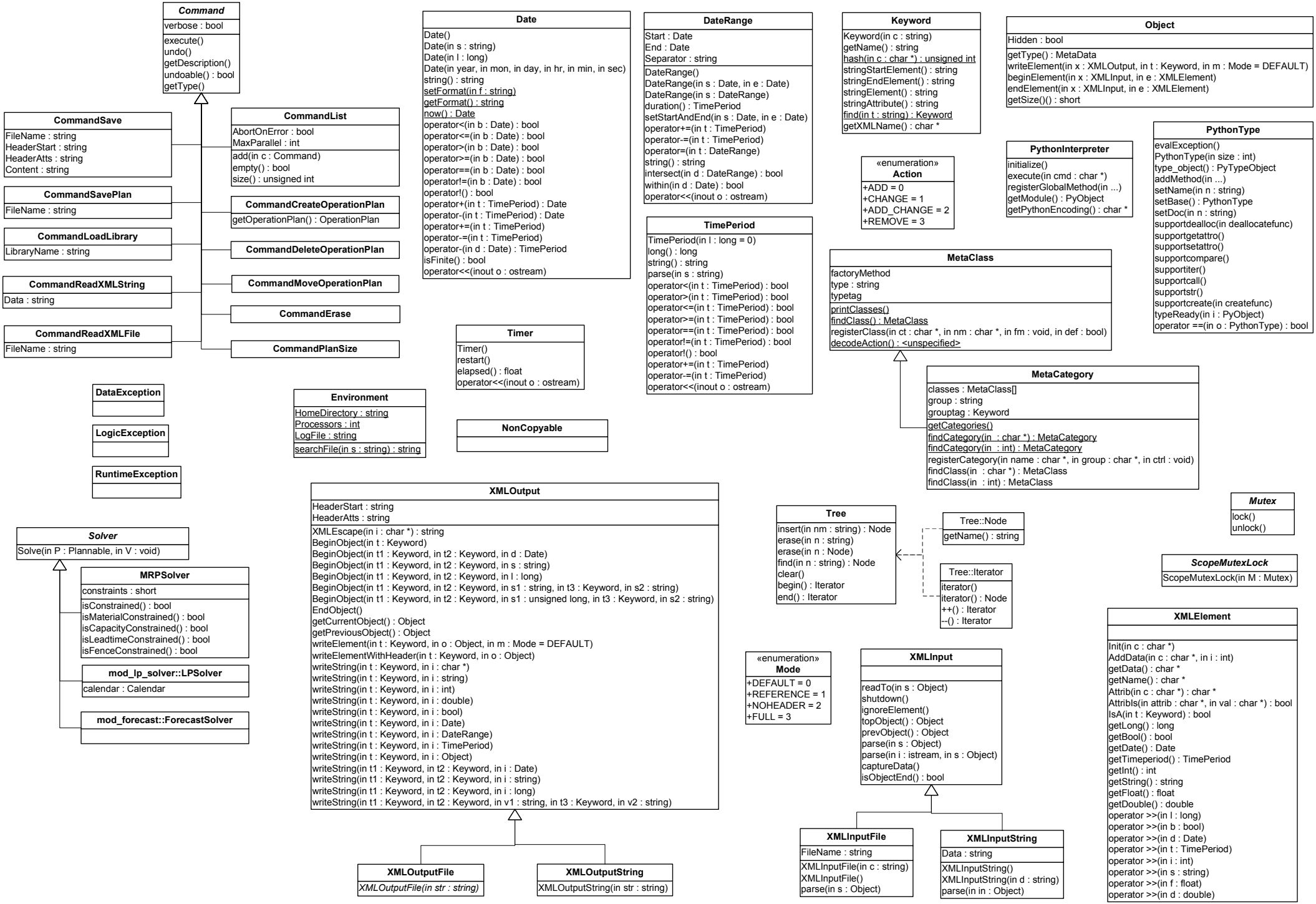
