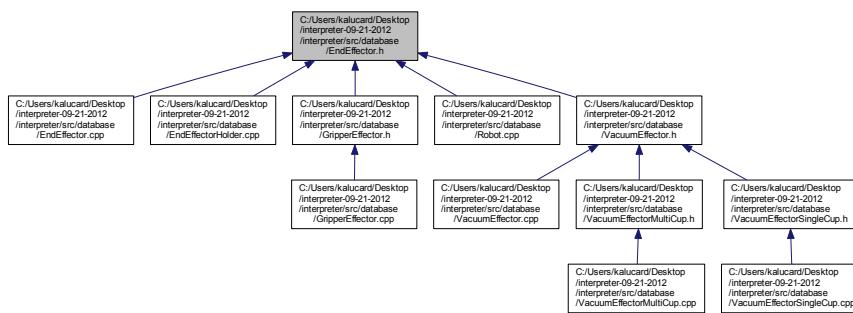


9.19 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffector.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "SolidObject.h"
Include dependency graph for EndEffector.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [EndEffector](#)

9.20

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffectorChangingStation.cpp

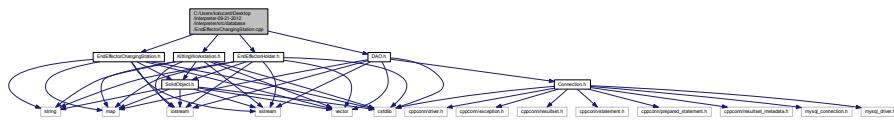
File Reference

431

File Reference 431
9.20 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffector-
ChangingStation.cpp File Reference

```
#include "EndEffectorChangingStation.h"
#include "KittingWorkstation.h"
#include "DAO.h"
#include "EndEffectorHolder.h"
Include dependency graph for EndEffectorChangingStation.cpp:
```

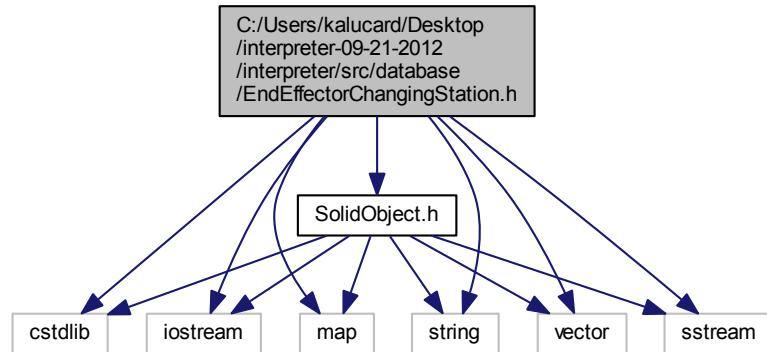
Include dependency graph for EndEffectorChangingStation.cpp:



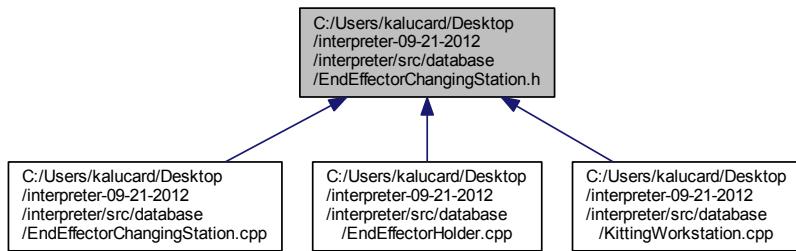
9.21 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffector-ChangingStation.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "SolidObject.h"
Include dependency graph for EndEffectorChangingStation.h:
```

Include dependency graph for EndEffectorChangingStation.h:



This graph shows which files directly or indirectly include this file:



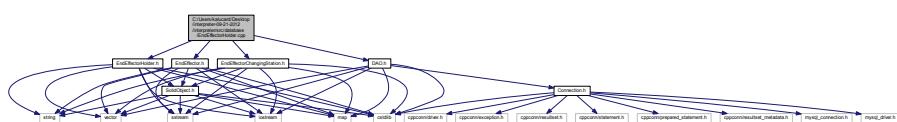
Classes

- class [EndEffectorChangingStation](#)

9.22 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffectorHolder.cpp File Reference

```
#include "EndEffectorHolder.h"
#include "EndEffector.h"
#include "EndEffectorChangingStation.h"
#include "DAO.h"
```

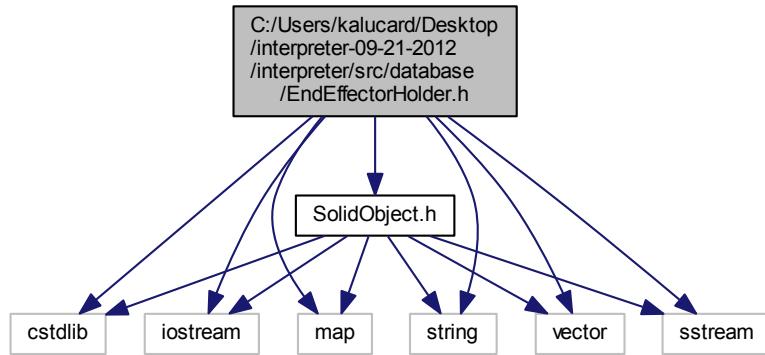
Include dependency graph for EndEffectorHolder.cpp:



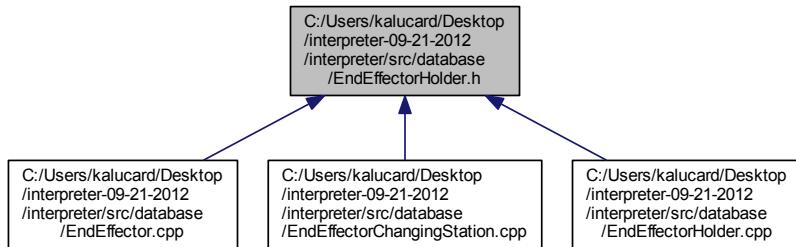
9.23 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffectorHolder.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "SolidObject.h"
```

Include dependency graph for EndEffectorHolder.h:



This graph shows which files directly or indirectly include this file:



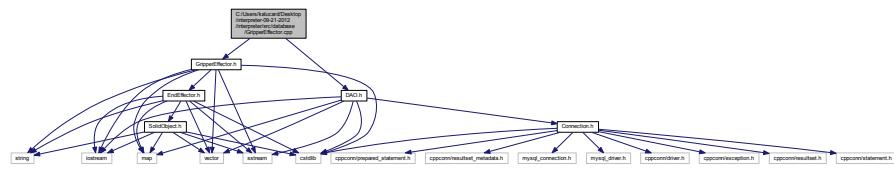
Classes

- class [EndEffectorHolder](#)

9.24 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/GripperEffector.cpp File Reference

```
#include "GripperEffector.h"
#include "DAO.h"
```

Include dependency graph for GripperEffector.cpp:

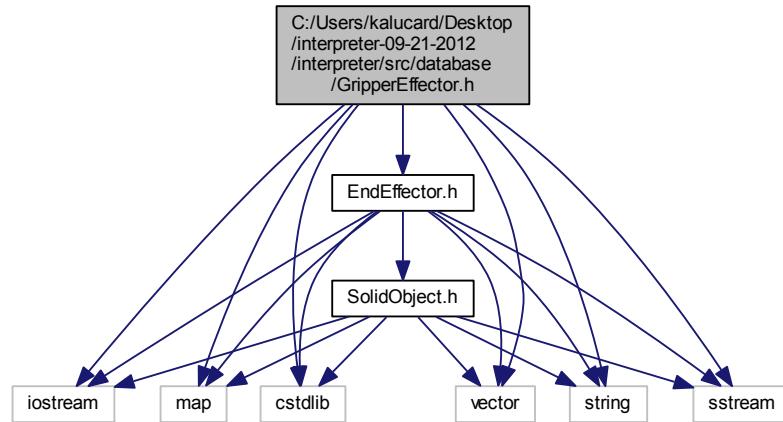


9.25 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/GripperEffector.h File Reference

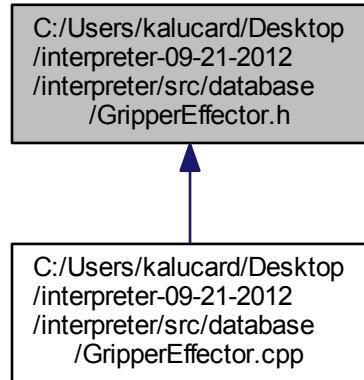
```

#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "EndEffector.h"
  
```

Include dependency graph for GripperEffector.h:



This graph shows which files directly or indirectly include this file:



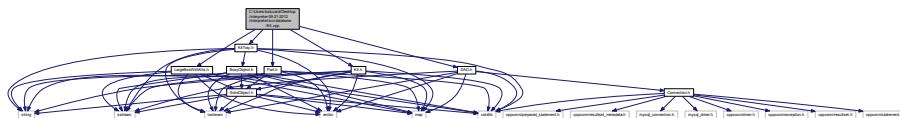
Classes

- class [GripperEffector](#)

9.26 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Kit.cpp File Reference

```
#include "Kit.h"
#include "LargeBoxWithKits.h"
#include "KitTray.h"
#include "DAO.h"
#include "Part.h"
```

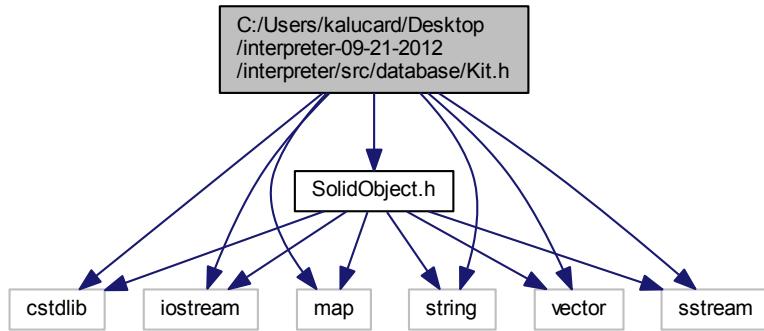
Include dependency graph for Kit.cpp:



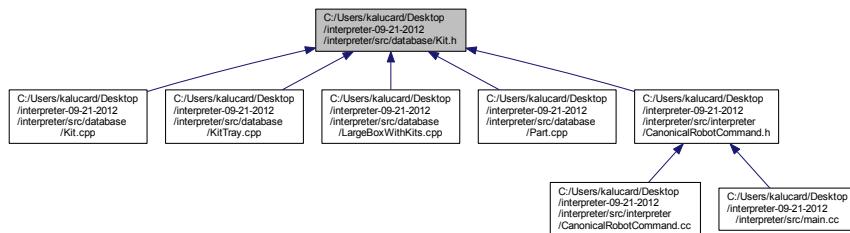
9.27 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Kit.h File Reference

```
#include <cstdlib>
```

```
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "SolidObject.h"
Include dependency graph for Kit.h:
```



This graph shows which files directly or indirectly include this file:



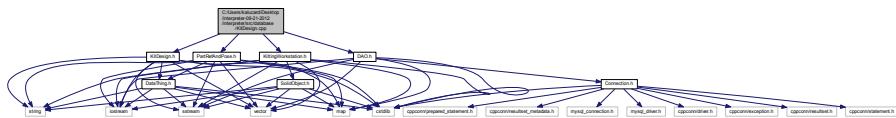
Classes

- class [Kit](#)

9.28 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KitDesign.cpp File Reference

```
#include "KitDesign.h"
#include "KittingWorkstation.h"
#include "DAO.h"
#include "PartRefAndPose.h"
```

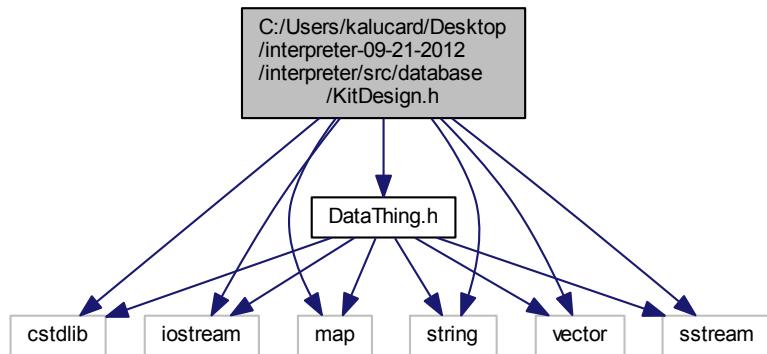
Include dependency graph for KitDesign.cpp:



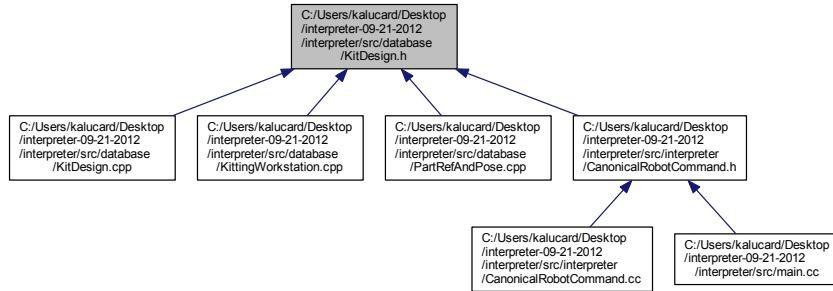
9.29 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KitDesign.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "DataThing.h"
Include dependency graph for KitDesign.h:
```

Include dependency graph for KitDesign.h:



This graph shows which files directly or indirectly include this file:



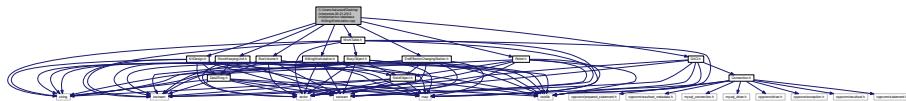
Classes

- class [KitDesign](#)

9.30 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KittingWorkstation.cpp File Reference

```
#include "KittingWorkstation.h"
#include "Robot.h"
#include "BoxVolume.h"
#include "KitDesign.h"
#include "StockKeepingUnit.h"
#include "EndEffectorChangingStation.h"
#include "DAO.h"
#include "WorkTable.h"
```

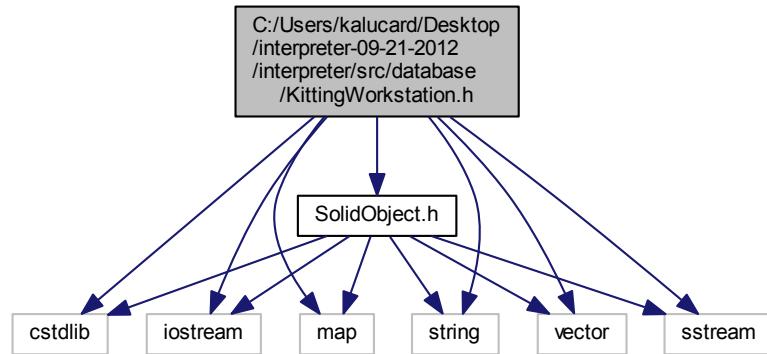
Include dependency graph for KittingWorkstation.cpp:



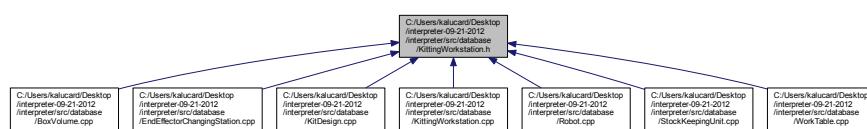
9.31 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KittingWorkstation.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "SolidObject.h"
```

Include dependency graph for KittingWorkstation.h:



This graph shows which files directly or indirectly include this file:



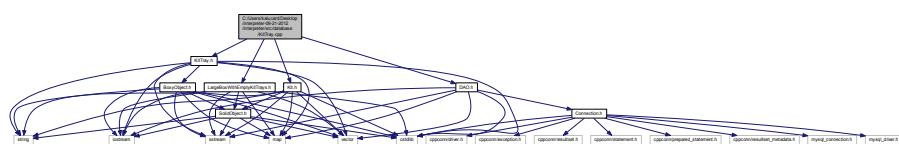
Classes

- class `KittingWorkstation`

9.32 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KitTray.cpp
File Reference

```
#include "KitTray.h"
#include "LargeBoxWithEmptyKitTrays.h"
#include "Kit.h"
#include "DAO.h"
```

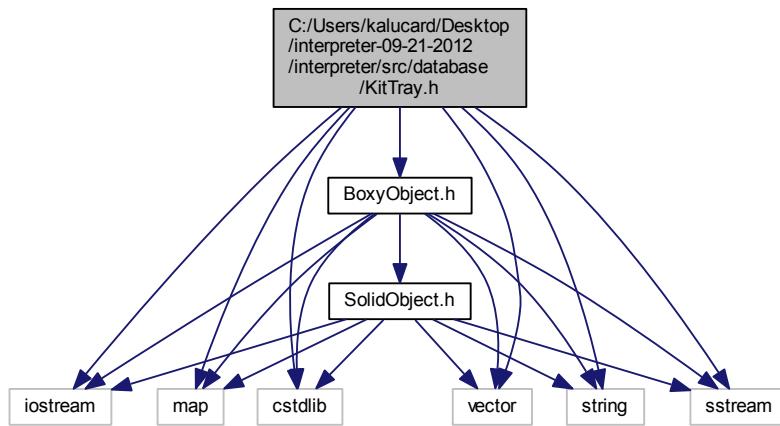
Include dependency graph for KitTray.cpp:



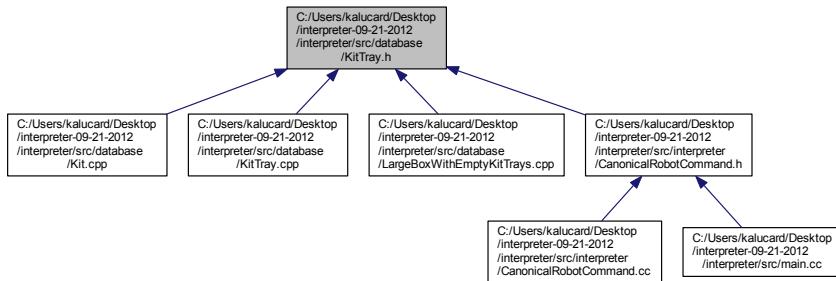
9.33 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KitTray.h

File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "BoxyObject.h"
Include dependency graph for KitTray.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [KitTray](#)

9.34

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/LargeBoxWithEmptyKitTrays.cpp

File Reference

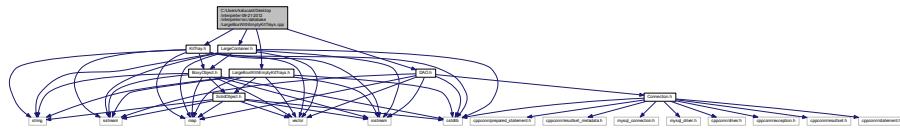
441

9.34 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/LargeBox-

WithEmptyKitTrays.cpp File Reference

```
#include "LargeBoxWithEmptyKitTrays.h"
#include "KitTray.h"
#include "LargeContainer.h"
#include "DAO.h"
```

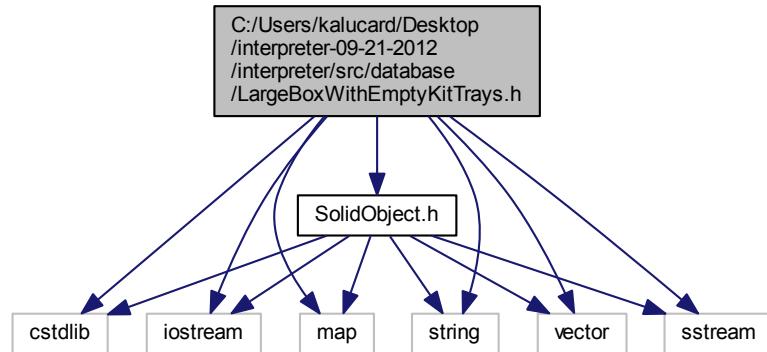
Include dependency graph for LargeBoxWithEmptyKitTrays.cpp:



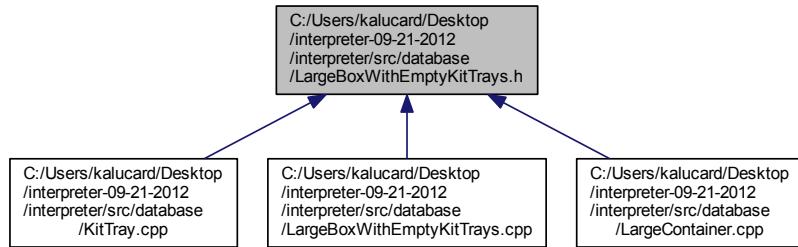
9.35 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/LargeBox- WithEmptyKitTrays.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "SolidObject.h"
```

Include dependency graph for LargeBoxWithEmptyKitTrays.h:



This graph shows which files directly or indirectly include this file:



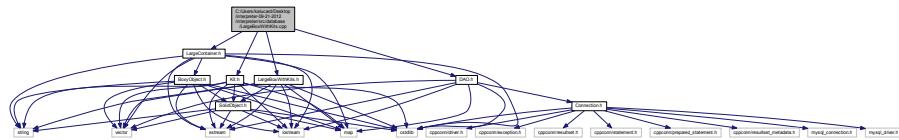
Classes

- class [LargeBoxWithEmptyKitTrays](#)

9.36 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/LargeBox-WithKits.cpp File Reference

```
#include "LargeBoxWithKits.h"
#include "Kit.h"
#include "LargeContainer.h"
#include "DAO.h"
```

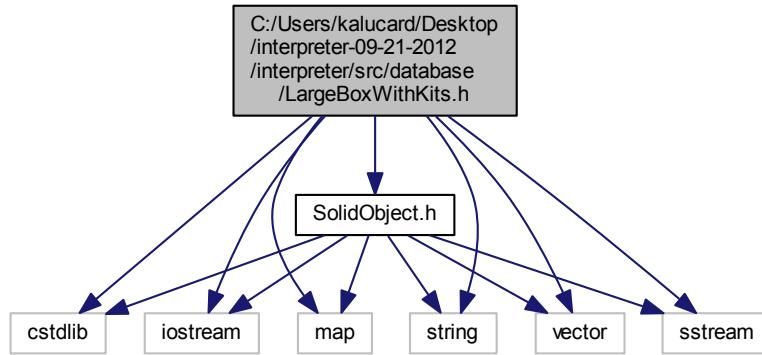
Include dependency graph for LargeBoxWithKits.cpp:



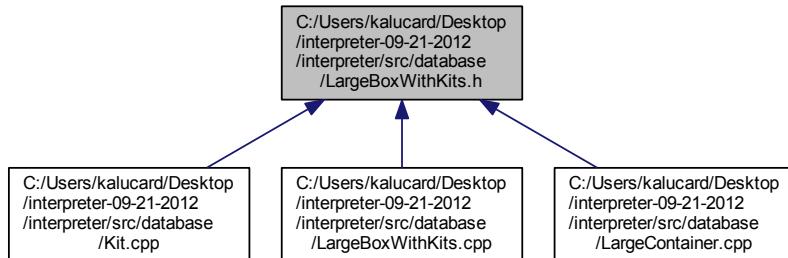
9.37 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/LargeBox-WithKits.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "SolidObject.h"
```

Include dependency graph for LargeBoxWithKits.h:



This graph shows which files directly or indirectly include this file:



Classes

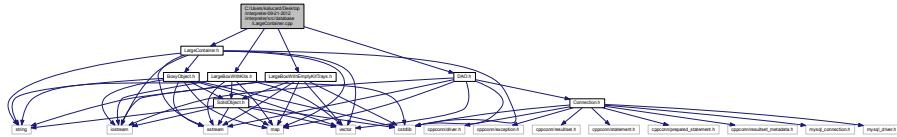
- class [LargeBoxWithKits](#)

9.38 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/LargeContainer.cpp File Reference

```

#include "LargeContainer.h"
#include "LargeBoxWithKits.h"
#include "LargeBoxWithEmptyKitTrays.h"
#include "DAO.h"
  
```

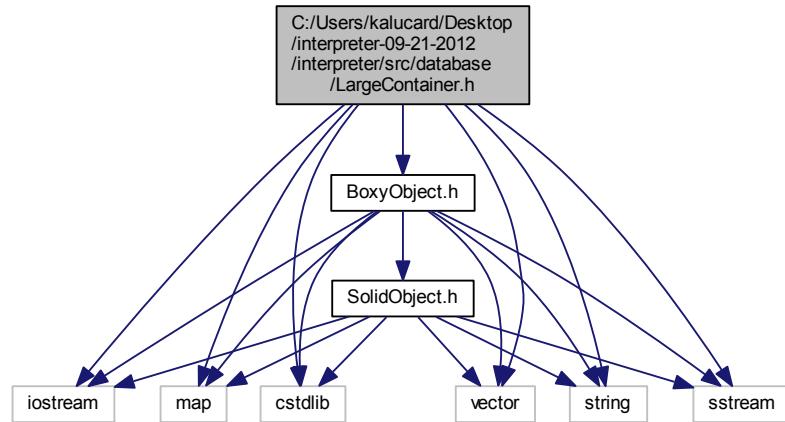
Include dependency graph for LargeContainer.cpp:



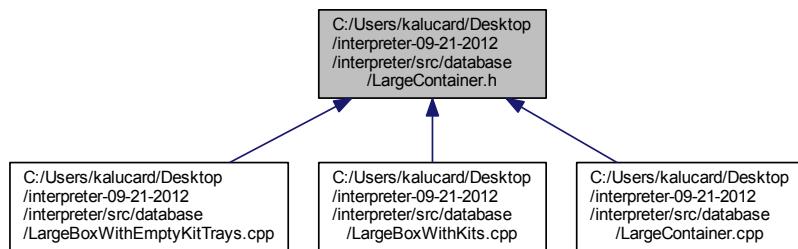
9.39 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/LargeContainer.h

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "BoxyObject.h"
Include dependency graph for LargeContainer.h:
```

Include dependency graph for LargeContainer.h:



This graph shows which files directly or indirectly include this file:



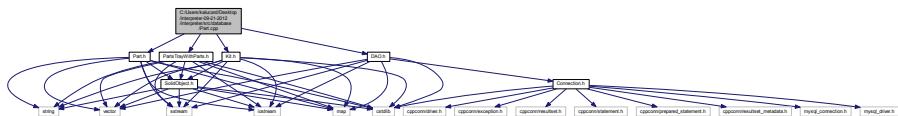
Classes

- class LargeContainer

9.40 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Part.cpp File Reference

```
#include "Part.h"
#include "PartsTrayWithParts.h"
#include "Kit.h"
#include "DAO.h"
Include dependency graph for Part.cpp:
```

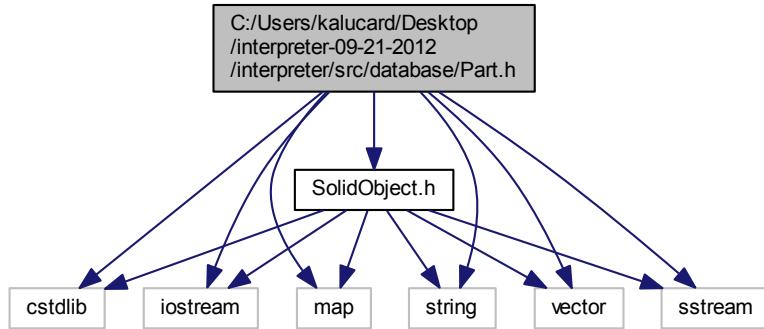
Include dependency graph for Part.cpp:



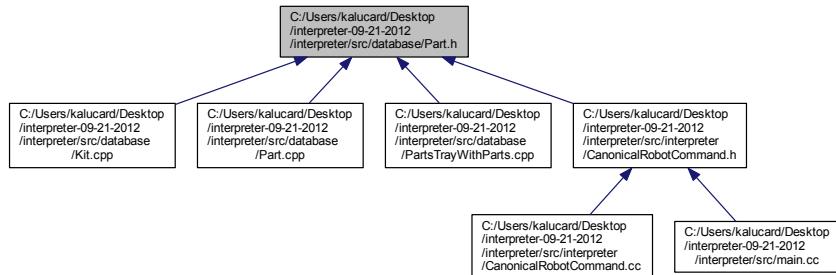
9.41 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Part.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "SolidObject.h"
```

Include dependency graph for Part.h:



This graph shows which files directly or indirectly include this file:



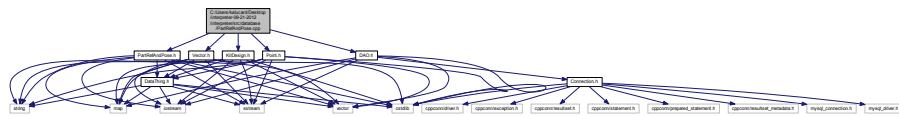
Classes

- class [Part](#)

9.42 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartRefAndPose.cpp File Reference

```
#include "PartRefAndPose.h"
#include "DAO.h"
#include "Vector.h"
#include "KitDesign.h"
#include "Point.h"
```

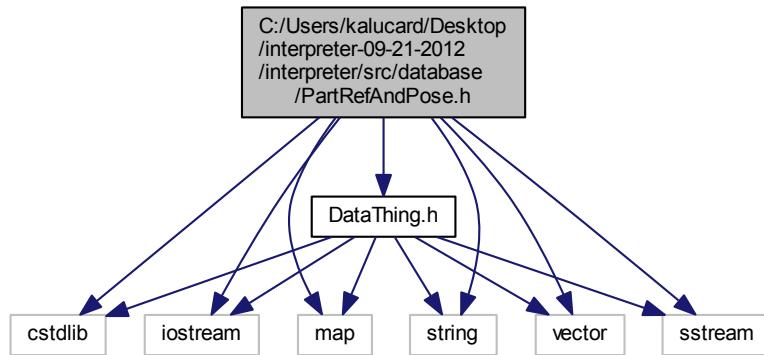
Include dependency graph for PartRefAndPose.cpp:



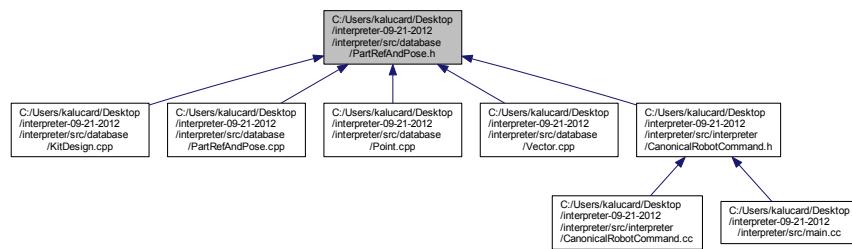
9.43 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartRefAndPose.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "DataThing.h"
Include dependency graph for PartRefAndPose.h:
```

Include dependency graph for PartRefAndPose.h:



This graph shows which files directly or indirectly include this file:



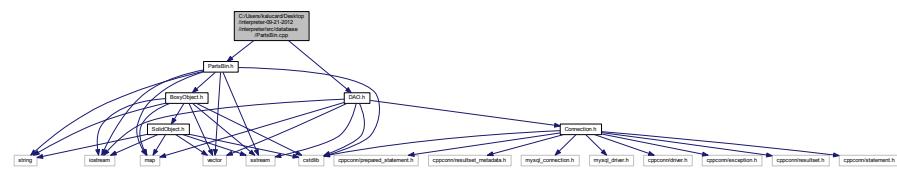
Classes

- class [PartRefAndPose](#)

9.44 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsBin.cpp File Reference

```
#include "PartsBin.h"
#include "DAO.h"
```

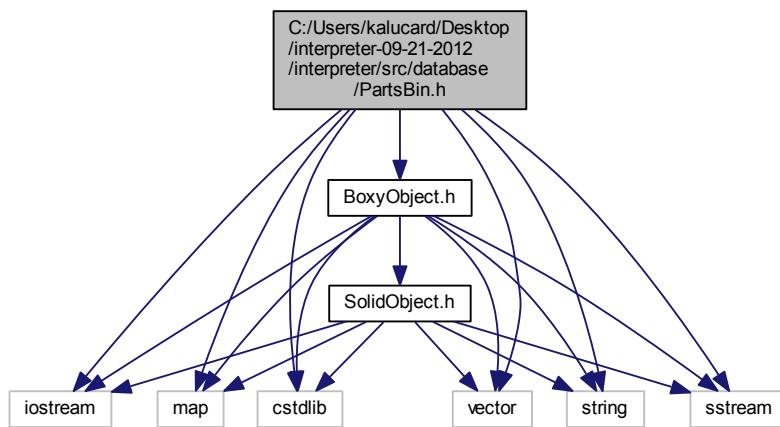
Include dependency graph for PartsBin.cpp:



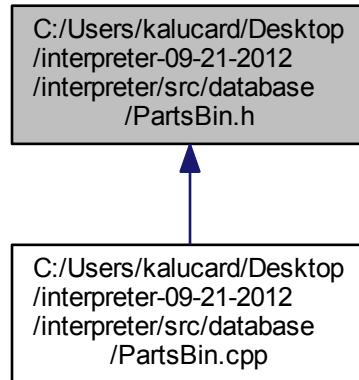
9.45 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsBin.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "BoxyObject.h"
```

Include dependency graph for PartsBin.h:



This graph shows which files directly or indirectly include this file:

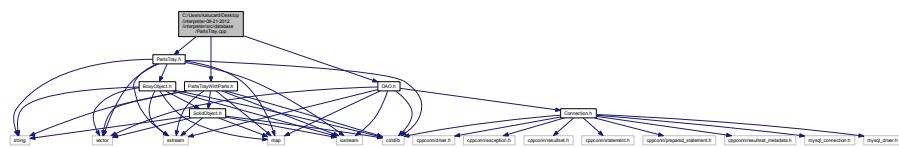


Classes

- class [PartsBin](#)

9.46 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsTray.cpp File Reference

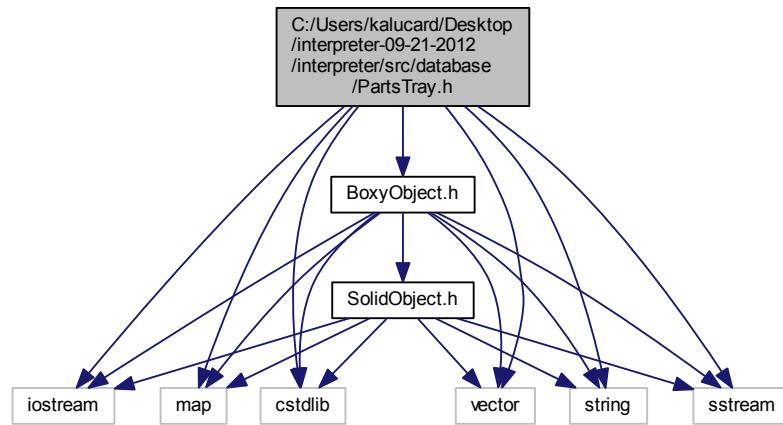
```
#include "PartsTray.h"
#include "PartsTrayWithParts.h"
#include "DAO.h"
Include dependency graph for PartsTray.cpp:
```



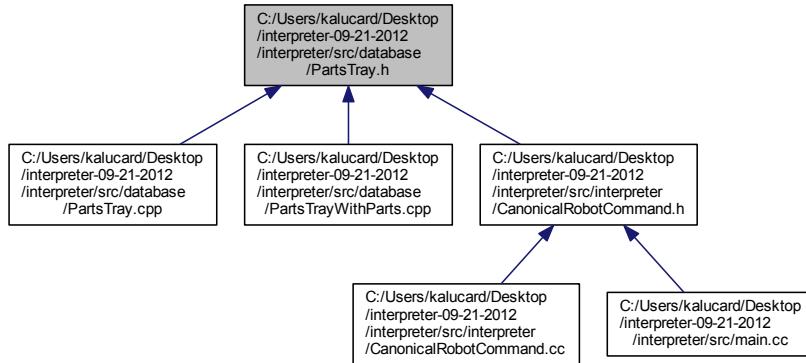
9.47 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsTray.h File Reference

```
#include <cstdlib>
```

```
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "BoxyObject.h"
Include dependency graph for PartsTray.h:
```



This graph shows which files directly or indirectly include this file:

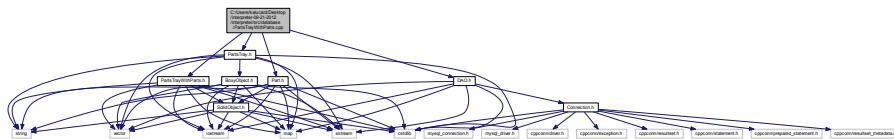


Classes

- class [PartsTray](#)

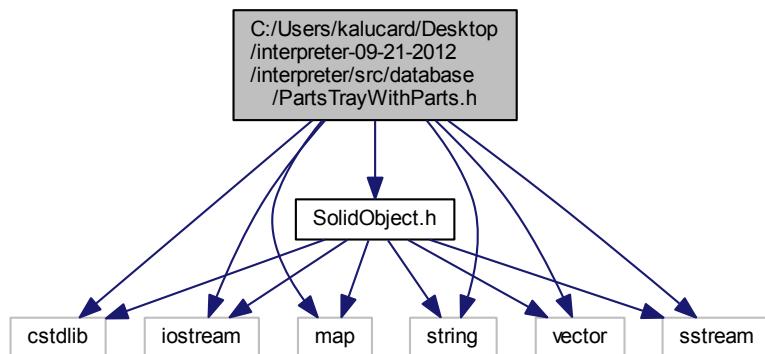
9.48 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsTray- WithParts.cpp File Reference

```
#include "PartsTrayWithParts.h"
#include "DAO.h"
#include "Part.h"
#include "PartsTray.h"
Include dependency graph for PartsTrayWithParts.cpp:
```

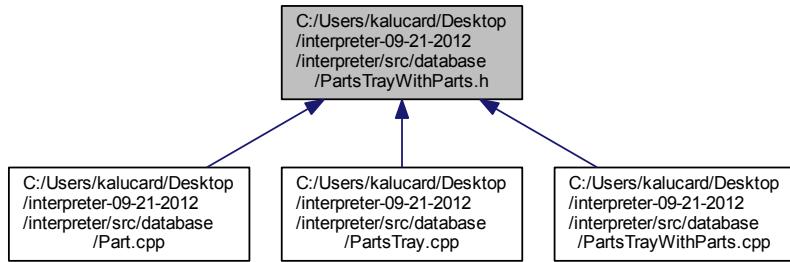


9.49 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsTray- WithParts.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "SolidObject.h"
Include dependency graph for PartsTrayWithParts.h:
```



This graph shows which files directly or indirectly include this file:

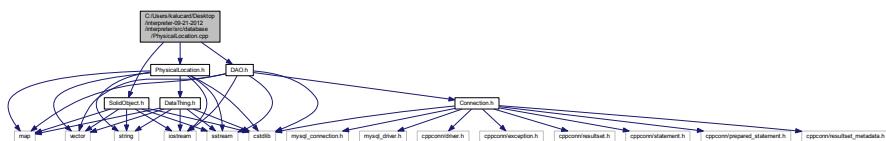


Classes

- class [PartsTrayWithParts](#)

9.50 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PhysicalLocation.cpp File Reference

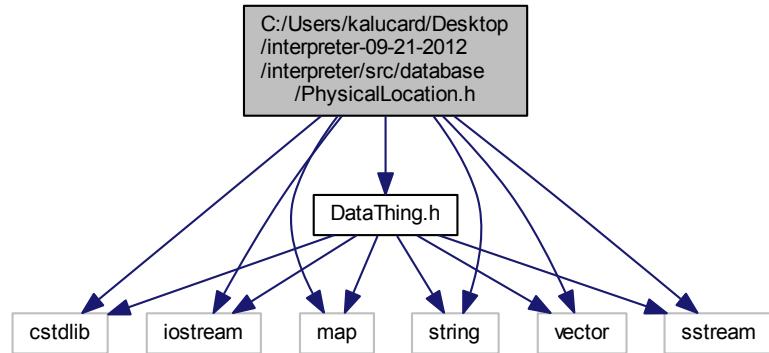
```
#include "PhysicalLocation.h"
#include "SolidObject.h"
#include "DAO.h"
Include dependency graph for PhysicalLocation.cpp:
```



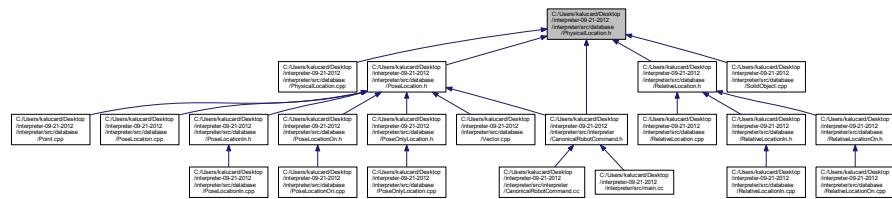
9.51 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PhysicalLocation.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "DataThing.h"
```

Include dependency graph for PhysicalLocation.h:



This graph shows which files directly or indirectly include this file:



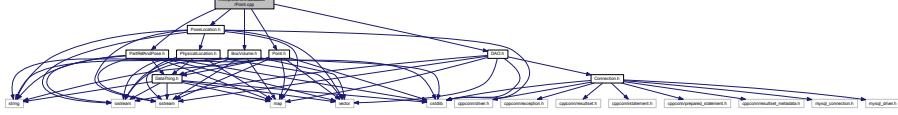
Classes

- class PhysicalLocation

9.52 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Point.cpp

File Reference

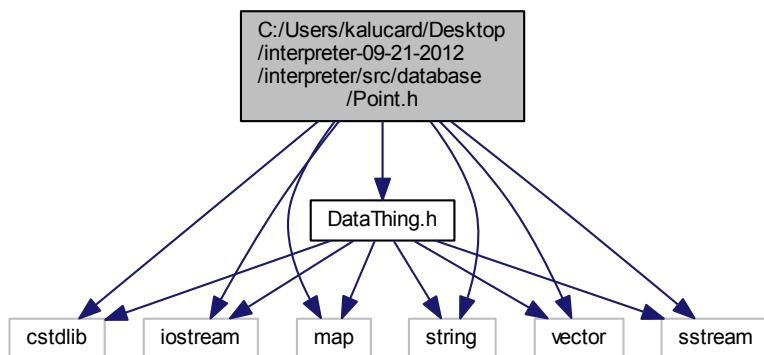
```
#include "Point.h"
#include "BoxVolume.h"
#include "DAO.h"
#include "PartRefAndPose.h"
#include "PoseLocation.h"
Include dependency graph for Point.cpp:
```



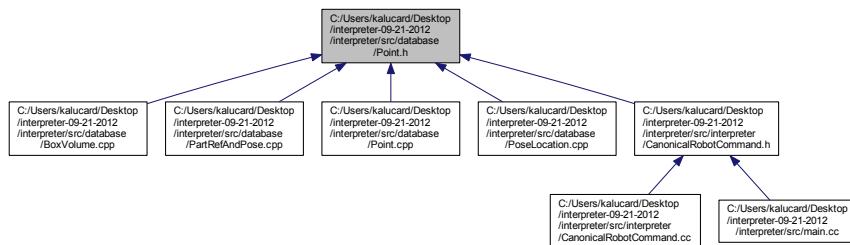
9.53 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Point.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "DataThing.h"
```

Include dependency graph for Point.h:



This graph shows which files directly or indirectly include this file:

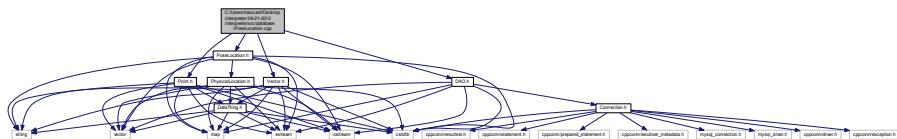


Classes

- class [Point](#)

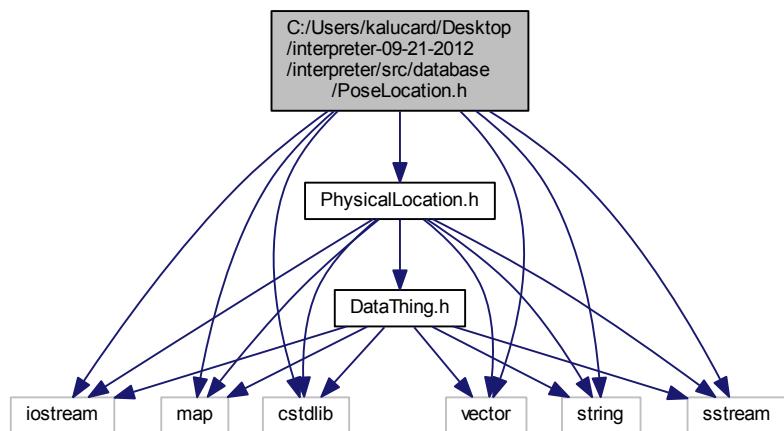
9.54 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseLocation.cpp
File Reference

```
#include "PoseLocation.h"
#include "DAO.h"
#include "Vector.h"
#include "Point.h"
Include dependency graph for PoseLocation.cpp:
```

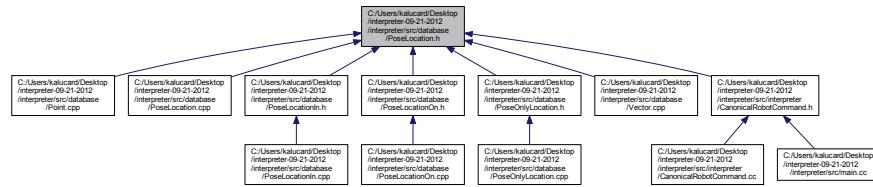


9.55 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseLocation.h

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "PhysicalLocation.h"
Include dependency graph for PoseLocation.h:
```



This graph shows which files directly or indirectly include this file:

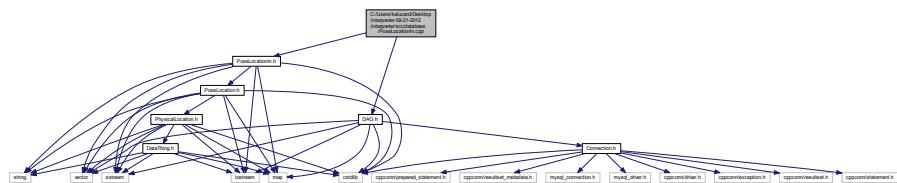


Classes

- class [PoseLocation](#)

9.56 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseLocationIn.cpp File Reference

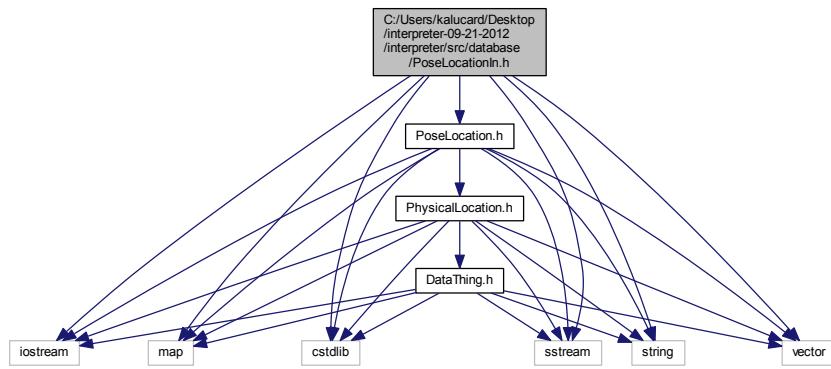
```
#include "PoseLocationIn.h"
#include "DAO.h"
Include dependency graph for PoseLocationIn.cpp:
```



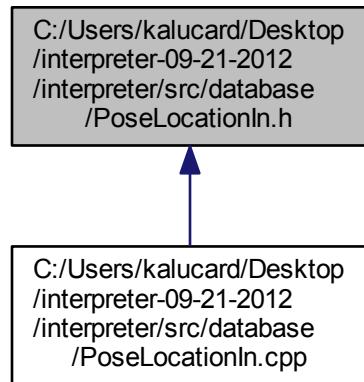
9.57 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseLocationIn.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "PoseLocation.h"
```

Include dependency graph for PoseLocationIn.h:



This graph shows which files directly or indirectly include this file:



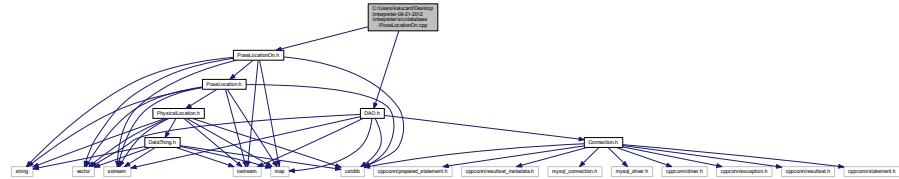
Classes

- class [PoseLocationIn](#)

9.58 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseLocationOn.cpp File Reference

```
#include "PoseLocationOn.h"
#include "DAO.h"
```

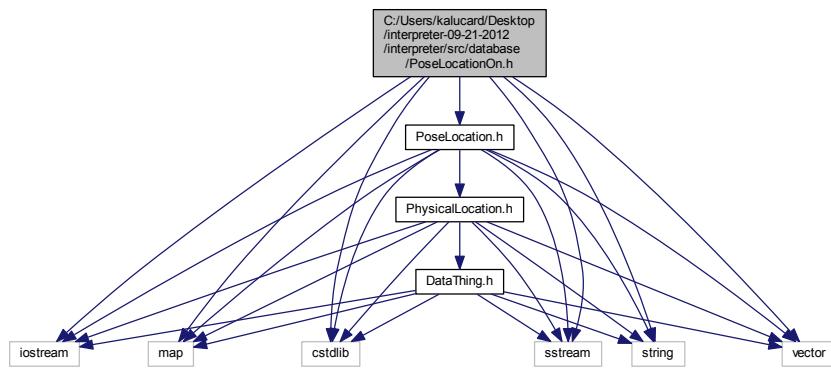
Include dependency graph for PoseLocationOn.cpp:



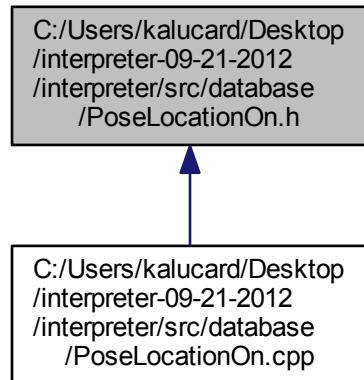
9.59 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseLocationOn.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "PoseLocation.h"
```

Include dependency graph for PoseLocationOn.h:



This graph shows which files directly or indirectly include this file:

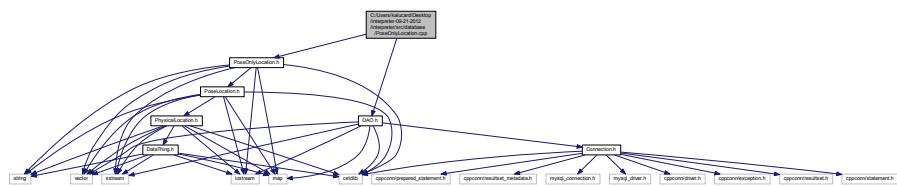


Classes

- class PoseLocationOn

9.60 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseOnly-Location.cpp File Reference

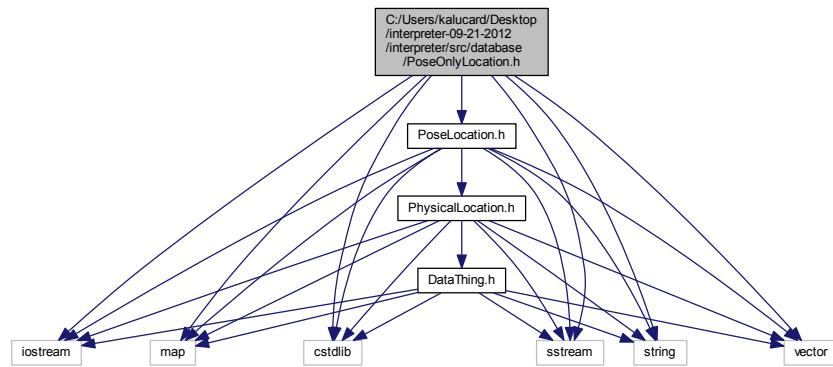
```
#include "PoseOnlyLocation.h"
#include "DAO.h"
Include dependency graph for PoseOnlyLocation.cpp:
```



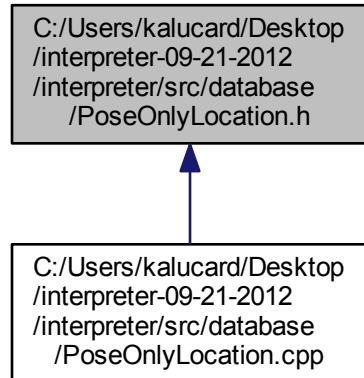
9.61 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseOnly-Location.h File Reference

```
#include <cstdlib>
```

```
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "PoseLocation.h"
Include dependency graph for PoseOnlyLocation.h:
```



This graph shows which files directly or indirectly include this file:

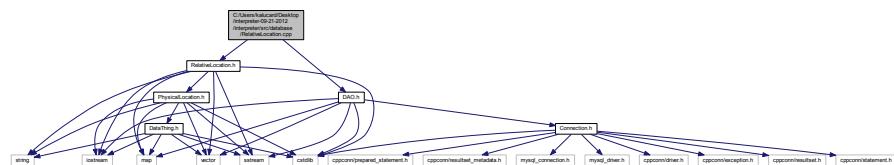


Classes

- class [PoseOnlyLocation](#)

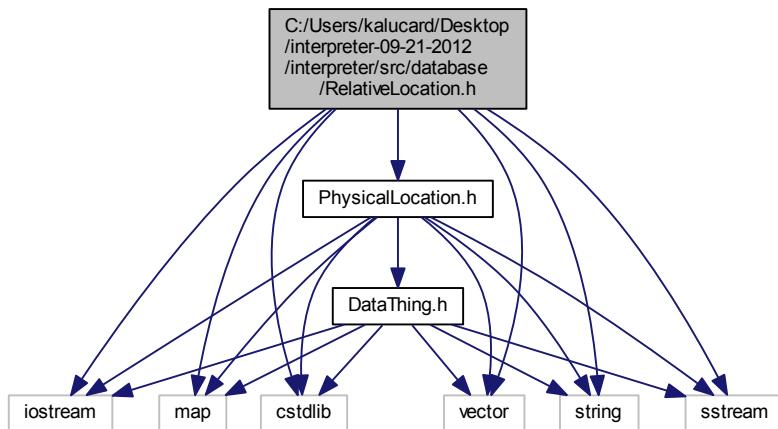
9.62 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/RelativeLocation.cpp File Reference

```
#include "RelativeLocation.h"
#include "DAO.h"
Include dependency graph for RelativeLocation.cpp:
```

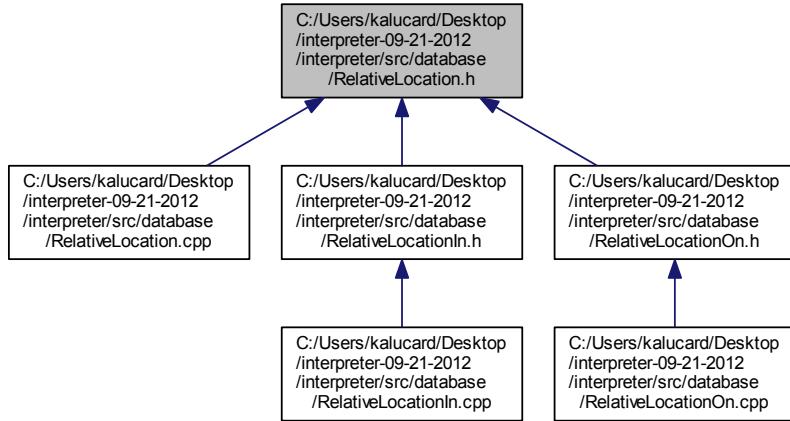


9.63 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/RelativeLocation.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "PhysicalLocation.h"
Include dependency graph for RelativeLocation.h:
```



This graph shows which files directly or indirectly include this file:



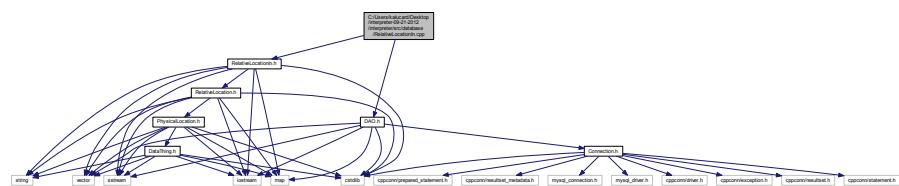
Classes

- class [RelativeLocation](#)

9.64 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/RelativeLocationIn.cpp File Reference

```
#include "RelativeLocationIn.h"
#include "DAO.h"
```

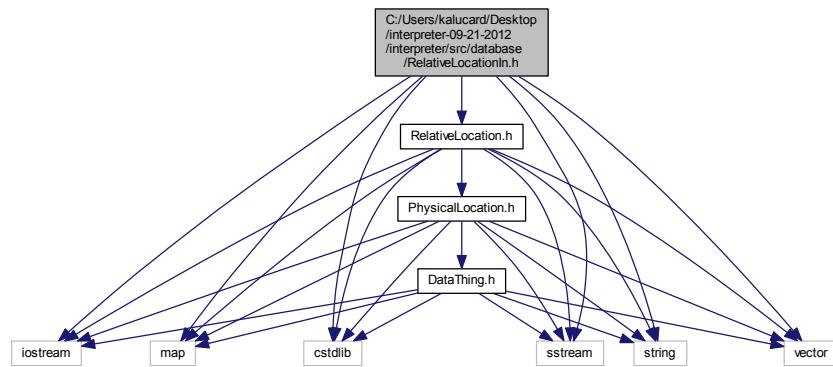
Include dependency graph for RelativeLocationIn.cpp:



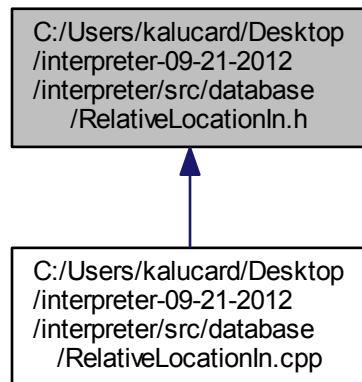
9.65 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/RelativeLocationIn.h File Reference

```
#include <cstdlib>
```

```
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "RelativeLocation.h"
Include dependency graph for RelativeLocationIn.h:
```



This graph shows which files directly or indirectly include this file:

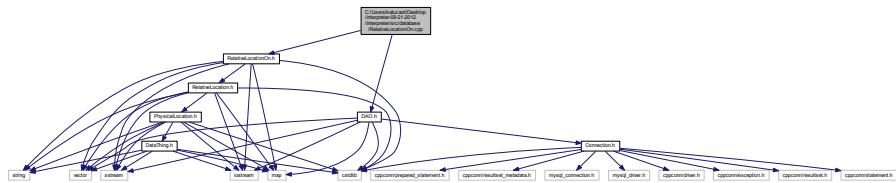


Classes

- class [RelativeLocationIn](#)

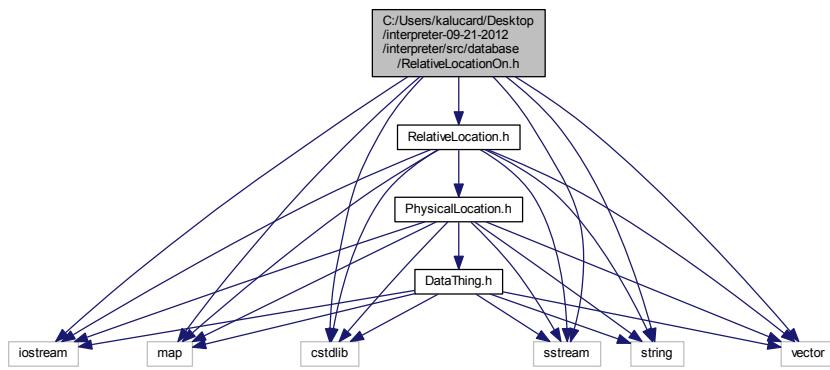
9.66 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/RelativeLocationOn.cpp File Reference

```
#include "RelativeLocationOn.h"
#include "DAO.h"
Include dependency graph for RelativeLocationOn.cpp:
```

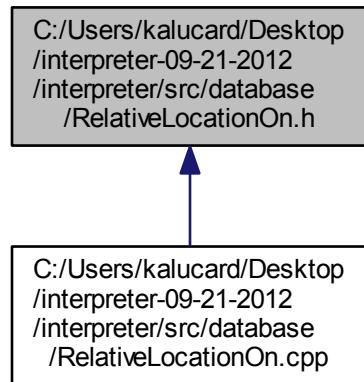


9.67 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/RelativeLocationOn.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "RelativeLocation.h"
Include dependency graph for RelativeLocationOn.h:
```



This graph shows which files directly or indirectly include this file:



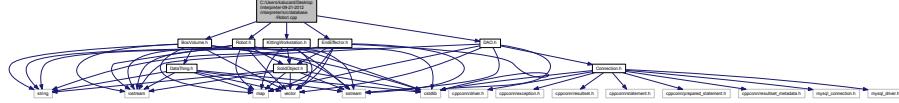
Classes

- class `RelativeLocationOn`

9.68 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Robot.cpp

```
#include "Robot.h"
#include "BoxVolume.h"
#include "EndEffector.h"
#include "KittingWorkstation.h"
#include "DAO.h"
Include dependency graph for Robot.cpp:
```

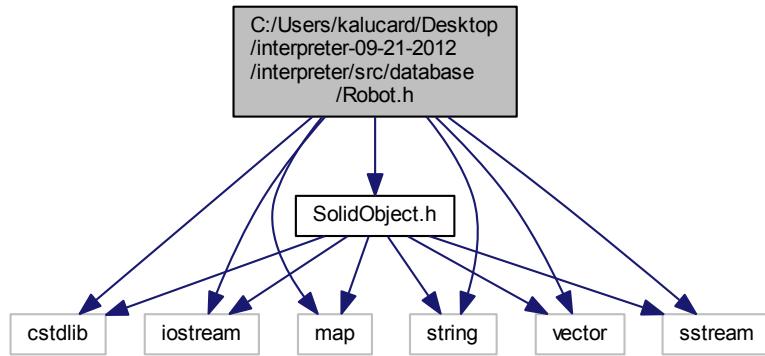
Include dependency graph for Robot.cpp:



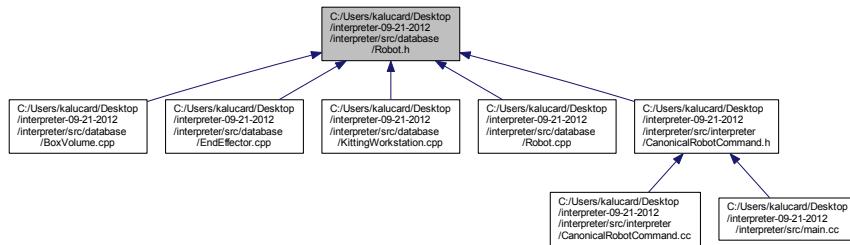
9.69 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Robot.h File Reference

```
#include <cstdlib>
```

```
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "SolidObject.h"
Include dependency graph for Robot.h:
```



This graph shows which files directly or indirectly include this file:



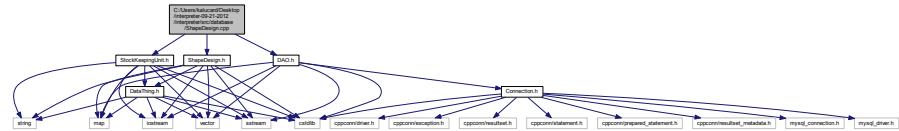
Classes

- class [Robot](#)

9.70 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ShapeDesign.cpp File Reference

```
#include "ShapeDesign.h"
#include "StockKeepingUnit.h"
#include "DAO.h"
```

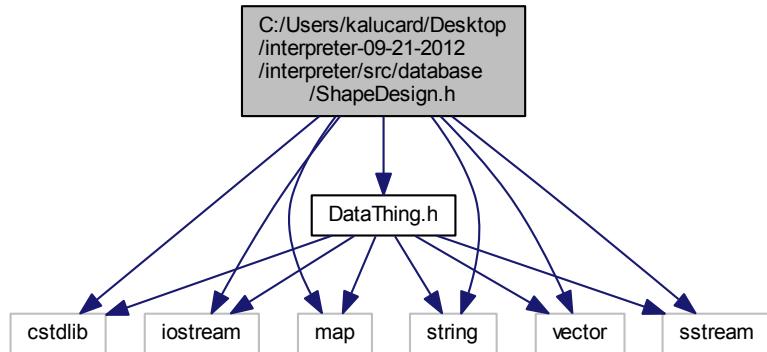
Include dependency graph for ShapeDesign.cpp:



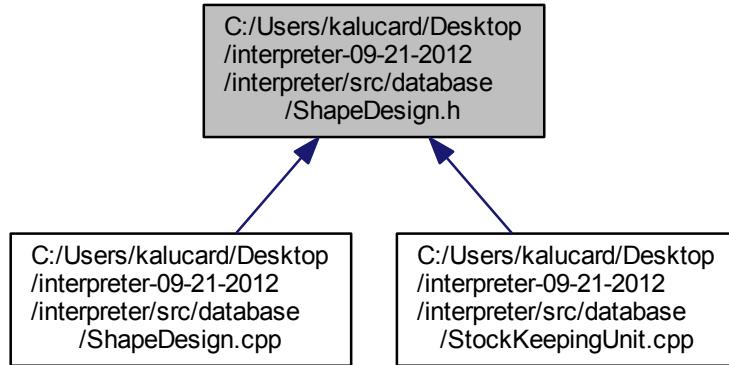
9.71 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/ShapeDesign.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "DataThing.h"
```

Include dependency graph for ShapeDesign.h:



This graph shows which files directly or indirectly include this file:

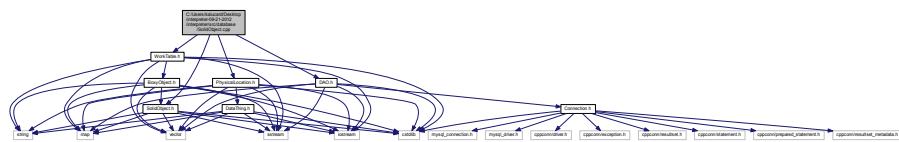


Classes

- class [ShapeDesign](#)

9.72 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/SolidObject.cpp File Reference

```
#include "SolidObject.h"
#include "PhysicalLocation.h"
#include "DAO.h"
#include "WorkTable.h"
Include dependency graph for SolidObject.cpp:
```

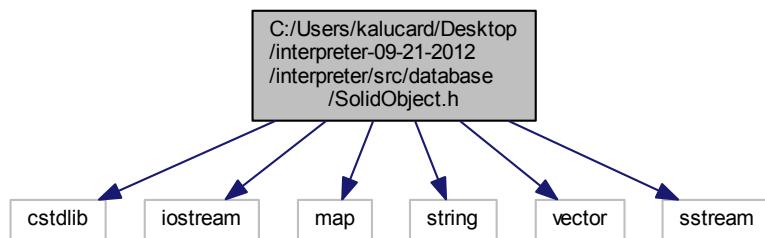


9.73 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/SolidObject.h File Reference

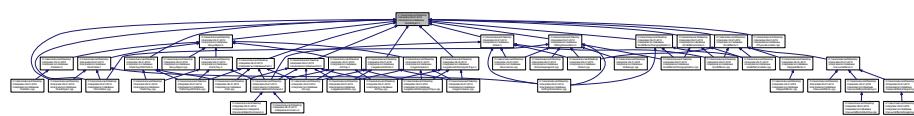
```
#include <cstdlib>
```

```
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
```

Include dependency graph for SolidObject.h:



This graph shows which files directly or indirectly include this file:



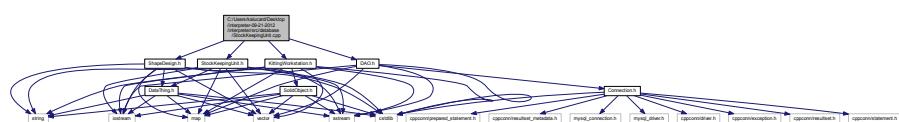
Classes

- class [SolidObject](#)

9.74 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/StockKeepingUnit.cpp File Reference

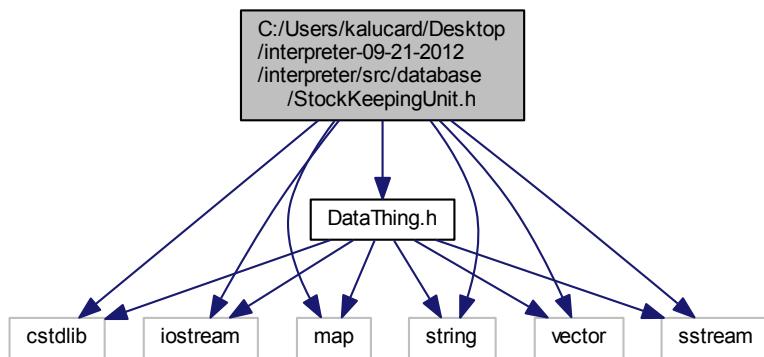
```
#include "StockKeepingUnit.h"
#include "KittingWorkstation.h"
#include "ShapeDesign.h"
#include "DAO.h"
```

Include dependency graph for StockKeepingUnit.cpp:

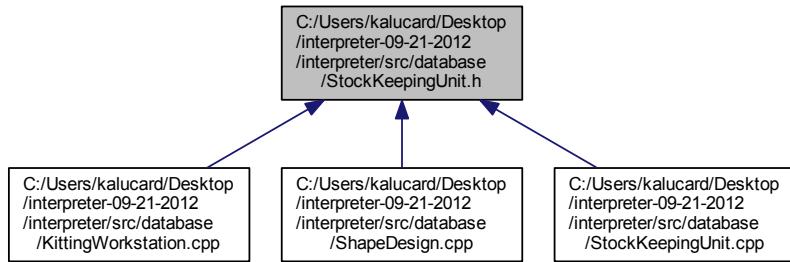


9.75 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/StockKeepingUnit.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "DataThing.h"
Include dependency graph for StockKeepingUnit.h:
```



This graph shows which files directly or indirectly include this file:



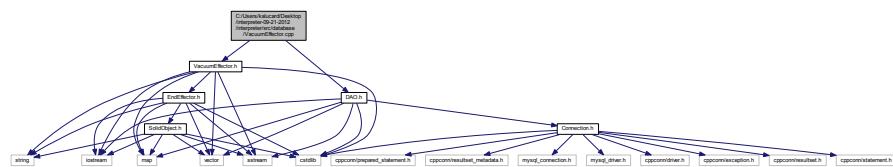
Classes

- class [StockKeepingUnit](#)

9.76 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/VacuumEffector.cpp File Reference

```
#include "VacuumEffector.h"
#include "DAO.h"
```

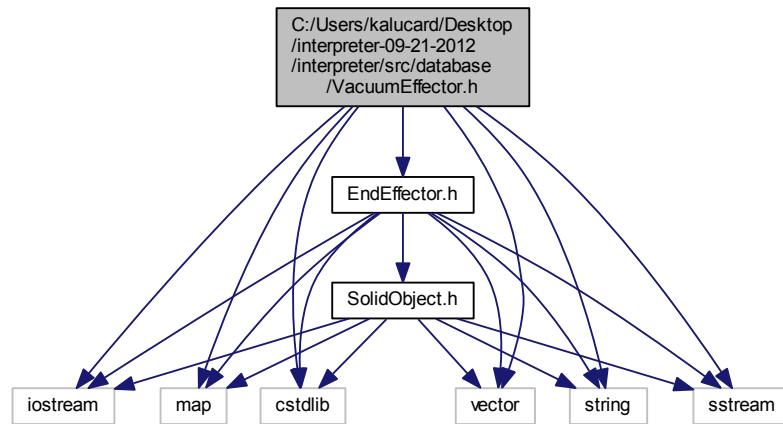
Include dependency graph for VacuumEffector.cpp:



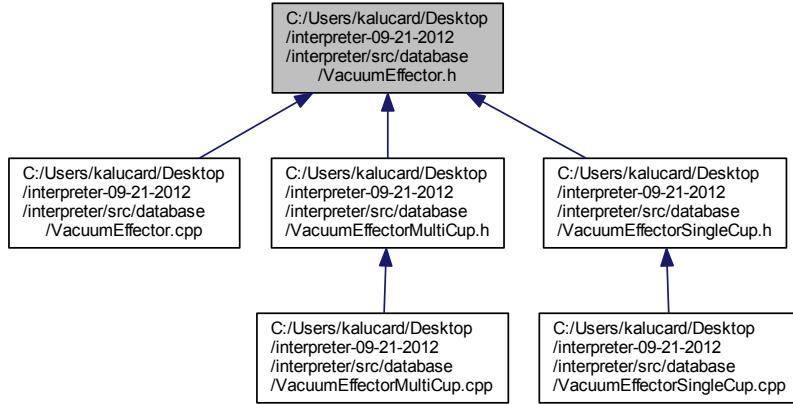
9.77 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/VacuumEffector.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "EndEffect.h"
```

Include dependency graph for VacuumEffector.h:



This graph shows which files directly or indirectly include this file:



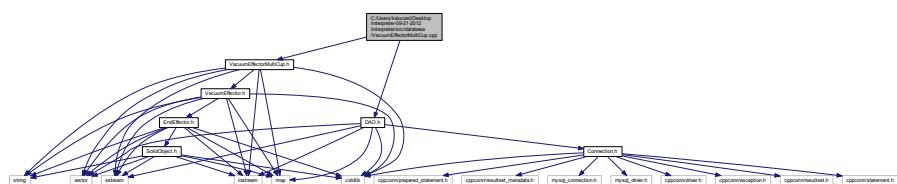
Classes

- class VacuumEffector

9.78 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Vacuum-EffectMultiCup.cpp File Reference

```
#include "VacuumEffectMultiCup.h"
#include "DAO.h"
```

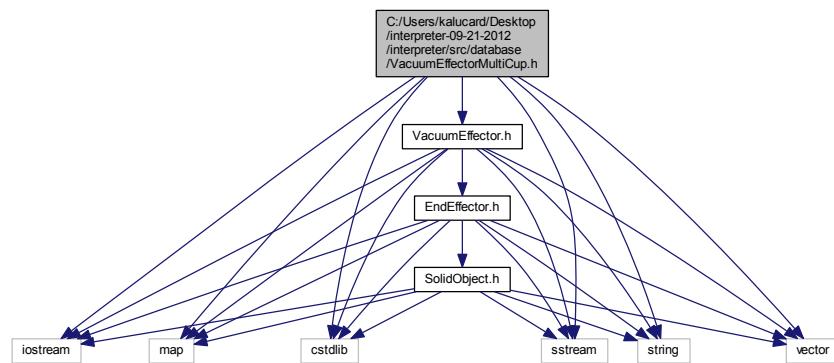
Include dependency graph for VacuumEffectorMultiCup.cpp:



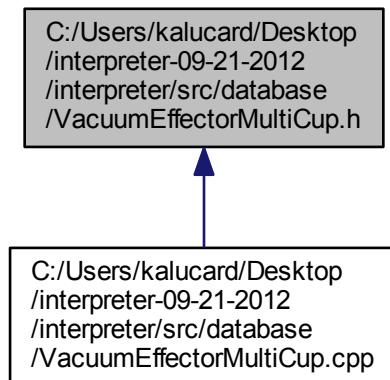
9.79 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Vacuum-
EffectoMultiCup.h File Reference

```
#include <cstdlib>
```

```
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "VacuumEffector.h"
Include dependency graph for VacuumEffectorMultiCup.h:
```



This graph shows which files directly or indirectly include this file:



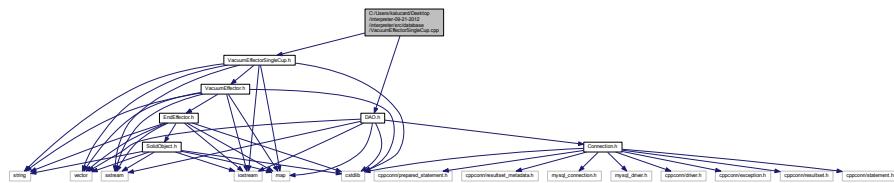
Classes

- class [VacuumEffectorMultiCup](#)

9.80 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/VacuumEffectorSingleCup.cpp File Reference

```
#include "VacuumEffectorSingleCup.h"
#include "DAO.h"

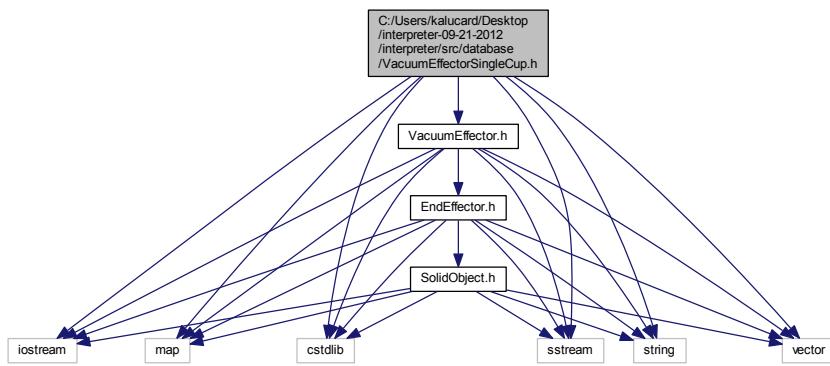
Include dependency graph for VacuumEffectorSingleCup.cpp:
```



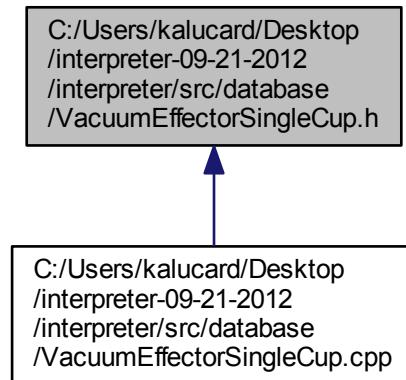
9.81 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/VacuumEffectorSingleCup.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "VacuumEffector.h"

Include dependency graph for VacuumEffectorSingleCup.h:
```



This graph shows which files directly or indirectly include this file:

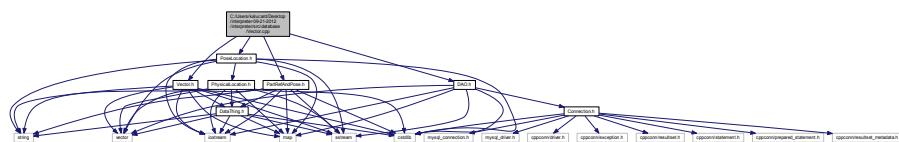


Classes

- class VacuumEffectoSsingleCup

9.82 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Vector.cpp

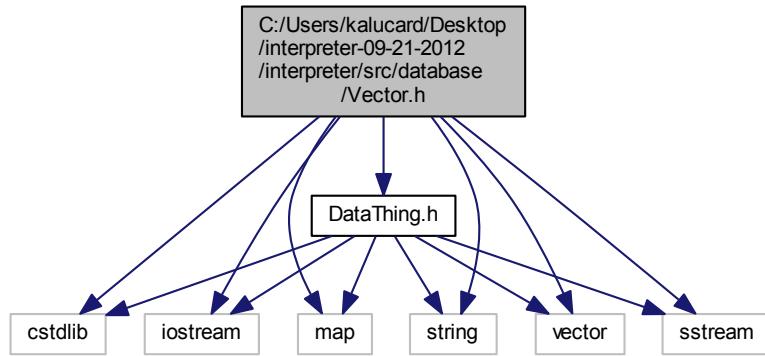
```
#include "Vector.h"
#include "DAO.h"
#include "PartRefAndPose.h"
#include "PoseLocation.h"
Include dependency graph for Vector.cpp:
```



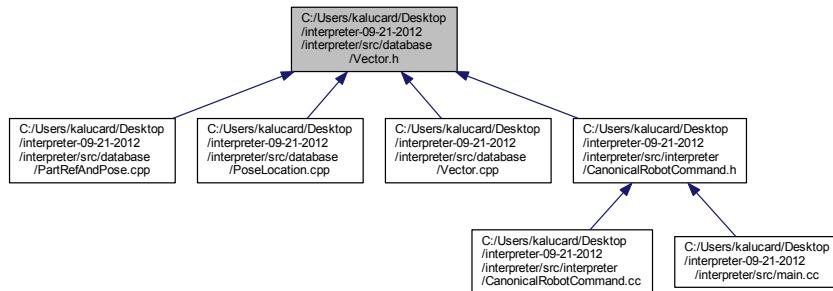
9.83 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Vector.h File Reference

```
#include <cstdlib>
```

```
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "DataThing.h"
Include dependency graph for Vector.h:
```



This graph shows which files directly or indirectly include this file:



Classes

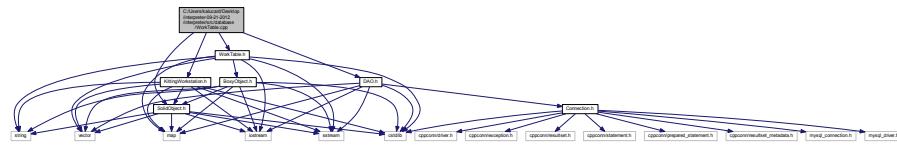
- class [Vector](#)

9.84 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/WorkTable.cpp

File Reference

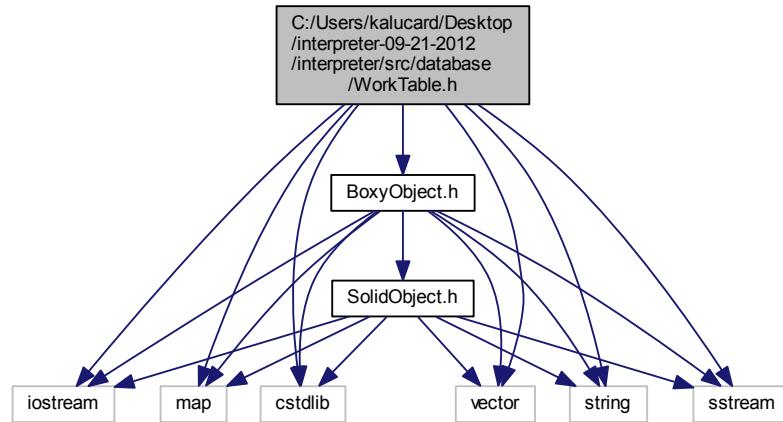
```
#include "WorkTable.h"
```

```
#include "KittingWorkstation.h"
#include "SolidObject.h"
#include "DAO.h"
Include dependency graph for WorkTable.cpp:
```

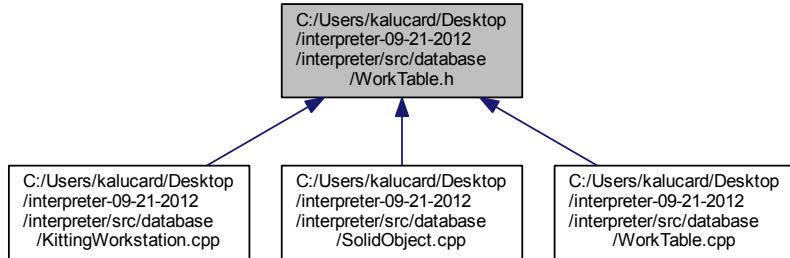


9.85 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/WorkTable.h File Reference

```
#include <cstdlib>
#include <iostream>
#include <map>
#include <string>
#include <vector>
#include <sstream>
#include "BoxyObject.h"
Include dependency graph for WorkTable.h:
```



This graph shows which files directly or indirectly include this file:

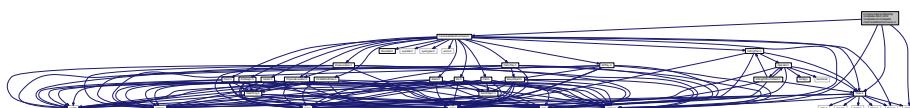


Classes

- class [WorkTable](#)

9.86 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/Canonical-RobotCommand.cc File Reference

```
#include "CanonicalRobotCommand.h"
#include "Tools.h"
#include <iostream>
#include <algorithm>
Include dependency graph for CanonicalRobotCommand.cc:
```

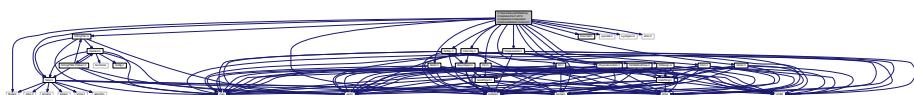


9.87 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/Canonical-RobotCommand.h File Reference

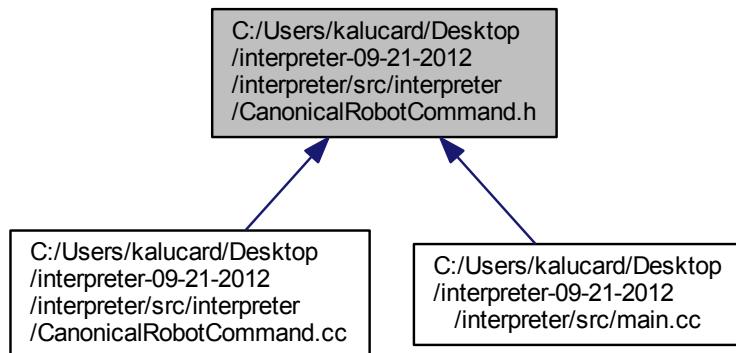
Test.

```
#include "Tools.h"
#include "KittingPlan.h"
#include "Kit.h"
#include "KitDesign.h"
#include "Part.h"
#include "PartRefAndPose.h"
#include "PartsTray.h"
#include "Point.h"
#include "PoseLocation.h"
#include "PhysicalLocation.h"
#include "Robot.h"
#include "Vector.h"
#include "KitTray.h"
#include "Structdef.h"
#include <iostream>
#include <fstream>
#include <sys/stat.h>
#include <sys/types.h>
#include <errno.h>
```

Include dependency graph for CanonicalRobotCommand.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [CanonicalRobotCommand](#)

*This class provides functions to build **Canonical Robot Commands** from PDDL actions in the Plan.*

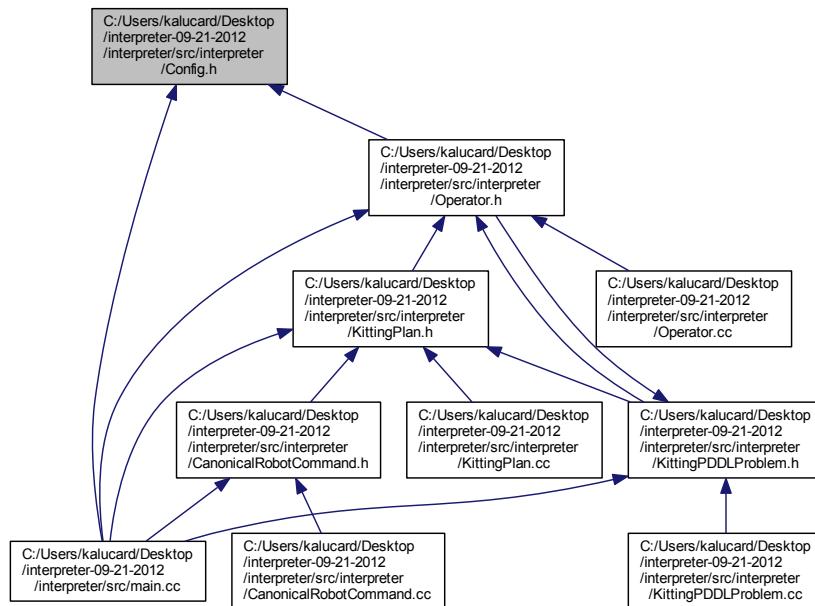
9.87.1 Detailed Description

Test.

Definition in file [CanonicalRobotCommand.h](#).

9.88 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/Config.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define PDDL_DOMAIN "domain_a2b2c1.pddl"
 - #define PDDL_FOLDER "etc/PDDL/"
 - #define PDDL_PROBLEM "problem_a2b2c1.pddl"
- Name and Path of the PDDL Problem File.*
- #define PLAN_FILE "plan_a2b2c1.txt"
 - #define PLAN_FOLDER "etc/Plan/"
 - #define ROBOT_COMMANDS_FILE "canon_commands_a2b2c1.txt"
 - #define ROBOT_COMMANDS_FOLDER "etc/CanonicalRobotCommands/"

9.88.1 Macro Definition Documentation

9.88.1.1 #define PDDL_DOMAIN "domain_a2b2c1.pddl"

Definition at line 37 of file Config.h.

9.88.1.2 #define PDDL_FOLDER "etc/PDDL/"

Definition at line 36 of file Config.h.

9.88.1.3 #define PDDL_PROBLEM "problem_a2b2c1.pddl"

Name and Path of the PDDL Problem File.

Definition at line 38 of file Config.h.

9.88.1.4 #define PLAN_FILE "plan_a2b2c1.txt"

Definition at line 30 of file Config.h.

9.88.1.5 #define PLAN_FOLDER "etc/Plan/"

Definition at line 29 of file Config.h.

9.88.1.6 #define ROBOT_COMMANDS_FILE "canon_commands_a2b2c1.txt"

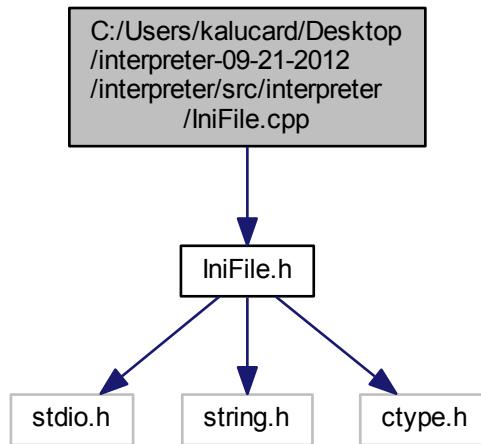
Definition at line 45 of file Config.h.

9.88.1.7 #define ROBOT_COMMANDS_FOLDER "etc/CanonicalRobotCommands/"

Definition at line 44 of file Config.h.

9.89 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/IniFile.cpp File Reference

```
#include "IniFile.h"  
Include dependency graph for IniFile.cpp:
```



Functions

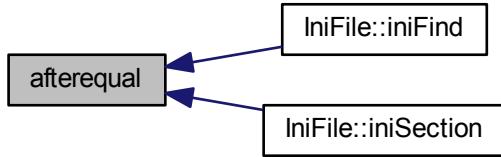
- static const char * [afterequal](#) (const char *string)
- static char * [skipwhite](#) (char *string)

9.89.1 Function Documentation

9.89.1.1 static const char* afterequal (const char * string) [static]

Definition at line 21 of file IniFile.cpp.

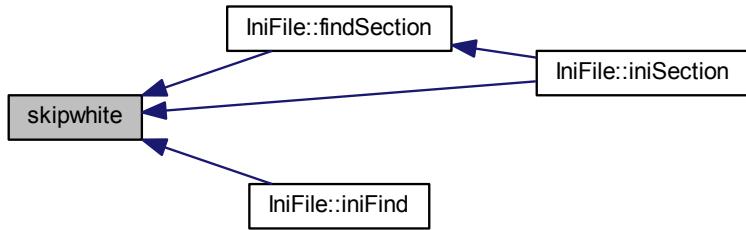
Here is the caller graph for this function:



9.89.1.2 static char* skipwhite (char * string) [static]

Definition at line 61 of file IniFile.cpp.

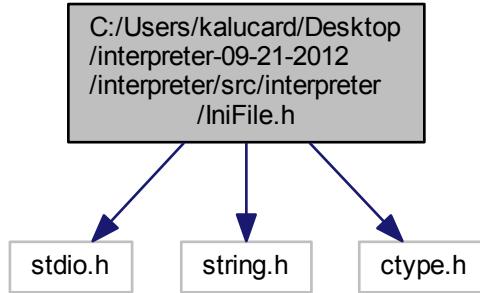
Here is the caller graph for this function:



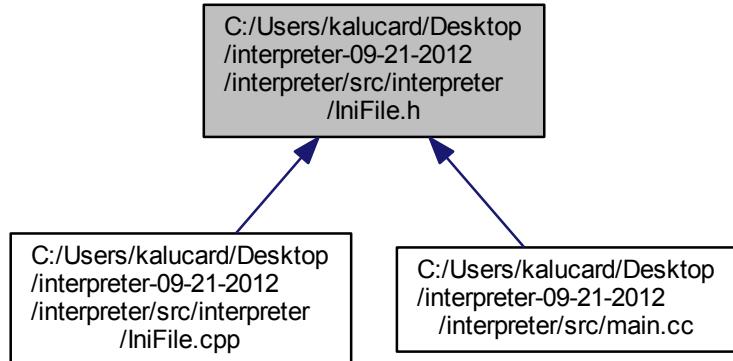
9.90 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/IniFile.h File Reference

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
```

Include dependency graph for IniFile.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [IniFile](#)
- struct [INIFILE_ENTRY](#)

Macros

- #define [COMMENT_CHAR](#) ';' /* signifies a comment */
- #define [INI_DEFAULT](#) 1
- #define [INI_INVALID](#) 2

- `#define INI_OK 0`
- `#define INIFILE_MAX_LINELEN 256 /* max number of chars in a line */`

9.90.1 Macro Definition Documentation

9.90.1.1 `#define COMMENT_CHAR ';' /* signifies a comment */`

Definition at line 11 of file IniFile.h.

9.90.1.2 `#define INI_DEFAULT 1`

Definition at line 14 of file IniFile.h.

9.90.1.3 `#define INI_INVALID 2`

Definition at line 15 of file IniFile.h.

9.90.1.4 `#define INI_OK 0`

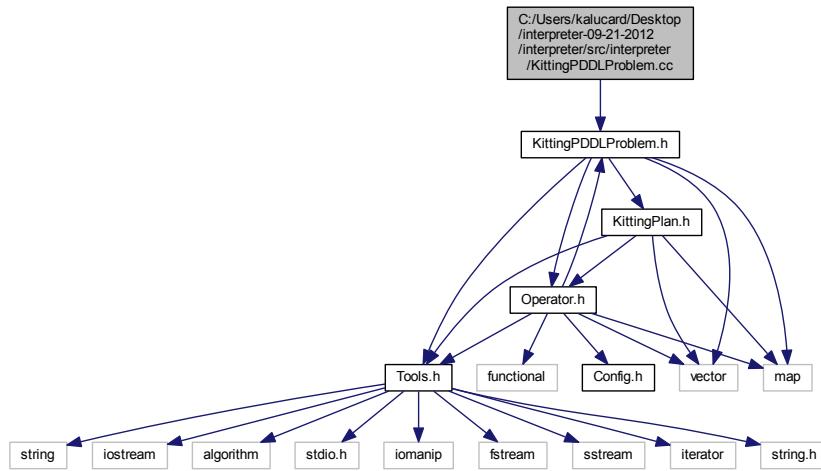
Definition at line 13 of file IniFile.h.

9.90.1.5 `#define INIFILE_MAX_LINELEN 256 /* max number of chars in a line */`

Definition at line 12 of file IniFile.h.

9.91 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/KittingPDDLProblem.cc File Reference

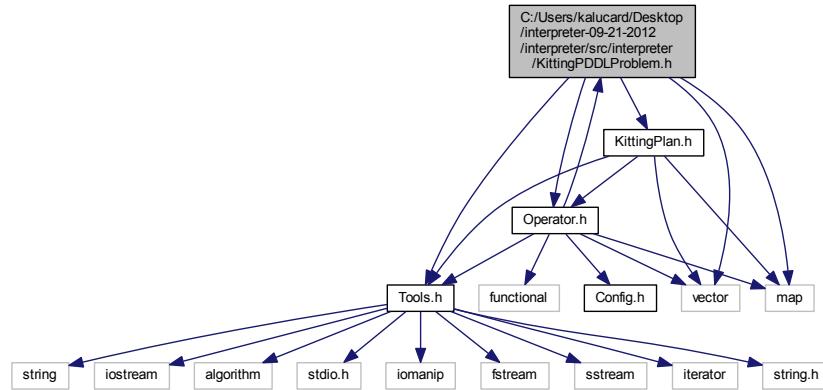
```
#include "KittingPDDLProblem.h"
Include dependency graph for KittingPDDLProblem.cc:
```



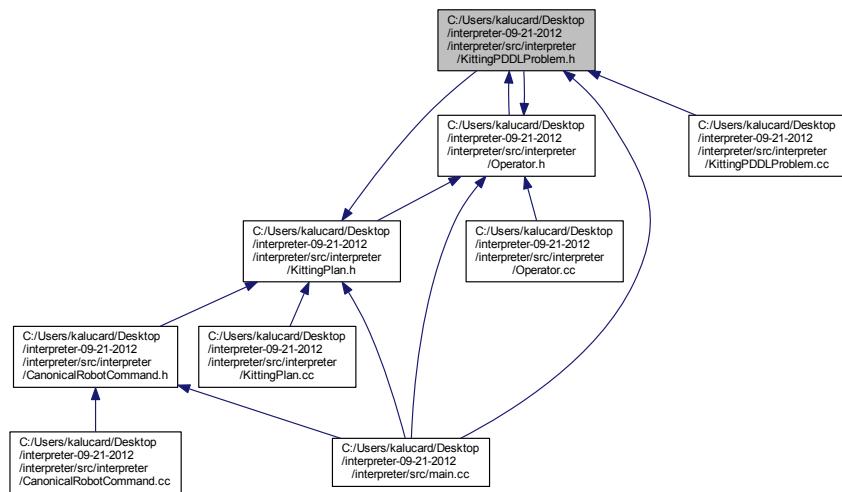
9.92 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/KittingPDDLProblem.h File Reference

```
#include "Tools.h"
#include <vector>
#include <map>
#include "Operator.h"
#include "KittingPlan.h"
```

Include dependency graph for KittingPDDLProblem.h:



This graph shows which files directly or indirectly include this file:



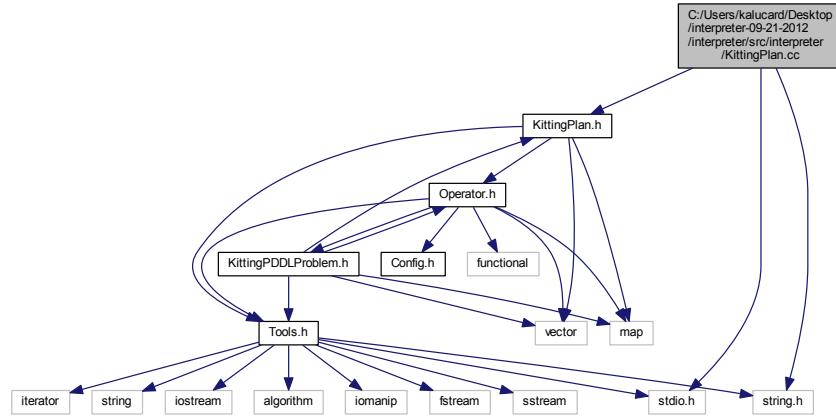
Classes

- class [KittingPDDLProblem](#)
Class for the Kitting PDDL Problem.

9.93 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/KittingPlan.cc File Reference

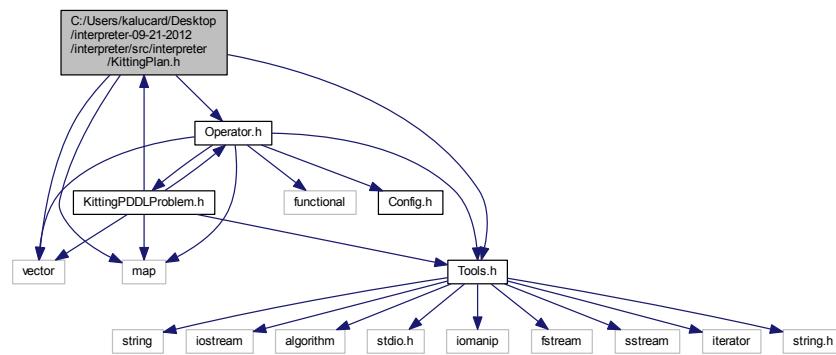
```
#include "KittingPlan.h"
```

```
#include <stdio.h>
#include <string.h>
Include dependency graph for KittingPlan.cc:
```

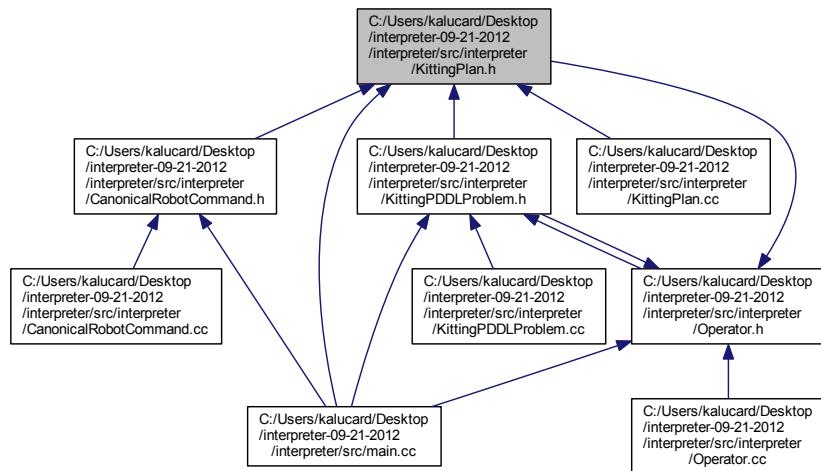


9.94 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/KittingPlan.h File Reference

```
#include "Tools.h"
#include <vector>
#include <map>
#include "Operator.h"
Include dependency graph for KittingPlan.h:
```

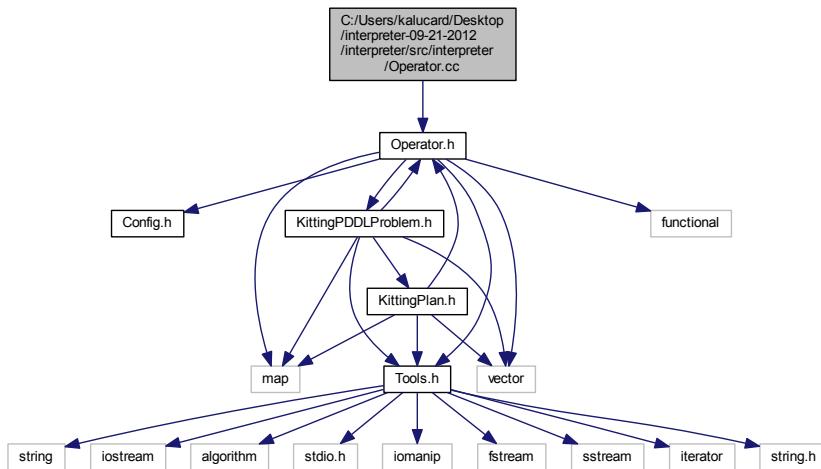


This graph shows which files directly or indirectly include this file:



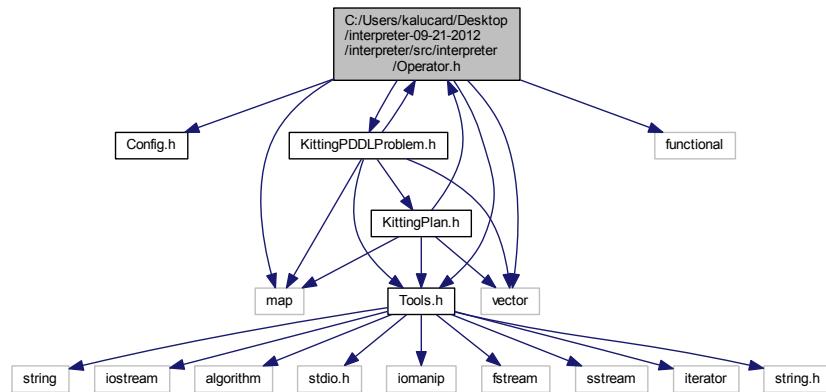
9.95 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/Operator.cc
File Reference

```
#include "Operator.h"
Include dependency graph for Operator.cc:
```

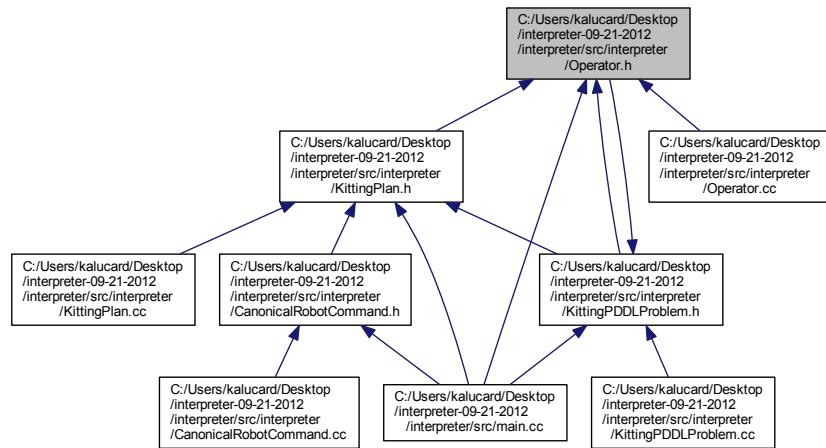


9.96 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/Operator.h File Reference

```
#include "Config.h"
#include "Tools.h"
#include "KittingPDDLProblem.h"
#include <vector>
#include <map>
#include <functional>
Include dependency graph for Operator.h:
```



This graph shows which files directly or indirectly include this file:



Classes

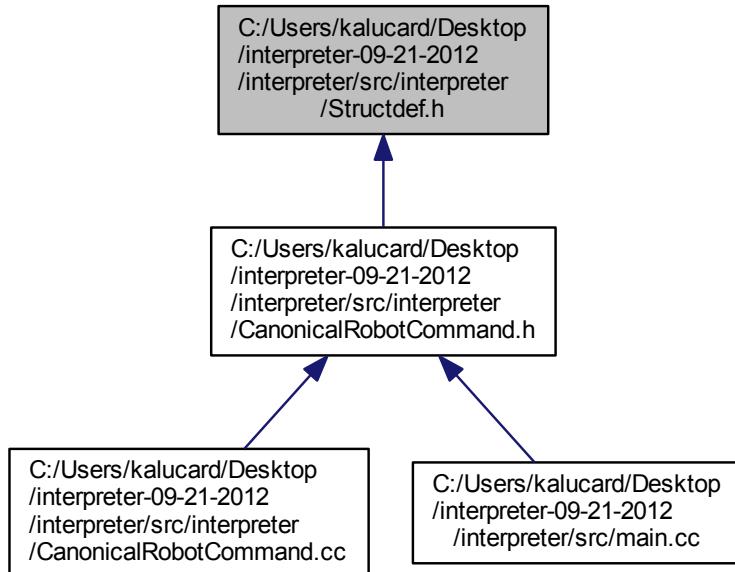
- class [FileOperator](#)

Class for operations on files.

9.97 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/Structdef.h File Reference

Structures used to define the location for different structures.

This graph shows which files directly or indirectly include this file:



Classes

- struct [KitTrayLocStruct](#)

This structure represents a kit tray location.

- struct [PartLocStruct](#)

Represents a part location.

- struct [PartsTrayLocStruct](#)

This structure represents a parts tray location.

9.97.1 Detailed Description

Structures used to define the location for different structures.

Author

Zeid Kootbally zeid.kootbally@nist.gov

Date

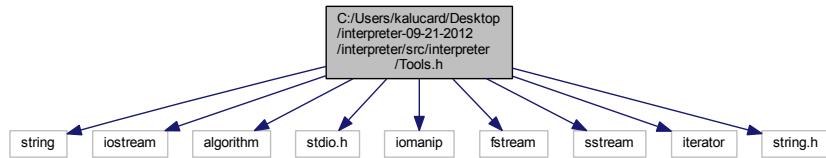
May 17, 2012

Definition in file [Structdef.h](#).

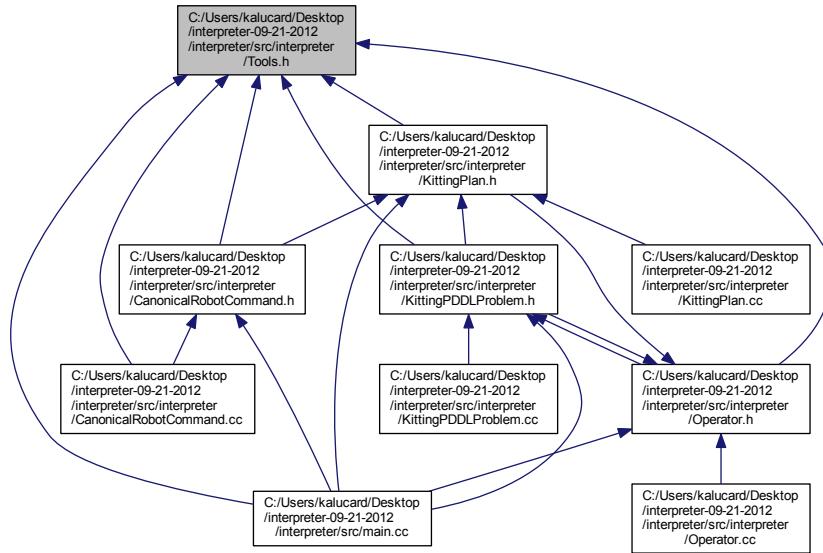
9.98 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter/Tools.h File Reference

```
#include <string>
#include <iostream>
#include <algorithm>
#include <stdio.h>
#include <iomanip>
#include <fstream>
#include <sstream>
#include <iterator>
#include <string.h>
```

Include dependency graph for Tools.h:



This graph shows which files directly or indirectly include this file:



9.99 C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/main.cc File Reference

```

#include "interpreter/Operator.h"
#include "interpreter/KittingPlan.h"
#include "interpreter/KittingPDDLProblem.h"
#include "interpreter/Config.h"
#include "interpreter/Tools.h"
#include "interpreter/CanonicalRobotCommand.h"
#include "interpreter/IniFile.h"
Include dependency graph for main.cc:
  
```



Functions

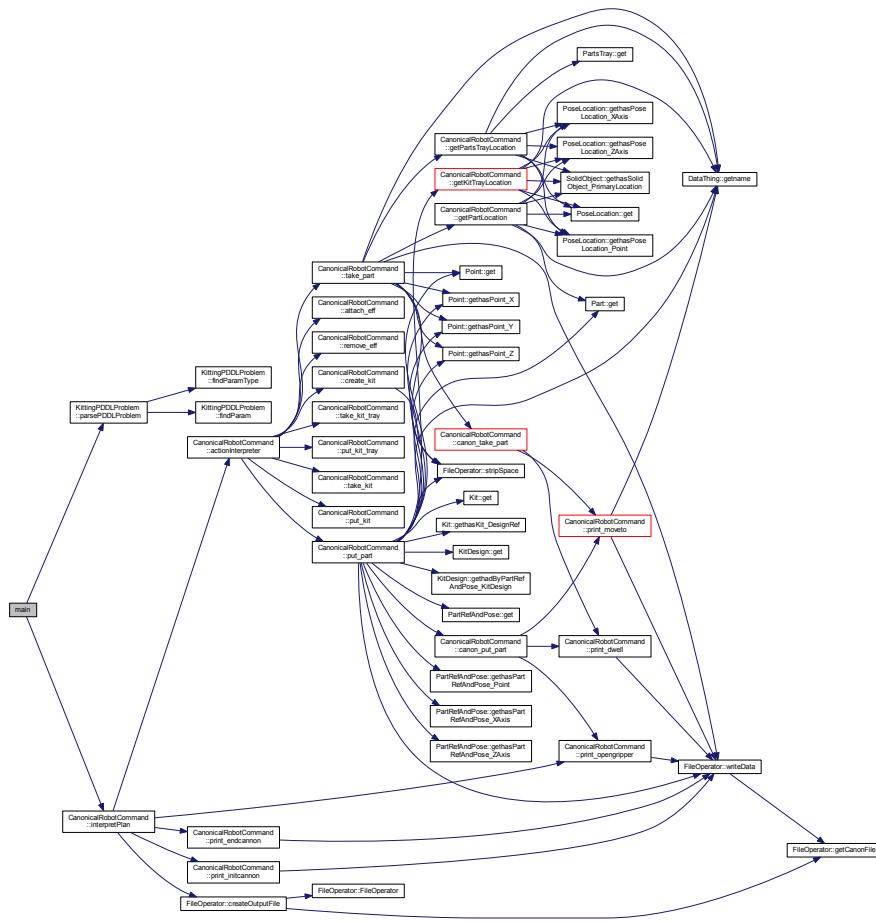
- int `main` (int argc, const char *argv[])

9.99.1 Function Documentation

9.99.1.1 int main (int argc, const char * argv[])

Definition at line 36 of file main.cc.

Here is the call graph for this function:



Index

- ~BoxVolume
 - BoxVolume, 20
- ~BoxyObject
 - BoxyObject, 27
- ~CanonicalRobotCommand
 - CanonicalRobotCommand, 38
- ~CloseGripperMsg
 - CloseGripperMsg, 60
- ~Connection
 - Connection, 61
- ~Controller
 - Controller, 66
- ~DAO
 - DAO, 77
- ~DataThing
 - DataThing, 86
- ~DwellMsg
 - DwellMsg, 94
- ~EndCanonMsg
 - EndCanonMsg, 97
- ~EndEffector
 - EndEffector, 100
- ~EndEffectorChangingStation
 - EndEffectorChangingStation, 108
- ~EndEffectorHolder
 - EndEffectorHolder, 114
- ~FileOperator
 - FileOperator, 119
- ~GripperEffector
 - GripperEffector, 126
- ~IniFile
 - IniFile, 130
- ~InitCanonMsg
 - InitCanonMsg, 135
- ~Kit
 - Kit, 138
- ~KitDesign
 - KitDesign, 146
- ~KitTray
 - KitTray, 168
- ~KittingPDDLProblem
 - KittingPDDLProblem, 152
- ~KittingWorkstation
 - KittingWorkstation, 160
- ~LargeBoxWithEmptyKitTrays
 - LargeBoxWithEmptyKitTrays, 177
- ~LargeBoxWithKits
 - LargeBoxWithKits, 183
- ~LargeContainer
 - LargeContainer, 190
- ~MessageMsg
 - MessageMsg, 196
- ~MoveSmoothlyToMsg
 - MoveSmoothlyToMsg, 199
- ~MoveStraightToMsg
 - MoveStraightToMsg, 202
- ~MoveToMsg
 - MoveToMsg, 204
- ~MyController
 - MyController, 207
- ~OpenGripperMsg
 - OpenGripperMsg, 212
- ~Part
 - Part, 215
- ~PartRefAndPose
 - PartRefAndPose, 224
- ~PartsBin
 - PartsBin, 231
- ~PartsTray
 - PartsTray, 237
- ~PartsTrayWithParts
 - PartsTrayWithParts, 245
- ~PhysicalLocation
 - PhysicalLocation, 251
- ~Point
 - Point, 259
- ~PoseLocation
 - PoseLocation, 268
- ~PoseLocationIn
 - PoseLocationIn, 276
- ~PoseLocationOn
 - PoseLocationOn, 281
- ~PoseOnlyLocation
 - PoseOnlyLocation, 286
- ~RelativeLocation
 - RelativeLocation, 291
- ~RelativeLocationIn
 - RelativeLocationIn, 297
- ~RelativeLocationOn
 - RelativeLocationOn, 302

~Robot
 Robot, 307

~SetAbsoluteAccelerationMsg
 SetAbsoluteAccelerationMsg, 314

~SetAngleUnitsMsg
 SetAngleUnitsMsg, 317

~SetEndAngleToleranceMsg
 SetEndAngleToleranceMsg, 320

~SetEndPointToleranceMsg
 SetEndPointToleranceMsg, 323

~SetIntermediatePointToleranceMsg
 SetIntermediatePointToleranceMsg, 326

~SetLengthUnitsMsg
 SetLengthUnitsMsg, 329

~SetRelativeAccelerationMsg
 SetRelativeAccelerationMsg, 332

~SetRelativeSpeedMsg
 SetRelativeSpeedMsg, 335

~ShapeDesign
 ShapeDesign, 338

~SolidObject
 SolidObject, 345

~StockKeepingUnit
 StockKeepingUnit, 356

~VacuumEffector
 VacuumEffector, 365

~VacuumEffectorMultiCup
 VacuumEffectorMultiCup, 371

~VacuumEffectorSingleCup
 VacuumEffectorSingleCup, 377

~Vector
 Vector, 382

~WorkTable
 WorkTable, 390

acceleration
 SetAbsoluteAccelerationMsg, 314

actionInterpreter
 CanonicalRobotCommand, 38

addr
 unix_shm_struct, 361

afterequal
 IniFile.cpp, 482

attach_eff
 CanonicalRobotCommand, 39

BoxVolume, 17
 ~BoxVolume, 20
 BoxVolume, 20
 BoxVolumeID, 23
 BoxVolume, 20
 copy, 21
 dao, 23
 Explode, 21
 get, 21

 getBoxVolumeID, 22
 getdao, 22
 gethasBoxVolume_MaximumPoint, 22
 gethasBoxVolume_MinimumPoint, 22
 gethasRobot_WorkVolume, 22
 gethasWorkstation_OtherObstacles, 22
 hasBoxVolume_MaximumPoint, 23
 hasBoxVolume_MinimumPoint, 24
 hasRobot_WorkVolume, 24
 hasWorkstation_OtherObstacles, 24
 set, 22
 setdao, 23
 sethasBoxVolume_MaximumPoint, 23
 sethasBoxVolume_MinimumPoint, 23
 sethasRobot_WorkVolume, 23
 sethasWorkstation_OtherObstacles, 23

BoxVolumeID
 BoxVolume, 23

BoxyObject, 24
 ~BoxyObject, 27
 BoxyObject, 27
 BoxyObjectID, 30
 BoxyObject, 27
 copy, 27
 dao, 30
 Explode, 28
 get, 28
 getBoxyObjectID, 29
 getdao, 29
 gethasBox_Height, 29
 gethasBox_Length, 29
 gethasBox_Width, 29
 hasBox_Height, 30
 hasBox_Length, 30
 hasBox_Width, 31
 set, 29
 setdao, 30
 sethasBox_Height, 30
 sethasBox_Length, 30
 sethasBox_Width, 30

BoxyObjectID
 BoxyObject, 30

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/
 Msg.hh, 395

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/
 cpp, 398

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/
 hh, 399

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/
 Controller.cpp, 401

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/controller/
 Controller.hh, 401

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KittingWorkstation.h, 438
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KitTrays.cpp, 441
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KitTrays.h, 441
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/BoxVolume.cpp, 424
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/BoxVolume.h, 424
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/BoxyObject.cpp, 425
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/BoxyObject.h, 425
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Connection.cpp, 426
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Connection.h, 427
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/DAO.cpp, 427
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/DAO.h, 428
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/DataThing.cpp, 428
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/DataThing.h, 428
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffector.cpp, 429
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffector.h, 430
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffectorChangingStation.cpp, 431
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffectorChangingStation.h, 431
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffectorHolder.cpp, 432
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/EndEffectorHolder.h, 432
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/GripperEffector.cpp, 433
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/GripperEffector.h, 434
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Kit.cpp, 435
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Kit.h, 435
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KitDesign.cpp, 436
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KitDesign.h, 437
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KitTray.cpp, 439
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KitTray.h, 440
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KittingWorkstation.cpp, 438
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/KittingWorkstation.h, 438
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/LargeBoxWithEmptyKitTrays.cpp, 441
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/LargeBoxWithEmptyKitTrays.h, 441
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/LargeBoxWithKits.cpp, 442
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/LargeBoxWithKits.h, 442
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/LargeContainer.cpp, 443
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/LargeContainer.h, 444
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Part.cpp, 445
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Part.h, 445
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartRefAndPose.cpp, 446
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartRefAndPose.h, 447
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsBin.cpp, 448
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsBin.h, 448
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsTray.cpp, 449
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsTray.h, 449
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsTrayWithParts.cpp, 451
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PartsTrayWithParts.h, 451
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PhysicalLocation.cpp, 452
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PhysicalLocation.h, 452
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Point.cpp, 453
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/Point.h, 454
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseLocation.cpp, 455
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseLocation.h, 455
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseLocationIn.cpp, 456
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseLocationIn.h, 456
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseLocationOn.cpp, 457
C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/database/PoseLocationOn.h, 458

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 PoseOnlyLocation.cpp, 459 CanonicalRobotCommand.h, 478

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 PoseOnlyLocation.h, 459 Config.h, 480

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 RelativeLocation.cpp, 461 IniFile.cpp, 482

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 RelativeLocation.h, 461 IniFile.h, 483

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 RelativeLocationIn.cpp, 462 KittingPDDLProblem.cc, 486

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 RelativeLocationIn.h, 462 KittingPDDLProblem.h, 486

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 RelativeLocationOn.cpp, 464 KittingPlan.cc, 487

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 RelativeLocationOn.h, 464 KittingPlan.h, 488

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 Robot.cpp, 465 Operator.cc, 489

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 Robot.h, 465 Operator.h, 490

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 ShapeDesign.cpp, 466 Structdef.h, 491

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 ShapeDesign.h, 467 Tools.h, 492

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/main.-
 cc, 493

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 SolidObject.cpp, 468 COMMENT CHAR

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 SolidObject.h, 468 IniFile.h, 485

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 StockKeepingUnit.cpp, 469 CanonicalRobotCommand, 40

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 StockKeepingUnit.h, 470 CanonicalRobotCommand, 41

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 VacuumEffector.cpp, 471 CloseGripper, 397

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 VacuumEffector.h, 471 EndCanon, 397

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 VacuumEffectorMultiCup.cpp, 472 Message, 397

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 VacuumEffectorMultiCup.h, 472 MoveSmoothlyTo, 397

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 VacuumEffectorSingleCup.cpp, 474 MoveStraightTo, 397

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 VacuumEffectorSingleCup.h, 474 OpenGripper, 397

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 SetAbsoluteAcceleration, 397

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 SetAngleUnits, 397

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 SetSingleTolerance, 397

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 Vector.cpp, 475 SetEndPointTolerance, 397

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 Vector.h, 475 SetIntermediatePointTolerance, 398

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 SetLengthUnits, 398

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 SetRelativeAcceleration, 398

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 WorkTable.cpp, 476 SetRelativeSpeed, 398

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 WorkTable.h, 477 CanonicalRobotCommand/31

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 msgID, 31

C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/C:/Users/kalucard/Desktop/interpreter-09-21-2012/interpreter/src/interpreter
 Interpreter/31

CanonicalRobotCommand.cc, 478 time, 32

CanonicalMsg, 32
 CanonicalMsg, 33
 CanonicalMsg, 33
 getMsgID, 33
 getTime, 33
 getType, 34
 hdr, 35
 nextID, 35
 setHeader, 34
canonicalMsg.hh
 CanonicalType, 397
CanonicalRobotCommand, 35
 ~CanonicalRobotCommand, 38
 actionInterpreter, 38
 attach_eff, 39
 canon_put_part, 40
 canon_take_part, 41
 CanonicalRobotCommand, 38
 CanonicalRobotCommand, 38
 create_kit, 41
 getKitTrayLocation, 42
 getPartLocation, 44
 getPartsTrayLocation, 45
 interpretPlan, 47
 m_dwell, 57
 m_kit_tray, 57
 m_safe_z, 57
 print_closegripper, 48
 print_dwell, 49
 print_endcannon, 49
 print_initcannon, 50
 print_moveto, 51
 print.opengripper, 52
 put_kit, 53
 put_kit_tray, 53
 put_part, 53
 remove_eff, 54
 take_kit, 55
 take_kit_tray, 55
 take_part, 56
CanonicalType
 canonicalMsg.hh, 397
className
 DAO, 83
close
 IniFile, 130
CloseGripper
 canonicalMsg.hh, 397
CloseGripperMsg, 57
 ~CloseGripperMsg, 60
 CloseGripperMsg, 59
 CloseGripperMsg, 59
cmdQueue
 Controller, 74
con
 Connection, 63
Config.h
 PDDL_DOMAIN, 480
 PDDL_FOLDER, 481
 PDDL_PROBLEM, 481
 PLAN_FILE, 481
 PLAN_FOLDER, 481
 ROBOT_COMMANDS_FILE, 481
Connection, 60
 ~Connection, 61
 con, 63
 Connection, 61
 driver, 63
 getCon, 61
 getDriver, 62
 setCon, 62
 setDriver, 63
connection
 DAO, 83
Controller, 63
 ~Controller, 66
 cmdQueue, 74
 Controller, 66
 dequeueMsg, 66
 processCloseGripper, 67
 processDwell, 68
 processEndCanon, 68
 processInitCanon, 68
 processMessage, 69
 processMoveSmoothlyTo, 69
 processMoveStraightTo, 69
 processMoveTo, 70
 processOpenGripper, 70
 processSetAbsoluteAcceleration, 70
 processSetAngleUnits, 71
 processSetEndAngleTolerance, 71
 processSetEndPointTolerance, 71
 processSetIntermediatePointTolerance, 72
 processSetLengthUnits, 72
 processSetRelativeAcceleration, 72
 processSetRelativeSpeed, 73
 queue_length, 74
 queueCloseGripper, 73
 queueDwell, 73
 queueMsg, 73
copy
 BoxVolume, 21
 BoxyObject, 27
 DataThing, 86
 EndEffector, 101
 EndEffectorChangingStation, 108
 EndEffectorHolder, 114
 GripperEffector, 126

Kit, 139
 KitDesign, 147
 KittingWorkstation, 160
 KitTray, 169
 LargeBoxWithEmptyKitTrays, 177
 LargeBoxWithKits, 184
 LargeContainer, 191
 Part, 215
 PartRefAndPose, 224
 PartsBin, 231
 PartsTray, 237
 PartsTrayWithParts, 245
 PhysicalLocation, 251
 Point, 260
 PoseLocation, 268
 PoseLocationIn, 276
 PoseLocationOn, 281
 PoseOnlyLocation, 286
 RelativeLocation, 291
 RelativeLocationIn, 297
 RelativeLocationOn, 302
 Robot, 308
 ShapeDesign, 338
 SolidObject, 345
 StockKeepingUnit, 357
 VacuumEffector, 365
 VacuumEffectorMultiCup, 371
 VacuumEffectorSingleCup, 377
 Vector, 383
 WorkTable, 390
 copyVector
 FileOperator, 119
 create_kit
 CanonicalRobotCommand, 41
 createOutputFile
 FileOperator, 119

 DAO, 74
 ~DAO, 77
 className, 83
 connection, 83
 DAO, 76
 DAO, 76
 Explode, 77
 fillGetSqlQueries, 78
 get, 79
 getSqlQueriesDataMulti, 83
 getSqlQueriesDataSingle, 83
 getSqlQueriesObjectMulti, 83
 getSqlQueriesObjectSingle, 83
 getClassName, 80
 getConnection, 80
 map, 83
 nameDone, 83

 path, 83
 pathmulti, 83
 set, 81
 setSqlQueries, 83
 setClassName, 82
 setconnection, 82
 updateSqlQueries, 84
 DIGITS_IN
 ulapi.hh, 414
 dao
 BoxVolume, 23
 BoxyObject, 30
 DataThing, 91
 EndEffector, 104
 EndEffectorChangingStation, 111
 EndEffectorHolder, 116
 GripperEffector, 128
 Kit, 142
 KitDesign, 150
 KittingWorkstation, 164
 KitTray, 171
 LargeBoxWithEmptyKitTrays, 180
 LargeBoxWithKits, 186
 LargeContainer, 193
 Part, 218
 PartRefAndPose, 228
 PartsBin, 233
 PartsTray, 240
 PartsTrayWithParts, 248
 PhysicalLocation, 255
 Point, 264
 PoseLocation, 272
 PoseLocationIn, 278
 PoseLocationOn, 283
 PoseOnlyLocation, 288
 RelativeLocation, 294
 RelativeLocationIn, 299
 RelativeLocationOn, 304
 Robot, 311
 ShapeDesign, 341
 SolidObject, 352
 StockKeepingUnit, 360
 VacuumEffector, 368
 VacuumEffectorMultiCup, 373
 VacuumEffectorSingleCup, 379
 Vector, 387
 WorkTable, 393
 DataThing, 84
 ~DataThing, 86
 copy, 86
 dao, 91
 DataThing, 86
 DataThingID, 91
 DataThing, 86

Explode, 87
get, 87, 88
getDataThingID, 88
getdao, 88
getname, 88
name, 91
set, 89, 90
setdao, 91
DataThingID
 DataThing, 91
dequeueMsg
 Controller, 66
driver
 Connection, 63
Dwell
 canonicalMsg.hh, 397
DwellMsg, 91
 ~DwellMsg, 94
 DwellMsg, 94
 DwellMsg, 94
 time, 94

EndCanon
 canonicalMsg.hh, 397
EndCanonMsg, 94
 ~EndCanonMsg, 97
 EndCanonMsg, 97
 EndCanonMsg, 97
 reason, 97
EndEffector, 97
 ~EndEffector, 100
 copy, 101
 dao, 104
 EndEffector, 100
 EndEffectorID, 104
 EndEffector, 100
 Explode, 101
 get, 101
 getEndEffectorID, 102
 setdao, 102
 gethadByEndEffector_Robot, 102
 gethasEffector_MaximumLoadWeight, 102
 gethasEndEffector_Description, 102
 gethasEndEffector_Id, 102
 gethasEndEffector_Weight, 102
 gethasEndEffectorHolder_EndEffector, 102
 hadByEndEffector_Robot, 104
 hasEffector_MaximumLoadWeight, 104
 hasEndEffector_Description, 104
 hasEndEffector_Id, 104
 hasEndEffector_Weight, 105
 hasEndEffectorHolder_EndEffector, 105
 set, 102, 103
 setdao, 103
 sethadByEndEffector_Robot, 103
 sethasEffector_MaximumLoadWeight, 104
 sethasEndEffector_Description, 104
 sethasEndEffector_Id, 104
 sethasEndEffector_Weight, 104
 sethasEndEffectorHolder_EndEffector, 104
EndEffectorChangingStation, 105
 ~EndEffectorChangingStation, 108
 copy, 108
 dao, 111
 EndEffectorChangingStation, 108
 EndEffectorChangingStationID, 111
 EndEffectorChangingStation, 108
 Explode, 109
 get, 109
 getEndEffectorChangingStationID, 110
 setdao, 110
 gethadByEndEffectorHolder_ChangingStation, 110
 gethasWorkstation_ChangingStation, 110
 hadByEndEffectorHolder_ChangingStation, 111
 hasWorkstation_ChangingStation, 111
 set, 110
 setdao, 110
 sethadByEndEffectorHolder_ChangingStation, 111
 sethasWorkstation_ChangingStation, 111
EndEffectorChangingStationID
 EndEffectorChangingStation, 111
EndEffectorHolder, 111
 ~EndEffectorHolder, 114
 copy, 114
 dao, 116
 EndEffectorHolder, 114
 EndEffectorHolderID, 116
 EndEffectorHolder, 114
 Explode, 115
 get, 115
 getEndEffectorHolderID, 115
 setdao, 115
 gethadByEndEffectorHolder_ChangingStation, 115
 gethasEndEffectorHolder_EndEffector, 115
 hadByEndEffectorHolder_ChangingStation, 116
 hasEndEffectorHolder_EndEffector, 117
 set, 116
 setdao, 116
 sethadByEndEffectorHolder_ChangingStation, 116
 sethasEndEffectorHolder_EndEffector, 116
EndEffectorHolderID
 EndEffectorHolder, 116
EndEffectorID
 EndEffector, 104
Explode
 BoxVolume, 21
 BoxyObject, 28
 DAO, 77

DataThing, 87
 EndEffector, 101
 EndEffectorChangingStation, 109
 EndEffectorHolder, 115
 GripperEffector, 126
 Kit, 139
 KitDesign, 147
 KittingWorkstation, 160
 KitTray, 169
 LargeBoxWithEmptyKitTrays, 178
 LargeBoxWithKits, 184
 LargeContainer, 191
 Part, 216
 PartRefAndPose, 225
 PartsBin, 232
 PartsTray, 238
 PartsTrayWithParts, 246
 PhysicalLocation, 252
 Point, 260
 PoseLocation, 269
 PoseLocationIn, 276
 PoseLocationOn, 281
 PoseOnlyLocation, 286
 RelativeLocation, 292
 RelativeLocationIn, 297
 RelativeLocationOn, 302
 Robot, 308
 ShapeDesign, 339
 SolidObject, 346
 StockKeepingUnit, 357
 VacuumEffector, 366
 VacuumEffectorMultiCup, 372
 VacuumEffectorSingleCup, 377
 Vector, 383
 WorkTable, 391

FileOperator, 117
 ~FileOperator, 119
 copyVector, 119
 createOutputFile, 119
 FileOperator, 119
 FileOperator, 119
 getCanonFile, 120
 openFile, 120
 readMap, 120
 readMapOfVector, 120
 readVector, 120
 readVectorOfVector, 121
 removeDuplicates, 121
 removeParentheses, 121
 splitString, 121
 stripSpace, 122
 writeData, 122

fillGetSqlQueries

DAO, 78
 find
 IniFile, 130
 findParam
 KittingPDDLProblem, 152
 findParamType
 KittingPDDLProblem, 153
 findSection
 IniFile, 130

get
 BoxVolume, 21
 BoxyObject, 28
 DAO, 79
 DataThing, 87, 88
 EndEffector, 101
 EndEffectorChangingStation, 109
 EndEffectorHolder, 115
 GripperEffector, 127
 Kit, 139, 140
 KitDesign, 147, 148
 KittingWorkstation, 161
 KitTray, 169
 LargeBoxWithEmptyKitTrays, 178
 LargeBoxWithKits, 184
 LargeContainer, 191
 Part, 216
 PartRefAndPose, 225
 PartsBin, 232
 PartsTray, 238
 PartsTrayWithParts, 246
 PhysicalLocation, 253
 Point, 260, 261
 PoseLocation, 269
 PoseLocationIn, 277
 PoseLocationOn, 282
 PoseOnlyLocation, 287
 RelativeLocation, 292
 RelativeLocationIn, 298
 RelativeLocationOn, 303
 Robot, 308
 ShapeDesign, 339
 SolidObject, 347
 StockKeepingUnit, 357
 VacuumEffector, 366
 VacuumEffectorMultiCup, 372
 VacuumEffectorSingleCup, 378
 Vector, 383, 384
 WorkTable, 391

getBoxVolumeID
 BoxVolume, 22
 getBoxyObjectID
 BoxyObject, 29
 getCanonFile

FileOperator, 120
getCon
 Connection, 61
getDataThingID
 DataThing, 88
getDriver
 Connection, 62
getEndEffectorChangingStationID
 EndEffectorChangingStation, 110
getEndEffectorHolderID
 EndEffectorHolder, 115
getEndEffectorID
 EndEffector, 102
getGripperEffectordID
 GripperEffectord, 127
getKitDesignID
 KitDesign, 149
getKitID
 Kit, 141
getKitTrayID
 KitTray, 170
getKitTrayLocation
 CanonicalRobotCommand, 42
getKittingWorkstationID
 KittingWorkstation, 162
getLargeBoxWithEmptyKitTraysID
 LargeBoxWithEmptyKitTrays, 179
getLargeBoxWithKitsID
 LargeBoxWithKits, 185
getLargeContainerID
 LargeContainer, 192
getMsgID
 CanonicalMsg, 33
getPartID
 Part, 217
getPartLocation
 CanonicalRobotCommand, 44
getPartRefAndPoseID
 PartRefAndPose, 226
getPartsBinID
 PartsBin, 233
getPartsTrayID
 PartsTray, 239
getPartsTrayLocation
 CanonicalRobotCommand, 45
getPartsTrayWithPartsID
 PartsTrayWithParts, 247
getPhysicalLocationID
 PhysicalLocation, 254
getPointID
 Point, 262
getPoseLocationID
 PoseLocation, 271
getPoseLocationInID
 PoseLocationIn, 277
getPoseLocationOnID
 PoseLocationOn, 282
getPoseOnlyLocationID
 PoseOnlyLocation, 287
getRelativeLocationID
 RelativeLocation, 293
getRelativeLocationInID
 RelativeLocationIn, 298
getRelativeLocationOnID
 RelativeLocationOn, 303
getRobotID
 Robot, 309
getShapeDesignID
 ShapeDesign, 340
getSolidObjectID
 SolidObject, 349
getSqlQueriesDataMulti
 DAO, 83
getSqlQueriesDataSingle
 DAO, 83
getSqlQueriesObjectMulti
 DAO, 83
getSqlQueriesObjectSingle
 DAO, 83
getStockKeepingUnitID
 StockKeepingUnit, 358
getTime
 CanonicalMsg, 33
getType
 CanonicalMsg, 34
getVacuumEffectordID
 VacuumEffectord, 367
getVacuumEffectormultiCupID
 VacuumEffectormultiCup, 373
getVacuumEffectorsingleCupID
 VacuumEffectorsingleCup, 378
getVectorID
 Vector, 385
getWorkTableID
 WorkTable, 392
getclassName
 DAO, 80
getconnection
 DAO, 80
getdao
 BoxVolume, 22
 BoxyObject, 29
 DataThing, 88
 EndEffector, 102
 EndEffectorChangingStation, 110
 EndEffectorHolder, 115
 GripperEffectord, 127
 Kit, 140

KitDesign, 148
 KittingWorkstation, 161
 KitTray, 170
 LargeBoxWithEmptyKitTrays, 179
 LargeBoxWithKits, 185
 LargeContainer, 191
 Part, 216
 PartRefAndPose, 225
 PartsBin, 232
 PartsTray, 238
 PartsTrayWithParts, 247
 PhysicalLocation, 253
 Point, 261
 PoseLocation, 270
 PoseLocationIn, 277
 PoseLocationOn, 282
 PoseOnlyLocation, 287
 RelativeLocation, 292
 RelativeLocationIn, 298
 RelativeLocationOn, 303
 Robot, 309
 ShapeDesign, 340
 SolidObject, 348
 StockKeepingUnit, 358
 VacuumEffector, 366
 VacuumEffectorMultiCup, 372
 VacuumEffectorSingleCup, 378
 Vector, 384
 WorkTable, 392
 gethadByEndEffector_Robot
 EndEffector, 102
 Robot, 309
 gethadByEndEffectorHolder_ChangingStation
 EndEffectorChangingStation, 110
 EndEffectorHolder, 115
 gethadByKit_LargeBoxWithKits
 Kit, 140
 LargeBoxWithKits, 185
 gethadByKitDesign_Workstation
 KitDesign, 148
 KittingWorkstation, 161
 gethadByKitTray_LargeBoxWithEmptyKitTrays
 KitTray, 170
 LargeBoxWithEmptyKitTrays, 179
 gethadByPart_Kit
 Kit, 140
 Part, 217
 gethadByPart_PartsTrayWithParts
 Part, 217
 PartsTrayWithParts, 247
 gethadByPartRefAndPose_KitDesign
 KitDesign, 148
 PartRefAndPose, 226
 gethadBySku_Workstation
 KittingWorkstation, 161
 StockKeepingUnit, 358
 gethadBySolidObject_WorkTable
 SolidObject, 348
 WorkTable, 392
 gethasBin_PartQuantity
 PartsBin, 232
 gethasBin_PartSkuRef
 PartsBin, 232
 gethasBox_Height
 BoxyObject, 29
 gethasBox_Length
 BoxyObject, 29
 gethasBox_Width
 BoxyObject, 29
 gethasBoxVolume_MaximumPoint
 BoxVolume, 22
 Point, 261
 gethasBoxVolume_MinimumPoint
 BoxVolume, 22
 Point, 261
 gethasEffector_MaximumLoadWeight
 EndEffector, 102
 gethasEndEffector_Description
 EndEffector, 102
 gethasEndEffector_Id
 EndEffector, 102
 gethasEndEffector_Weight
 EndEffector, 102
 gethasEndEffectorHolder_EndEffector
 EndEffector, 102
 EndEffectorHolder, 115
 gethasKit_DesignRef
 Kit, 140
 gethasKit_Tray
 Kit, 141
 KitTray, 170
 gethasKitDesign_Id
 KitDesign, 148
 gethasKitDesign_KitTraySkuRef
 KitDesign, 149
 gethasKitTray_SerialNumber
 KitTray, 170
 gethasKitTray_SkuRef
 KitTray, 170
 gethasLargeBoxWithEmptyKitTrays_LargeContainer
 LargeBoxWithEmptyKitTrays, 179
 LargeContainer, 191
 gethasLargeBoxWithKits_Capacity
 LargeBoxWithKits, 185
 gethasLargeBoxWithKits_KitDesignRef
 LargeBoxWithKits, 185
 gethasLargeBoxWithKits_LargeContainer
 LargeBoxWithKits, 185

LargeContainer, 192
gethasLargeContainer_SerialNumber
 LargeContainer, 192
gethasLargeContainer_SkuRef
 LargeContainer, 192
gethasMultiCup_ArrayNumber
 VacuumEffectorMultiCup, 372
gethasMultiCup_ArrayRadius
 VacuumEffectorMultiCup, 372
gethasPart_SerialNumber
 Part, 217
gethasPart_SkuRef
 Part, 217
gethasPartRefAndPose_Point
 PartRefAndPose, 226
 Point, 261
gethasPartRefAndPose_Ref
 PartRefAndPose, 226
gethasPartRefAndPose_XAxis
 PartRefAndPose, 226
 Vector, 384
gethasPartRefAndPose_ZAxis
 PartRefAndPose, 226
 Vector, 384
gethasPartsTray_SerialNumber
 PartsTray, 239
gethasPartsTray_SkuRef
 PartsTray, 239
gethasPartsTrayWithParts_Tray
 PartsTray, 239
 PartsTrayWithParts, 247
gethasPhysicalLocation_RefObject
 PhysicalLocation, 253
 SolidObject, 348
gethasPoint_X
 Point, 261
gethasPoint_Y
 Point, 262
gethasPoint_Z
 Point, 262
gethasPoseLocation_Point
 Point, 262
 PoseLocation, 270
gethasPoseLocation_XAxis
 PoseLocation, 270
 Vector, 384
gethasPoseLocation_ZAxis
 PoseLocation, 270
 Vector, 384
gethasRelativeLocation_Description
 RelativeLocation, 292
gethasRobot_Description
 Robot, 309
gethasRobot_Id
 Robot, 309
gethasRobot_MaximumLoadWeight
 Robot, 309
gethasRobot_WorkVolume
 BoxVolume, 22
 Robot, 309
gethasShapeDesign_Description
 ShapeDesign, 340
gethasSku_Description
 StockKeepingUnit, 358
gethasSku_EndEffectorRefs
 StockKeepingUnit, 358
gethasSku_Id
 StockKeepingUnit, 358
gethasSku_Shape
 ShapeDesign, 340
 StockKeepingUnit, 358
gethasSku_Weight
 StockKeepingUnit, 358
gethasSolidObject_PrimaryLocation
 PhysicalLocation, 254
 SolidObject, 348
gethasSolidObject_SecondaryLocation
 PhysicalLocation, 254
 SolidObject, 348
gethasVacuumEffector_CupDiameter
 VacuumEffector, 366
gethasVacuumEffector_Length
 VacuumEffector, 367
gethasVector_I
 Vector, 385
gethasVector_J
 Vector, 385
gethasVector_K
 Vector, 385
gethasWorkstation_AngleUnit
 KittingWorkstation, 161
gethasWorkstation_ChangingStation
 EndEffectorChangingStation, 110
 KittingWorkstation, 161
gethasWorkstation_LengthUnit
 KittingWorkstation, 162
gethasWorkstation_OtherObstacles
 BoxVolume, 22
 KittingWorkstation, 162
gethasWorkstation_Robot
 KittingWorkstation, 162
 Robot, 309
gethasWorkstation_WeightUnit
 KittingWorkstation, 162
gethasWorkstation_WorkTable
 KittingWorkstation, 162
 WorkTable, 392
getisKit_Finished

Kit, 141
 getname
 DataThing, 88
 SolidObject, 348
 GripperEffector, 123
 ~GripperEffector, 126
 copy, 126
 dao, 128
 Explode, 126
 get, 127
 getGripperEffectorID, 127
 getdao, 127
 GripperEffector, 126
 GripperEffectorID, 128
 GripperEffector, 126
 set, 127
 setdao, 128
 GripperEffectorID
 GripperEffector, 128

 hadByEndEffector_Robot
 EndEffector, 104
 Robot, 311
 hadByEndEffectorHolder_ChangingStation
 EndEffectorChangingStation, 111
 EndEffectorHolder, 116
 hadByKit_LargeBoxWithKits
 Kit, 142
 LargeBoxWithKits, 186
 hadByKitDesign_Workstation
 KitDesign, 150
 KittingWorkstation, 164
 hadByKitTray_LargeBoxWithEmptyKitTrays
 KitTray, 171
 LargeBoxWithEmptyKitTrays, 180
 hadByPart_Kit
 Kit, 142
 Part, 218
 hadByPart_PartsTrayWithParts
 Part, 218
 PartsTrayWithParts, 248
 hadByPartRefAndPose_KitDesign
 KitDesign, 150
 PartRefAndPose, 228
 hadBySku_Workstation
 KittingWorkstation, 164
 StockKeepingUnit, 360
 hadBySolidObject_WorkTable
 SolidObject, 352
 WorkTable, 393
 hasBin_PartQuantity
 PartsBin, 233
 hasBin_PartSkuRef
 PartsBin, 234

 hasBox_Height
 BoxyObject, 30
 hasBox_Length
 BoxyObject, 30
 hasBox_Width
 BoxyObject, 31
 hasBoxVolume_MaximumPoint
 BoxVolume, 23
 Point, 264
 hasBoxVolume_MinimumPoint
 BoxVolume, 24
 Point, 264
 hasEffecto_MaximumLoadWeight
 EndEffector, 104
 hasEndEffector_Description
 EndEffector, 104
 hasEndEffector_Id
 EndEffector, 104
 hasEndEffector_Weight
 EndEffector, 105
 hasEndEffectorHolder_EndEffector
 EndEffector, 105
 EndEffectorHolder, 117
 hasKit_DesignRef
 Kit, 142
 hasKit_Tray
 Kit, 142
 KitTray, 172
 hasKitDesign_Id
 KitDesign, 150
 hasKitDesign_KitTraySkuRef
 KitDesign, 150
 hasKitTray_SerialNumber
 KitTray, 172
 hasKitTray_SkuRef
 KitTray, 172
 hasLargeBoxWithEmptyKitTrays_LargeContainer
 LargeBoxWithEmptyKitTrays, 180
 LargeContainer, 193
 hasLargeBoxWithKits_Capacity
 LargeBoxWithKits, 186
 hasLargeBoxWithKits_KitDesignRef
 LargeBoxWithKits, 187
 hasLargeBoxWithKits_LargeContainer
 LargeBoxWithKits, 187
 LargeContainer, 193
 hasLargeContainer_SerialNumber
 LargeContainer, 193
 hasLargeContainer_SkuRef
 LargeContainer, 193
 hasMultiCup_ArrayNumber
 VacuumEffectoMultiCup, 373
 hasMultiCup_ArrayRadius
 VacuumEffectoMultiCup, 374

hasPart_SerialNumber
 Part, 218
hasPart_SkuRef
 Part, 218
hasPartRefAndPose_Point
 PartRefAndPose, 228
 Point, 264
hasPartRefAndPose_Ref
 PartRefAndPose, 228
hasPartRefAndPose_XAxis
 PartRefAndPose, 228
 Vector, 387
hasPartRefAndPose_ZAxis
 PartRefAndPose, 228
 Vector, 387
hasPartsTray_SerialNumber
 PartsTray, 240
hasPartsTray_SkuRef
 PartsTray, 240
hasPartsTrayWithParts_Tray
 PartsTray, 240
 PartsTrayWithParts, 248
hasPhysicalLocation_RefObject
 PhysicalLocation, 255
 SolidObject, 353
hasPoint_X
 Point, 264
hasPoint_Y
 Point, 264
hasPoint_Z
 Point, 264
hasPoseLocation_Point
 Point, 264
 PoseLocation, 272
hasPoseLocation_XAxis
 PoseLocation, 272
 Vector, 387
hasPoseLocation_ZAxis
 PoseLocation, 272
 Vector, 387
hasRelativeLocation_Description
 RelativeLocation, 294
hasRobot_Description
 Robot, 311
hasRobot_Id
 Robot, 311
hasRobot_MaximumLoadWeight
 Robot, 311
hasRobot_WorkVolume
 BoxVolume, 24
 Robot, 311
hasShapeDesign_Description
 ShapeDesign, 341
hasSku_Description
 StockKeepingUnit, 360
hasSku_EndEffectorRefs
 StockKeepingUnit, 360
hasSku_Id
 StockKeepingUnit, 360
hasSku_Shape
 ShapeDesign, 341
 StockKeepingUnit, 360
hasSku_Weight
 StockKeepingUnit, 360
hasSolidObject_PrimaryLocation
 PhysicalLocation, 255
 SolidObject, 353
hasSolidObject_SecondaryLocation
 PhysicalLocation, 256
 SolidObject, 353
hasVacuumEffectuator_CupDiameter
 VacuumEffectuator, 368
hasVacuumEffectuator_Length
 VacuumEffectuator, 368
hasVector_I
 Vector, 387
hasVector_J
 Vector, 387
hasVector_K
 Vector, 387
hasWorkstation_AngleUnit
 KittingWorkstation, 164
hasWorkstation_ChangingStation
 EndEffectorChangingStation, 111
 KittingWorkstation, 164
hasWorkstation_LengthUnit
 KittingWorkstation, 164
hasWorkstation_OtherObstacles
 BoxVolume, 24
 KittingWorkstation, 164
hasWorkstation_Robot
 KittingWorkstation, 164
 Robot, 311
hasWorkstation_WeightUnit
 KittingWorkstation, 164
hasWorkstation_WorkTable
 KittingWorkstation, 164
 WorkTable, 393
hdr
 CanonicalMsg, 35
INI_DEFAULT
 IniFile.h, 485
INI_INVALID
 IniFile.h, 485
INI_OK
 IniFile.h, 485
INIFILE_ENTRY, 132

rest, 132
 tag, 132
INIFILE_MAX_LINELEN
 IniFile.h, 485
id
 unix_shm_struct, 361
IniFile, 128
 ~IniFile, 130
 close, 130
 find, 130
 findSection, 130
 IniFile, 130
 iniFind, 130
 iniSection, 131
 IniFile, 130
 isSection, 131
 open, 131
 operator=, 131
 section, 131
 valid, 131
IniFile.cpp
 afterequal, 482
 skipwhite, 483
IniFile.h
 COMMENT_CHAR, 485
 INI_DEFAULT, 485
 INI_INVALID, 485
 INI_OK, 485
iniFind
 IniFile, 130
iniSection
 IniFile, 131
InitCanon
 canonicalMsg.hh, 397
InitCanonMsg, 133
 ~InitCanonMsg, 135
 InitCanonMsg, 134
 InitCanonMsg, 134
interpretPlan
 CanonicalRobotCommand, 47
isKit_Finished
 Kit, 142
isSection
 IniFile, 131
key
 unix_shm_struct, 361
Kit, 135
 ~Kit, 138
 copy, 139
 dao, 142
 Explode, 139
 get, 139, 140
 getKitID, 141
 getdao, 140
 gethadByKit_LargeBoxWithKits, 140
 gethadByPart_Kit, 140
 gethasKit_DesignRef, 140
 gethasKit_Tray, 141
 getisKit_Finished, 141
 hadByKit_LargeBoxWithKits, 142
 hadByPart_Kit, 142
 hasKit_DesignRef, 142
 hasKit_Tray, 142
 isKit_Finished, 142
 Kit, 138
 KitID, 142
 set, 141
 setdao, 141
 sethadByKit_LargeBoxWithKits, 141
 sethadByPart_Kit, 142
 sethasKit_DesignRef, 142
 sethasKit_Tray, 142
 setisKit_Finished, 142
KitDesign, 143
 ~KitDesign, 146
 copy, 147
 dao, 150
 Explode, 147
 get, 147, 148
 getKitDesignID, 149
 getdao, 148
 gethadByKitDesign_Workstation, 148
 gethadByPartRefAndPose_KitDesign, 148
 gethasKitDesign_Id, 148
 gethasKitDesign_KitTraySkuRef, 149
 hadByKitDesign_Workstation, 150
 hadByPartRefAndPose_KitDesign, 150
 hasKitDesign_Id, 150
 hasKitDesign_KitTraySkuRef, 150
 KitDesign, 146
 KitDesignID, 150
 KitDesign, 146
 set, 149
 setdao, 149
 sethadByKitDesign_Workstation, 149
 sethadByPartRefAndPose_KitDesign, 149
 sethasKitDesign_Id, 149
 sethasKitDesign_KitTraySkuRef, 150
KitDesignID
 KitDesign, 150
KitID
 Kit, 142
KitTray, 165
 ~KitTray, 168
 copy, 169
 dao, 171
 Explode, 169

get, 169
getKitTrayID, 170
getdao, 170
gethadByKitTray_LargeBoxWithEmptyKitTrays, 170
gethasKit_Tray, 170
gethasKitTray_SerialNumber, 170
gethasKitTray_SkuRef, 170
hadByKitTray_LargeBoxWithEmptyKitTrays, 171
hasKit_Tray, 172
hasKitTray_SerialNumber, 172
hasKitTray_SkuRef, 172
KitTray, 168
KitTrayID, 172
KitTray, 168
set, 170
setdao, 171
sethadByKitTray_LargeBoxWithEmptyKitTrays, 171
sethasKit_Tray, 171
sethasKitTray_SerialNumber, 171
sethasKitTray_SkuRef, 171
KitTrayID
 KitTray, 172
KitTrayLocStruct, 172
 point, 174
 x_axis, 174
 z_axis, 174
KittingPDDLProblem, 150
 ~KittingPDDLProblem, 152
 findParam, 152
 findParamType, 153
 KittingPDDLProblem, 152
 KittingPDDLProblem, 152
 parsePDDLProblem, 153
KittingPlan, 155
KittingWorkstation, 156
 ~KittingWorkstation, 160
 copy, 160
 dao, 164
 Explode, 160
 get, 161
 getKittingWorkstationID, 162
 getdao, 161
 gethadByKitDesign_Workstation, 161
 gethadBySku_Workstation, 161
 gethasWorkstation_AngleUnit, 161
 gethasWorkstation_ChangingStation, 161
 gethasWorkstation_LengthUnit, 162
 gethasWorkstation_OtherObstacles, 162
 gethasWorkstation_Robot, 162
 gethasWorkstation_WeightUnit, 162
 gethasWorkstation_WorkTable, 162
 hadByKitDesign_Workstation, 164
 hadBySku_Workstation, 164
 hasWorkstation_AngleUnit, 164
 hasWorkstation_ChangingStation, 164
 hasWorkstation_LengthUnit, 164
 hasWorkstation_OtherObstacles, 164
 hasWorkstation_Robot, 164
 hasWorkstation_WeightUnit, 164
 hasWorkstation_WorkTable, 164
 KittingWorkstation, 160
 KittingWorkstationID, 164
 KittingWorkstation, 160
 set, 162
 setdao, 162
 sethadByKitDesign_Workstation, 163
 sethadBySku_Workstation, 163
 sethasWorkstation_AngleUnit, 163
 sethasWorkstation_ChangingStation, 163
 sethasWorkstation_LengthUnit, 163
 sethasWorkstation_OtherObstacles, 163
 sethasWorkstation_Robot, 163
 sethasWorkstation_WeightUnit, 163
 sethasWorkstation_WorkTable, 163
 KittingWorkstationID
 KittingWorkstation, 164
LargeBoxWithEmptyKitTrays, 174
 ~LargeBoxWithEmptyKitTrays, 177
 copy, 177
 dao, 180
 Explode, 178
 get, 178
 getLargeBoxWithEmptyKitTraysID, 179
 getdao, 179
 gethadByKitTray_LargeBoxWithEmptyKitTrays, 179
 gethasLargeBoxWithEmptyKitTrays_LargeContainer,
 179
 hadByKitTray_LargeBoxWithEmptyKitTrays, 180
 hasLargeBoxWithEmptyKitTrays_LargeContainer,
 180
 LargeBoxWithEmptyKitTrays, 177
 LargeBoxWithEmptyKitTraysID, 180
 LargeBoxWithEmptyKitTrays, 177
 set, 179
 setdao, 179
 sethadByKitTray_LargeBoxWithEmptyKitTrays, 180
 sethasLargeBoxWithEmptyKitTrays_LargeContainer,
 180
LargeBoxWithEmptyKitTraysID
 LargeBoxWithEmptyKitTrays, 180
LargeBoxWithKits, 180
 ~LargeBoxWithKits, 183
 copy, 184
 dao, 186
 Explode, 184
 get, 184
 getLargeBoxWithKitsID, 185

getdao, 185
 gethadByKit_LargeBoxWithKits, 185
 gethasLargeBoxWithKits_Capacity, 185
 gethasLargeBoxWithKits_KitDesignRef, 185
 gethasLargeBoxWithKits_LargeContainer, 185
 hadByKit_LargeBoxWithKits, 186
 hasLargeBoxWithKits_Capacity, 186
 hasLargeBoxWithKits_KitDesignRef, 187
 hasLargeBoxWithKits_LargeContainer, 187
 LargeBoxWithKits, 183
 LargeBoxWithKitsID, 187
 LargeBoxWithKits, 183
 set, 185
 setdao, 186
 sethadByKit_LargeBoxWithKits, 186
 sethasLargeBoxWithKits_Capacity, 186
 sethasLargeBoxWithKits_KitDesignRef, 186
 sethasLargeBoxWithKits_LargeContainer, 186
LargeBoxWithKitsID
 LargeBoxWithKits, 187
LargeContainer, 187
 ~LargeContainer, 190
 copy, 191
 dao, 193
 Explode, 191
 get, 191
 getLargeContainerID, 192
 getdao, 191
 gethasLargeBoxWithEmptyKitTrays_LargeContainer,
 191
 gethasLargeBoxWithKits_LargeContainer, 192
 gethasLargeContainer_SerialNumber, 192
 gethasLargeContainer_SkuRef, 192
 hasLargeBoxWithEmptyKitTrays_LargeContainer,
 193
 hasLargeBoxWithKits_LargeContainer, 193
 hasLargeContainer_SerialNumber, 193
 hasLargeContainer_SkuRef, 193
 LargeContainer, 190
 LargeContainerID, 193
 LargeContainer, 190
 set, 192
 setdao, 192
 sethasLargeBoxWithEmptyKitTrays_LargeContainer,
 192
 sethasLargeBoxWithKits_LargeContainer, 193
 sethasLargeContainer_SerialNumber, 193
 sethasLargeContainer_SkuRef, 193
LargeContainerID
 LargeContainer, 193

m_dwell
 CanonicalRobotCommand, 57
m_kit_tray

CanonicalRobotCommand, 57
m_safe_z
 CanonicalRobotCommand, 57
main
 main.cc, 493
main.cc
 main, 493
map
 DAO, 83
Message
 canonicalMsg.hh, 397
message
 MessageMsg, 196
MessageMsg, 194
 ~MessageMsg, 196
 message, 196
 MessageMsg, 196
 MessageMsg, 196
MoveSmoothlyTo
 canonicalMsg.hh, 397
MoveStraightTo
 canonicalMsg.hh, 397
MoveTo
 canonicalMsg.hh, 397
MoveSmoothlyToMsg, 196
 ~MoveSmoothlyToMsg, 199
 MoveSmoothlyToMsg, 199
 MoveSmoothlyToMsg, 199
 num, 199
MoveStraightToMsg, 199
 ~MoveStraightToMsg, 202
 MoveStraightToMsg, 201
 MoveStraightToMsg, 201
MoveToMsg, 202
 ~MoveToMsg, 204
 MoveToMsg, 203
 MoveToMsg, 203
msgID
 CanonicalHdr, 31
msgType
 CanonicalHdr, 31
MyController, 204
 ~MyController, 207
 MyController, 207
 MyController, 207
 processCloseGripper, 207
 processDwell, 207
 processEndCanon, 207
 processInitCanon, 207
 processMessage, 208
 processMoveSmoothlyTo, 208
 processMoveStraightTo, 208
 processMoveTo, 208
 processOpenGripper, 208

processSetAbsoluteAcceleration, 208
processSetAngleUnits, 208
processSetEndAngleTolerance, 208
processSetEndPointTolerance, 209
processSetIntermediatePointTolerance, 209
processSetLengthUnits, 209
processSetRelativeAcceleration, 209
processSetRelativeSpeed, 209

name
 DataThing, 91
 SolidObject, 353

nameDone
 DAO, 83

nextID
 CanonicalMsg, 35

num
 MoveSmoothlyToMsg, 199

open
 IniFile, 131

OpenGripper
 canonicalMsg.hh, 397

openFile
 FileOperator, 120

OpenGripperMsg, 209
 ~OpenGripperMsg, 212
 OpenGripperMsg, 211
 OpenGripperMsg, 211

operator=
 IniFile, 131

PDDL_DOMAIN
 Config.h, 480

PDDL_FOLDER
 Config.h, 481

PDDL_PROBLEM
 Config.h, 481

PLAN_FILE
 Config.h, 481

PLAN_FOLDER
 Config.h, 481

parsePDDLProblem
 KittingPDDLProblem, 153

Part, 212
 ~Part, 215
 copy, 215
 dao, 218
 Explode, 216
 get, 216
 getPartID, 217
 getdao, 216
 gethadByPart_Kit, 217
 gethadByPart_PartsTrayWithParts, 217
 gethasPart_SerialNumber, 217

gethasPart_SkuRef, 217
hadByPart_Kit, 218
hadByPart_PartsTrayWithParts, 218
hasPart_SerialNumber, 218
hasPart_SkuRef, 218
Part, 215
PartID, 218
set, 217
setdao, 217
sethadByPart_Kit, 218
sethadByPart_PartsTrayWithParts, 218
sethasPart_SerialNumber, 218
sethasPart_SkuRef, 218

PartID
 Part, 218

PartLocStruct, 219
 point, 221
 x_axis, 221
 z_axis, 221

PartRefAndPose, 221
 ~PartRefAndPose, 224
 copy, 224
 dao, 228
 Explode, 225
 get, 225
 getPartRefAndPoseID, 226
 getdao, 225
 gethadByPartRefAndPose_KitDesign, 226
 gethasPartRefAndPose_Point, 226
 gethasPartRefAndPose_Ref, 226
 gethasPartRefAndPose_XAxis, 226
 gethasPartRefAndPose_ZAxis, 226
 hadByPartRefAndPose_KitDesign, 228
 hasPartRefAndPose_Point, 228
 hasPartRefAndPose_Ref, 228
 hasPartRefAndPose_XAxis, 228
 hasPartRefAndPose_ZAxis, 228
 PartRefAndPose, 224
 PartRefAndPoseID, 228
 PartRefAndPose, 224
 set, 227
 setdao, 227
 sethadByPartRefAndPose_KitDesign, 227
 sethasPartRefAndPose_Point, 227
 sethasPartRefAndPose_Ref, 227
 sethasPartRefAndPose_XAxis, 227
 sethasPartRefAndPose_ZAxis, 227

PartRefAndPoseID
 PartRefAndPose, 228

PartsBin, 228
 ~PartsBin, 231
 copy, 231
 dao, 233
 Explode, 232

get, 232
 getPartsBinID, 233
 getdao, 232
 gethasBin_PartQuantity, 232
 gethasBin_PartSkuRef, 232
 hasBin_PartQuantity, 233
 hasBin_PartSkuRef, 234
 PartsBin, 231
 PartsBinID, 234
 PartsBin, 231
 set, 233
 setdao, 233
 sethasBin_PartQuantity, 233
 sethasBin_PartSkuRef, 233
PartsBinID
 PartsBin, 234
PartsTray, 234
 ~PartsTray, 237
 copy, 237
 dao, 240
 Explode, 238
 get, 238
 getPartsTrayID, 239
 getdao, 238
 gethasPartsTray_SerialNumber, 239
 gethasPartsTray_SkuRef, 239
 gethasPartsTrayWithParts_Tray, 239
 hasPartsTray_SerialNumber, 240
 hasPartsTray_SkuRef, 240
 hasPartsTrayWithParts_Tray, 240
 PartsTray, 237
 PartsTrayID, 240
 PartsTray, 237
 set, 239
 setdao, 239
 sethasPartsTray_SerialNumber, 239
 sethasPartsTray_SkuRef, 240
 sethasPartsTrayWithParts_Tray, 240
PartsTrayID
 PartsTray, 240
PartsTrayLocStruct, 240
 point, 242
 x_axis, 242
 z_axis, 242
PartsTrayWithParts, 242
 ~PartsTrayWithParts, 245
 copy, 245
 dao, 248
 Explode, 246
 get, 246
 getPartsTrayWithPartsID, 247
 getdao, 247
 gethadByPart_PartsTrayWithParts, 247
 gethasPartsTrayWithParts_Tray, 247
 hadByPart_PartsTrayWithParts, 248
 hasPartsTrayWithParts_Tray, 248
 PartsTrayWithParts, 245
 PartsTrayWithPartsID, 248
 PartsTrayWithParts, 245
 set, 247
 setdao, 248
 sethadByPart_PartsTrayWithParts, 248
 sethasPartsTrayWithParts_Tray, 248
PartsTrayWithPartsID
 PartsTrayWithParts, 248
path
 DAO, 83
pathmulti
 DAO, 83
percent
 SetRelativeAccelerationMsg, 332
 SetRelativeSpeedMsg, 335
PhysicalLocation, 249
 ~PhysicalLocation, 251
 copy, 251
 dao, 255
 Explode, 252
 get, 253
 getPhysicalLocationID, 254
 getdao, 253
 gethasPhysicalLocation_RefObject, 253
 gethasSolidObject_PrimaryLocation, 254
 gethasSolidObject_SecondaryLocation, 254
 hasPhysicalLocation_RefObject, 255
 hasSolidObject_PrimaryLocation, 255
 hasSolidObject_SecondaryLocation, 256
 PhysicalLocation, 251
 PhysicalLocationID, 256
 PhysicalLocation, 251
 set, 254
 setdao, 255
 sethasPhysicalLocation_RefObject, 255
 sethasSolidObject_PrimaryLocation, 255
 sethasSolidObject_SecondaryLocation, 255
PhysicalLocationID
 PhysicalLocation, 256
Point, 256
 ~Point, 259
 copy, 260
 dao, 264
 Explode, 260
 get, 260, 261
 getPointID, 262
 getdao, 261
 gethasBoxVolume_MaximumPoint, 261
 gethasBoxVolume_MinimumPoint, 261
 gethasPartRefAndPose_Point, 261
 gethasPoint_X, 261

gethasPoint_Y, 262
gethasPoint_Z, 262
gethasPoseLocation_Point, 262
hasBoxVolume_MaximumPoint, 264
hasBoxVolume_MinimumPoint, 264
hasPartRefAndPose_Point, 264
hasPoint_X, 264
hasPoint_Y, 264
hasPoint_Z, 264
hasPoseLocation_Point, 264
Point, 259
PointID, 264
set, 262, 263
setdao, 263
sethasBoxVolume_MaximumPoint, 263
sethasBoxVolume_MinimumPoint, 263
sethasPartRefAndPose_Point, 263
sethasPoint_X, 263
sethasPoint_Y, 263
sethasPoint_Z, 263
sethasPoseLocation_Point, 264
point
 KitTrayLocStruct, 174
 PartLocStruct, 221
 PartsTrayLocStruct, 242
PointID
 Point, 264
PoseLocation, 265
 ~PoseLocation, 268
 copy, 268
 dao, 272
 Explode, 269
 get, 269
 getPoseLocationID, 271
 getdao, 270
 gethasPoseLocation_Point, 270
 gethasPoseLocation_XAxis, 270
 gethasPoseLocation_ZAxis, 270
 hasPoseLocation_Point, 272
 hasPoseLocation_XAxis, 272
 hasPoseLocation_ZAxis, 272
 PoseLocation, 268
 PoseLocationID, 273
 PoseLocation, 268
 set, 271
 setdao, 272
 sethasPoseLocation_Point, 272
 sethasPoseLocation_XAxis, 272
 sethasPoseLocation_ZAxis, 272
PoseLocationID
 PoseLocation, 273
PoseLocationIn, 273
 ~PoseLocationIn, 276
 copy, 276
 dao, 278
 Explode, 276
 get, 277
 getPoseLocationInID, 277
 getdao, 277
 PoseLocationIn, 276
 set, 277
 setdao, 278
PoseLocationInID
 PoseLocationIn, 278
PoseLocationOn, 278
 ~PoseLocationOn, 281
 copy, 281
 dao, 283
 Explode, 281
 get, 282
 getPoseLocationOnID, 282
 getdao, 282
 PoseLocationOn, 281
 PoseLocationOnID, 283
 PoseLocationOn, 281
 set, 282
 setdao, 283
PoseLocationOnID
 PoseLocationOn, 283
PoseOnlyLocation, 283
 ~PoseOnlyLocation, 286
 copy, 286
 dao, 288
 Explode, 286
 get, 287
 getPoseOnlyLocationID, 287
 getdao, 287
 PoseOnlyLocation, 286
 PoseOnlyLocationID, 288
 PoseOnlyLocation, 286
 set, 287
 setdao, 288
PoseOnlyLocationID
 PoseOnlyLocation, 288
print_closegripper
 CanonicalRobotCommand, 48
print_dwell
 CanonicalRobotCommand, 49
print_endcannon
 CanonicalRobotCommand, 49
print_initcannon
 CanonicalRobotCommand, 50
print_moveto
 CanonicalRobotCommand, 51
print.opengripper
 CanonicalRobotCommand, 52

processCloseGripper
 Controller, 67
 MyController, 207
 processDwell
 Controller, 68
 MyController, 207
 processEndCanon
 Controller, 68
 MyController, 207
 processInitCanon
 Controller, 68
 MyController, 207
 processMessage
 Controller, 69
 MyController, 208
 processMoveSmoothlyTo
 Controller, 69
 MyController, 208
 processMoveStraightTo
 Controller, 69
 MyController, 208
 processMoveTo
 Controller, 70
 MyController, 208
 processOpenGripper
 Controller, 70
 MyController, 208
 processSetAbsoluteAcceleration
 Controller, 70
 MyController, 208
 processSetAngleUnits
 Controller, 71
 MyController, 208
 processSetEndAngleTolerance
 Controller, 71
 MyController, 208
 processSetEndPointTolerance
 Controller, 71
 MyController, 209
 processSetIntermediatePointTolerance
 Controller, 72
 MyController, 209
 processSetLengthUnits
 Controller, 72
 MyController, 209
 processSetRelativeAcceleration
 Controller, 72
 MyController, 209
 processSetRelativeSpeed
 Controller, 73
 MyController, 209
 pthread_task_code
 ulapi.cpp, 405
 put_kit
 CanonicalRobotCommand, 53
 put_kit_tray
 CanonicalRobotCommand, 53
 put_part
 CanonicalRobotCommand, 53
 queue_length
 Controller, 74
 queueCloseGripper
 Controller, 73
 queueDwell
 Controller, 73
 queueMsg
 Controller, 73
 ROBOT_COMMANDS_FILE
 Config.h, 481
 readMap
 FileOperator, 120
 readMapOfVector
 FileOperator, 120
 readVector
 FileOperator, 120
 readVectorOfVector
 FileOperator, 121
 reason
 EndCanonMsg, 97
 RelativeLocation, 288
 ~RelativeLocation, 291
 copy, 291
 dao, 294
 Explode, 292
 get, 292
 getRelativeLocationID, 293
 getdao, 292
 gethasRelativeLocation_Description, 292
 hasRelativeLocation_Description, 294
 RelativeLocation, 291
 RelativeLocationID, 294
 RelativeLocation, 291
 set, 293
 setdao, 293
 sethasRelativeLocation_Description, 293
 RelativeLocationID
 RelativeLocation, 294
 RelativeLocationIn, 294
 ~RelativeLocationIn, 297
 copy, 297
 dao, 299
 Explode, 297
 get, 298
 getRelativeLocationInID, 298
 getdao, 298
 RelativeLocationIn, 297
 RelativeLocationInID, 299

RelativeLocationIn, 297
set, 298
setdao, 299
RelativeLocationInID
 RelativeLocationIn, 299
RelativeLocationOn, 299
 ~RelativeLocationOn, 302
 copy, 302
 dao, 304
 Explode, 302
 get, 303
 getRelativeLocationOnID, 303
 getdao, 303
 RelativeLocationOn, 302
 RelativeLocationOnID, 304
 RelativeLocationOn, 302
 set, 303
 setdao, 304
RelativeLocationOnID
 RelativeLocationOn, 304
remove_eff
 CanonicalRobotCommand, 54
removeDuplicates
 FileOperator, 121
removeParentheses
 FileOperator, 121
rest
 INIFILE_ENTRY, 132
Robot, 304
 ~Robot, 307
 copy, 308
 dao, 311
 Explode, 308
 get, 308
 getRobotID, 309
 getdao, 309
 gethadByEndEffector_Robot, 309
 gethasRobot_Description, 309
 gethasRobot_Id, 309
 gethasRobot_MaximumLoadWeight, 309
 gethasRobot_WorkVolume, 309
 gethasWorkstation_Robot, 309
 hadByEndEffector_Robot, 311
 hasRobot_Description, 311
 hasRobot_Id, 311
 hasRobot_MaximumLoadWeight, 311
 hasRobot_WorkVolume, 311
 hasWorkstation_Robot, 311
 Robot, 307
 RobotID, 311
 set, 309, 310
 setdao, 310
 sethadByEndEffector_Robot, 310
 sethasRobot_Description, 310
sethasRobot_Id, 310
sethasRobot_MaximumLoadWeight, 310
sethasRobot_WorkVolume, 310
sethasWorkstation_Robot, 310
section
 IniFile, 131
set
 BoxVolume, 22
 BoxyObject, 29
 DAO, 81
 DataThing, 89, 90
 EndEffector, 102, 103
 EndEffectorChangingStation, 110
 EndEffectorHolder, 116
 GripperEffector, 127
 Kit, 141
 KitDesign, 149
 KittingWorkstation, 162
 KitTray, 170
 LargeBoxWithEmptyKitTrays, 179
 LargeBoxWithKits, 185
 LargeContainer, 192
 Part, 217
 PartRefAndPose, 227
 PartsBin, 233
 PartsTray, 239
 PartsTrayWithParts, 247
 PhysicalLocation, 254
 Point, 262, 263
 PoseLocation, 271
 PoseLocationIn, 277
 PoseLocationOn, 282
 PoseOnlyLocation, 287
 RelativeLocation, 293
 RelativeLocationIn, 298
 RelativeLocationOn, 303
 Robot, 309, 310
 ShapeDesign, 340
 SolidObject, 350, 351
 StockKeepingUnit, 358, 359
 VacuumEffect, 367
 VacuumEffectMultiCup, 373
 VacuumEffectSingleCup, 378
 Vector, 385
 WorkTable, 392
SetAbsoluteAcceleration
 canonicalMsg.hh, 397
SetAngleUnits
 canonicalMsg.hh, 397
SetEndAngleTolerance
 canonicalMsg.hh, 397

SetEndPointTolerance
 canonicalMsg.hh, 397

SetIntermediatePointTolerance
 canonicalMsg.hh, 398

SetLengthUnits
 canonicalMsg.hh, 398

SetRelativeAcceleration
 canonicalMsg.hh, 398

SetRelativeSpeed
 canonicalMsg.hh, 398

SetAbsoluteAccelerationMsg, 312
 ~SetAbsoluteAccelerationMsg, 314
 acceleration, 314

SetAbsoluteAccelerationMsg, 314

SetAbsoluteAccelerationMsg, 314

SetAngleUnitsMsg, 314
 ~SetAngleUnitsMsg, 317

SetAngleUnitsMsg, 317

SetAngleUnitsMsg, 317

unitName, 317

setCon
 Connection, 62

setDriver
 Connection, 63

SetEndAngleToleranceMsg, 317
 ~SetEndAngleToleranceMsg, 320

SetEndAngleToleranceMsg, 320

SetEndAngleToleranceMsg, 320

tolerance, 320

SetEndPointToleranceMsg, 320
 ~SetEndPointToleranceMsg, 323

SetEndPointToleranceMsg, 323

SetEndPointToleranceMsg, 323

tolerance, 323

setHeader
 CanonicalMsg, 34

SetIntermediatePointToleranceMsg, 323
 ~SetIntermediatePointToleranceMsg, 326

SetIntermediatePointToleranceMsg, 326

SetIntermediatePointToleranceMsg, 326

tolerance, 326

SetLengthUnitsMsg, 326
 ~SetLengthUnitsMsg, 329

SetLengthUnitsMsg, 329

SetLengthUnitsMsg, 329

unitName, 329

SetRelativeAccelerationMsg, 329
 ~SetRelativeAccelerationMsg, 332

percent, 332

SetRelativeAccelerationMsg, 332

SetRelativeAccelerationMsg, 332

SetRelativeSpeedMsg, 332
 ~SetRelativeSpeedMsg, 335

percent, 335

SetRelativeSpeedMsg, 335
 SetRelativeSpeedMsg, 335

setSqlQueries
 DAO, 83

setclassName
 DAO, 82

setconnection
 DAO, 82

setdao
 BoxVolume, 23

BoxyObject, 30

DataThing, 91

EndEffector, 103

EndEffectorChangingStation, 110

EndEffectorHolder, 116

GripperEffector, 128

Kit, 141

KitDesign, 149

KittingWorkstation, 162

KitTray, 171

LargeBoxWithEmptyKitTrays, 179

LargeBoxWithKits, 186

LargeContainer, 192

Part, 217

PartRefAndPose, 227

PartsBin, 233

PartsTray, 239

PartsTrayWithParts, 248

PhysicalLocation, 255

Point, 263

PoseLocation, 272

PoseLocationIn, 278

PoseLocationOn, 283

PoseOnlyLocation, 288

RelativeLocation, 293

RelativeLocationIn, 299

RelativeLocationOn, 304

Robot, 310

ShapeDesign, 341

SolidObject, 352

StockKeepingUnit, 359

VacuumEffector, 367

VacuumEffectorMultiCup, 373

VacuumEffectorSingleCup, 379

Vector, 386

WorkTable, 393

sethadByEndEffector_Robot
 EndEffector, 103

Robot, 310

sethadByEndEffectorHolder_ChangingStation
 EndEffectorChangingStation, 111

EndEffectorHolder, 116

sethadByKit_LargeBoxWithKits
 Kit, 141

LargeBoxWithKits, 186
sethadByKitDesign_Workstation
 KitDesign, 149
 KittingWorkstation, 163
sethadByKitTray_LargeBoxWithEmptyKitTrays
 KitTray, 171
 LargeBoxWithEmptyKitTrays, 180
sethadByPart_Kit
 Kit, 142
 Part, 218
sethadByPart_PartsTrayWithParts
 Part, 218
 PartsTrayWithParts, 248
sethadByPartRefAndPose_KitDesign
 KitDesign, 149
 PartRefAndPose, 227
sethadBySku_Workstation
 KittingWorkstation, 163
 StockKeepingUnit, 359
sethadBySolidObject_WorkTable
 SolidObject, 352
 WorkTable, 393
sethasBin_PartQuantity
 PartsBin, 233
sethasBin_PartSkuRef
 PartsBin, 233
sethasBox_Height
 BoxyObject, 30
sethasBox_Length
 BoxyObject, 30
sethasBox_Width
 BoxyObject, 30
sethasBoxVolume_MaximumPoint
 BoxVolume, 23
 Point, 263
sethasBoxVolume_MinimumPoint
 BoxVolume, 23
 Point, 263
sethasEffector_MaximumLoadWeight
 EndEffector, 104
sethasEndEffector_Description
 EndEffector, 104
sethasEndEffector_Id
 EndEffector, 104
sethasEndEffector_Weight
 EndEffector, 104
sethasEndEffectorHolder_EndEffector
 EndEffector, 104
 EndEffectorHolder, 116
sethasKit_DesignRef
 Kit, 142
sethasKit_Tray
 Kit, 142
 KitTray, 171
sethasKitDesign_Id
 KitDesign, 149
sethasKitDesign_KitTraySkuRef
 KitDesign, 150
sethasKitTray_SerialNumber
 KitTray, 171
sethasKitTray_SkuRef
 KitTray, 171
sethasLargeBoxWithEmptyKitTrays_LargeContainer
 LargeBoxWithEmptyKitTrays, 180
 LargeContainer, 192
sethasLargeBoxWithKits_Capacity
 LargeBoxWithKits, 186
sethasLargeBoxWithKits_KitDesignRef
 LargeBoxWithKits, 186
sethasLargeBoxWithKits_LargeContainer
 LargeBoxWithKits, 186
 LargeContainer, 193
sethasLargeContainer_SerialNumber
 LargeContainer, 193
sethasLargeContainer_SkuRef
 LargeContainer, 193
sethasMultiCup_ArrayNumber
 VacuumEffectorMultiCup, 373
sethasMultiCup_ArrayRadius
 VacuumEffectorMultiCup, 373
sethasPart_SerialNumber
 Part, 218
sethasPart_SkuRef
 Part, 218
sethasPartRefAndPose_Point
 PartRefAndPose, 227
 Point, 263
sethasPartRefAndPose_Ref
 PartRefAndPose, 227
sethasPartRefAndPose_XAxis
 PartRefAndPose, 227
 Vector, 386
sethasPartRefAndPose_ZAxis
 PartRefAndPose, 227
 Vector, 386
sethasPartsTray_SerialNumber
 PartsTray, 239
sethasPartsTray_SkuRef
 PartsTray, 240
sethasPartsTrayWithParts_Tray
 PartsTray, 240
 PartsTrayWithParts, 248
sethasPhysicalLocation_RefObject
 PhysicalLocation, 255
 SolidObject, 352
sethasPoint_X
 Point, 263
sethasPoint_Y

Point, 263
 sethasPoint_Z
 Point, 263
 sethasPoseLocation_Point
 Point, 264
 PoseLocation, 272
 sethasPoseLocation_XAxis
 PoseLocation, 272
 Vector, 386
 sethasPoseLocation_ZAxis
 PoseLocation, 272
 Vector, 386
 sethasRelativeLocation_Description
 RelativeLocation, 293
 sethasRobot_Description
 Robot, 310
 sethasRobot_Id
 Robot, 310
 sethasRobot_MaximumLoadWeight
 Robot, 310
 sethasRobot_WorkVolume
 BoxVolume, 23
 Robot, 310
 sethasShapeDesign_Description
 ShapeDesign, 341
 sethasSku_Description
 StockKeepingUnit, 359
 sethasSku_EndEffectorRefs
 StockKeepingUnit, 359
 sethasSku_Id
 StockKeepingUnit, 359
 sethasSku_Shape
 ShapeDesign, 341
 StockKeepingUnit, 359
 sethasSku_Weight
 StockKeepingUnit, 359
 sethasSolidObject_PrimaryLocation
 PhysicalLocation, 255
 SolidObject, 352
 sethasSolidObject_SecondaryLocation
 PhysicalLocation, 255
 SolidObject, 352
 sethasVacuumEffector_CupDiameter
 VacuumEffector, 368
 sethasVacuumEffector_Length
 VacuumEffector, 368
 sethasVector_I
 Vector, 386
 sethasVector_J
 Vector, 386
 sethasVector_K
 Vector, 386
 sethasWorkstation_AngleUnit
 KittingWorkstation, 163
 sethasWorkstation_ChangingStation
 EndEffectorChangingStation, 111
 KittingWorkstation, 163
 sethasWorkstation_LengthUnit
 KittingWorkstation, 163
 sethasWorkstation_OtherObstacles
 BoxVolume, 23
 KittingWorkstation, 163
 sethasWorkstation_Robot
 KittingWorkstation, 163
 Robot, 310
 sethasWorkstation_WeightUnit
 KittingWorkstation, 163
 sethasWorkstation_WorkTable
 KittingWorkstation, 163
 WorkTable, 393
 setisKit_Finished
 Kit, 142
 ShapeDesign, 335
 ~ShapeDesign, 338
 copy, 338
 dao, 341
 Explode, 339
 get, 339
 getShapeDesignID, 340
 getdao, 340
 gethasShapeDesign_Description, 340
 gethasSku_Shape, 340
 hasShapeDesign_Description, 341
 hasSku_Shape, 341
 set, 340
 setdao, 341
 sethasShapeDesign_Description, 341
 sethasSku_Shape, 341
 ShapeDesign, 338
 ShapeDesignID, 341
 ShapeDesign, 338
 ShapeDesignID
 ShapeDesign, 341
 size
 unix_shm_struct, 361
 skipwhite
 IniFile.cpp, 483
 SolidObject, 342
 ~SolidObject, 345
 copy, 345
 dao, 352
 Explode, 346
 get, 347
 getSolidObjectID, 349
 getdao, 348
 gethadBySolidObject_WorkTable, 348
 gethasPhysicalLocation_RefObject, 348
 gethasSolidObject_PrimaryLocation, 348

gethasSolidObject_SecondaryLocation, 348
getname, 348
hadBySolidObject_WorkTable, 352
hasPhysicalLocation_RefObject, 353
hasSolidObject_PrimaryLocation, 353
hasSolidObject_SecondaryLocation, 353
name, 353
set, 350, 351
setdao, 352
sethadBySolidObject_WorkTable, 352
sethasPhysicalLocation_RefObject, 352
sethasSolidObject_PrimaryLocation, 352
sethasSolidObject_SecondaryLocation, 352
SolidObject, 344
SolidObjectID, 353
SolidObject, 344
SolidObjectID
 SolidObject, 353
splitString
 FileOperator, 121
StockKeepingUnit, 353
 ~StockKeepingUnit, 356
 copy, 357
 dao, 360
 Explode, 357
 get, 357
 getStockKeepingUnitID, 358
 getdao, 358
 gethadBySku_Workstation, 358
 gethasSku_Description, 358
 gethasSku_EndEffectorRefs, 358
 gethasSku_Id, 358
 gethasSku_Shape, 358
 gethasSku_Weight, 358
 hadBySku_Workstation, 360
 hasSku_Description, 360
 hasSku_EndEffectorRefs, 360
 hasSku_Id, 360
 hasSku_Shape, 360
 hasSku_Weight, 360
 set, 358, 359
 setdao, 359
 sethadBySku_Workstation, 359
 sethasSku_Description, 359
 sethasSku_EndEffectorRefs, 359
 sethasSku_Id, 359
 sethasSku_Shape, 359
 sethasSku_Weight, 359
StockKeepingUnit, 356
StockKeepingUnitID, 360
 StockKeepingUnit, 356
StockKeepingUnitID
 StockKeepingUnit, 360
stripSpace
 FileOperator, 122
tag
 INIFILE_ENTRY, 132
take_kit
 CanonicalRobotCommand, 55
take_kit_tray
 CanonicalRobotCommand, 55
take_part
 CanonicalRobotCommand, 56
time
 CanonicalHdr, 32
 DwellMsg, 94
tolerance
 SetEndAngleToleranceMsg, 320
 SetEndPointToleranceMsg, 323
 SetIntermediatePointToleranceMsg, 326
UL_USE_DEFAULT
 ulapi.hh, 415
UL_USE_RTAI
 ulapi.hh, 415
UL_USE_UNIX
 ulapi.hh, 415
ULAPI_BAD_ARGS
 ulapi.hh, 415
ULAPI_DEBUG_ALL
 ulapi.hh, 416
ULAPI_DEBUG_ERROR
 ulapi.hh, 416
ULAPI_DEBUG_WARN
 ulapi.hh, 416
ULAPI_ERROR
 ulapi.hh, 415
ULAPI_IMPL_ERROR
 ulapi.hh, 415
ULAPI_OK
 ulapi.hh, 415
ULAPI_STDERR
 ulapi.hh, 416
ULAPI_STDIN
 ulapi.hh, 416
ULAPI_STDOUT
 ulapi.hh, 416
ULAPI_DECL_SHARED
 ulapi.hh, 414
ulapi.hh
 UL_USE_DEFAULT, 415
 UL_USE_RTAI, 415
 UL_USE_UNIX, 415
 ULAPI_BAD_ARGS, 415
 ULAPI_DEBUG_ALL, 416
 ULAPI_DEBUG_ERROR, 416
 ULAPI_DEBUG_WARN, 416
 ULAPI_ERROR, 415

ULAPIO_IMPL_ERROR, 415
 ULAPIO_OK, 415
 ULAPIO_STDERR, 416
 ULAPIO_STDIN, 416
 ULAPIO_STDOUT, 416
 ulapi.cpp
 pthread_task_code, 405
 ulapi_exit, 405
 ulapi_init, 405
 ulapi_mutex_delete, 405
 ulapi_mutex_give, 406
 ulapi_mutex_new, 406
 ulapi_mutex_take, 406
 ulapi_prio_highest, 406
 ulapi_prio_lowest, 406
 ulapi_prio_next_higher, 407
 ulapi_prio_next_lower, 407
 ulapi_socket_close, 407
 ulapi_socket_get_client_id, 407
 ulapi_socket_read, 408
 ulapi_socket_write, 408
 ulapi_task_delete, 408
 ulapi_task_init, 408
 ulapi_task_new, 408
 ulapi_task_pause, 408
 ulapi_task_resume, 408
 ulapi_task_set_period, 408
 ulapi_task_start, 408
 ulapi_task_stop, 408
 ulapi_time, 409
 unix_ulapi_exit, 409
 unix_ulapi_fifo_delete, 409
 unix_ulapi_fifo_new, 409
 unix_ulapi_fifo_read, 409
 unix_ulapi_fifo_write, 409
 unix_ulapi_shm_addr, 409
 unix_ulapi_shm_delete, 409
 unix_ulapi_shm_new, 409
 ulapi.hh
 DIGITS_IN, 414
 ULAPI_DECL_SHARED, 414
 ulapi_address_to_hostname, 416
 ulapi_cond_broadcast, 416
 ulapi_cond_delete, 416
 ulapi_cond_new, 416
 ulapi_cond_signal, 416
 ulapi_cond_wait, 416
 ulapi_exit, 416
 ulapi_fd_close, 416
 ulapi_fd_delete, 417
 ulapi_fd_new, 417
 ulapi_fd_open, 417
 ulapi_fd_read, 417
 ulapi_fd_set_blocking, 417
 ulapi_fd_set_nonblocking, 417
 ulapi_fd_write, 417
 ulapi_fifo_delete, 417
 ulapi_fifo_new, 417
 ulapi_fifo_read, 417
 ulapi_fifo_write, 417
 ulapi_flag, 414
 ulapi_free_argv, 417
 ulapi_get_host_address, 417
 ulapi_hostname_to_address, 417
 ulapi_id, 414
 ulapi_init, 417
 ulapi_integer, 414
 ulapi_mutex_delete, 418
 ulapi_mutex_give, 418
 ulapi_mutex_new, 418
 ulapi_mutex_take, 418
 ulapi_print, 414
 ulapi_prio, 415
 ulapi_prio_highest, 418
 ulapi_prio_lowest, 418
 ulapi_prio_next_higher, 419
 ulapi_prio_next_lower, 419
 ulapi_real, 415
 ulapi_result, 415
 ulapi_self_set_period, 419
 ulapi_sem_delete, 419
 ulapi_sem_give, 420
 ulapi_sem_new, 420
 ulapi_sem_take, 420
 ulapi_serial_close, 420
 ulapi_serial_delete, 420
 ulapi_serial_new, 420
 ulapi_serial_open, 420
 ulapi_serial_read, 420
 ulapi_serial_set_blocking, 420
 ulapi_serial_set_nonblocking, 420
 ulapi_serial_write, 420
 ulapi_set_debug, 420
 ulapi_shm_addr, 420
 ulapi_shm_delete, 420
 ulapi_shm_new, 421
 ulapi_sleep, 421
 ulapi_snprintf, 414
 ulapi_socket_broadcast, 421
 ulapi_socket_close, 421
 ulapi_socket_get_broadcastee_id, 421
 ulapi_socket_get_broadcaster_id, 421
 ulapi_socket_get_client_id, 421
 ulapi_socket_get_connection_id, 421
 ulapi_socket_get_server_id, 421
 ulapi_socket_read, 421
 ulapi_socket_set_blocking, 422
 ulapi_socket_set_nonblocking, 422

ulapi_socket_write, 422
ulapi_sscanf, 414
ulapi_std_open, 422
ulapi_stdio, 416
ulapi_strcasecmp, 414
ulapi_strdup, 414
ulapi_strncpy, 414
ulapi_system, 422
ulapi_task_code, 415
ulapi_task_delete, 422
ulapi_task_exit, 422
ulapi_task_id, 422
ulapi_task_init, 422
ulapi_task_join, 422
ulapi_task_new, 422
ulapi_task_pause, 423
ulapi_task_resume, 423
ulapi_task_set_period, 423
ulapi_task_start, 423
ulapi_task_stop, 423
ulapi_time, 423
ulapi_to_argv, 423
ulapi_wait, 423
ulapi_address_to_hostname
 ulapi.hh, 416
ulapi_cond_broadcast
 ulapi.hh, 416
ulapi_cond_delete
 ulapi.hh, 416
ulapi_cond_new
 ulapi.hh, 416
ulapi_cond_signal
 ulapi.hh, 416
ulapi_cond_wait
 ulapi.hh, 416
ulapi_exit
 ulapi.cpp, 405
 ulapi.hh, 416
ulapi_fd_close
 ulapi.hh, 416
ulapi_fd_delete
 ulapi.hh, 417
ulapi_fd_new
 ulapi.hh, 417
ulapi_fd_open
 ulapi.hh, 417
ulapi_fd_read
 ulapi.hh, 417
ulapi_fd_set_blocking
 ulapi.hh, 417
ulapi_fd_set_nonblocking
 ulapi.hh, 417
ulapi_fd_write
 ulapi.hh, 417

ulapi_fifo_delete
 ulapi.hh, 417
ulapi_fifo_new
 ulapi.hh, 417
ulapi_fifo_read
 ulapi.hh, 417
ulapi_fifo_write
 ulapi.hh, 417
ulapi_flag
 ulapi.hh, 414
ulapi_free_argv
 ulapi.hh, 417
ulapi_get_host_address
 ulapi.hh, 417
ulapi_hostname_to_address
 ulapi.hh, 417
ulapi_id
 ulapi.hh, 414
ulapi_init
 ulapi.cpp, 405
 ulapi.hh, 417
ulapi_integer
 ulapi.hh, 414
ulapi_mutex_delete
 ulapi.cpp, 405
 ulapi.hh, 418
ulapi_mutex_give
 ulapi.cpp, 406
 ulapi.hh, 418
ulapi_mutex_new
 ulapi.cpp, 406
 ulapi.hh, 418
ulapi_mutex_take
 ulapi.cpp, 406
 ulapi.hh, 418
ulapi_print
 ulapi.hh, 414
ulapi_prio
 ulapi.hh, 415
ulapi_prio_highest
 ulapi.cpp, 406
 ulapi.hh, 418
ulapi_prio_lowest
 ulapi.cpp, 406
 ulapi.hh, 418
ulapi_prio_next_higher
 ulapi.cpp, 407
 ulapi.hh, 419
ulapi_prio_next_lower
 ulapi.cpp, 407
 ulapi.hh, 419
ulapi_real
 ulapi.hh, 415
ulapi_result

ulapi.hh, 415
ulapi_self_set_period
 ulapi.hh, 419
ulapi_sem_delete
 ulapi.hh, 419
ulapi_sem_give
 ulapi.hh, 420
ulapi_sem_new
 ulapi.hh, 420
ulapi_sem_take
 ulapi.hh, 420
ulapi_serial_close
 ulapi.hh, 420
ulapi_serial_delete
 ulapi.hh, 420
ulapi_serial_new
 ulapi.hh, 420
ulapi_serial_open
 ulapi.hh, 420
ulapi_serial_read
 ulapi.hh, 420
ulapi_serial_set_blocking
 ulapi.hh, 420
ulapi_serial_set_nonblocking
 ulapi.hh, 420
ulapi_serial_write
 ulapi.hh, 420
ulapi_set_debug
 ulapi.hh, 420
ulapi_shm_addr
 ulapi.hh, 420
ulapi_shm_delete
 ulapi.hh, 420
ulapi_shm_new
 ulapi.hh, 421
ulapi_sleep
 ulapi.hh, 421
ulapi_snprintf
 ulapi.hh, 414
ulapi_socket_broadcast
 ulapi.hh, 421
ulapi_socket_close
 ulapi.cpp, 407
 ulapi.hh, 421
ulapi_socket_get_broadcastee_id
 ulapi.hh, 421
ulapi_socket_get_broadcaster_id
 ulapi.hh, 421
ulapi_socket_get_client_id
 ulapi.cpp, 407
 ulapi.hh, 421
ulapi_socket_get_connection_id
 ulapi.hh, 421
ulapi_socket_get_server_id

ulapi.hh, 421
ulapi_socket_read
 ulapi.cpp, 408
 ulapi.hh, 421
ulapi_socket_set_blocking
 ulapi.hh, 422
ulapi_socket_set_nonblocking
 ulapi.hh, 422
ulapi_socket_write
 ulapi.cpp, 408
 ulapi.hh, 422
ulapi_sscanf
 ulapi.hh, 414
ulapi_std_open
 ulapi.hh, 422
ulapi_stdio
 ulapi.hh, 416
ulapi_strcasecmp
 ulapi.hh, 414
ulapi_strdup
 ulapi.hh, 414
ulapi_strncpy
 ulapi.hh, 414
ulapi_system
 ulapi.hh, 422
ulapi_task_code
 ulapi.hh, 415
ulapi_task_delete
 ulapi.cpp, 408
 ulapi.hh, 422
ulapi_task_exit
 ulapi.hh, 422
ulapi_task_id
 ulapi.hh, 422
ulapi_task_init
 ulapi.cpp, 408
 ulapi.hh, 422
ulapi_task_join
 ulapi.hh, 422
ulapi_task_new
 ulapi.cpp, 408
 ulapi.hh, 422
ulapi_task_pause
 ulapi.cpp, 408
 ulapi.hh, 423
ulapi_task_resume
 ulapi.cpp, 408
 ulapi.hh, 423
ulapi_task_set_period
 ulapi.cpp, 408
 ulapi.hh, 423
ulapi_task_start
 ulapi.cpp, 408
 ulapi.hh, 423

ulapi_task_stop
 ulapi.cpp, 408
 ulapi.hh, 423

ulapi_time
 ulapi.cpp, 409
 ulapi.hh, 423

ulapi_to_argv
 ulapi.hh, 423

ulapi_wait
 ulapi.hh, 423

unitName
 SetAngleUnitsMsg, 317
 SetLengthUnitsMsg, 329

unix_shm_struct, 361
 addr, 361
 id, 361
 key, 361
 size, 361

unix_ulapi_exit
 ulapi.cpp, 409

unix_ulapi_fifo_delete
 ulapi.cpp, 409

unix_ulapi_fifo_new
 ulapi.cpp, 409

unix_ulapi_fifo_read
 ulapi.cpp, 409

unix_ulapi_fifo_write
 ulapi.cpp, 409

unix_ulapi_shm_addr
 ulapi.cpp, 409

unix_ulapi_shm_delete
 ulapi.cpp, 409

unix_ulapi_shm_new
 ulapi.cpp, 409

updateSqlQueries
 DAO, 84

User-Level API, 15

VacuumEffector, 362
 ~VacuumEffector, 365
 copy, 365
 dao, 368
 Explode, 366
 get, 366
 getVacuumEffectorID, 367
 getdao, 366
 gethasVacuumEffector_CupDiameter, 366
 gethasVacuumEffector_Length, 367
 hasVacuumEffector_CupDiameter, 368
 hasVacuumEffector_Length, 368
 set, 367
 setdao, 367
 sethasVacuumEffector_CupDiameter, 368
 sethasVacuumEffector_Length, 368

VacuumEffector, 365
 VacuumEffectorID, 368
 VacuumEffector, 365

VacuumEffectorID
 VacuumEffectorMultiCup, 368
 ~VacuumEffectorMultiCup, 371
 copy, 371
 dao, 373
 Explode, 372
 get, 372
 getVacuumEffectorMultiCupID, 373
 getdao, 372
 gethasMultiCup_ArrayNumber, 372
 gethasMultiCup_ArrayRadius, 372
 hasMultiCup_ArrayNumber, 373
 hasMultiCup_ArrayRadius, 374
 set, 373
 setdao, 373
 sethasMultiCup_ArrayNumber, 373
 sethasMultiCup_ArrayRadius, 373
 VacuumEffectorMultiCup, 371
 VacuumEffectorMultiCupID, 374
 VacuumEffectorMultiCup, 371

VacuumEffectorMultiCupID
 VacuumEffectorMultiCup, 374

VacuumEffectorSingleCup, 374
 ~VacuumEffectorSingleCup, 377
 copy, 377
 dao, 379
 Explode, 377
 get, 378
 getVacuumEffectorSingleCupID, 378
 getdao, 378
 set, 378
 setdao, 379
 VacuumEffectorSingleCup, 377
 VacuumEffectorSingleCupID, 379
 VacuumEffectorSingleCup, 377

VacuumEffectorSingleCupID
 VacuumEffectorSingleCup, 379

valid
 IniFile, 131

Vector, 379
 ~Vector, 382
 copy, 383
 dao, 387
 Explode, 383
 get, 383, 384
 getVectorID, 385
 getdao, 384
 gethasPartRefAndPose_XAxis, 384
 gethasPartRefAndPose_ZAxis, 384
 gethasPoseLocation_XAxis, 384

gethasPoseLocation_ZAxis, 384
gethasVector_I, 385
gethasVector_J, 385
gethasVector_K, 385
hasPartRefAndPose_XAxis, 387
hasPartRefAndPose_ZAxis, 387
hasPoseLocation_XAxis, 387
hasPoseLocation_ZAxis, 387
hasVector_I, 387
hasVector_J, 387
hasVector_K, 387
set, 385
setdao, 386
sethasPartRefAndPose_XAxis, 386
sethasPartRefAndPose_ZAxis, 386
sethasPoseLocation_XAxis, 386
sethasPoseLocation_ZAxis, 386
sethasVector_I, 386
sethasVector_J, 386
sethasVector_K, 386
Vector, 382
VectorID, 387
VectorID
Vector, 387

WorkTable, 388
~WorkTable, 390
copy, 390
dao, 393
Explode, 391
get, 391
getWorkTableID, 392
getdao, 392
gethadBySolidObject_WorkTable, 392
gethasWorkstation_WorkTable, 392
hadBySolidObject_WorkTable, 393
hasWorkstation_WorkTable, 393
set, 392
setdao, 393
sethadBySolidObject_WorkTable, 393
sethasWorkstation_WorkTable, 393
WorkTable, 390
WorkTableID, 393
WorkTable, 390
WorkTableID
WorkTable, 393
writeData
FileOperator, 122

x_axis
KitTrayLocStruct, 174
PartLocStruct, 221
PartsTrayLocStruct, 242

z_axis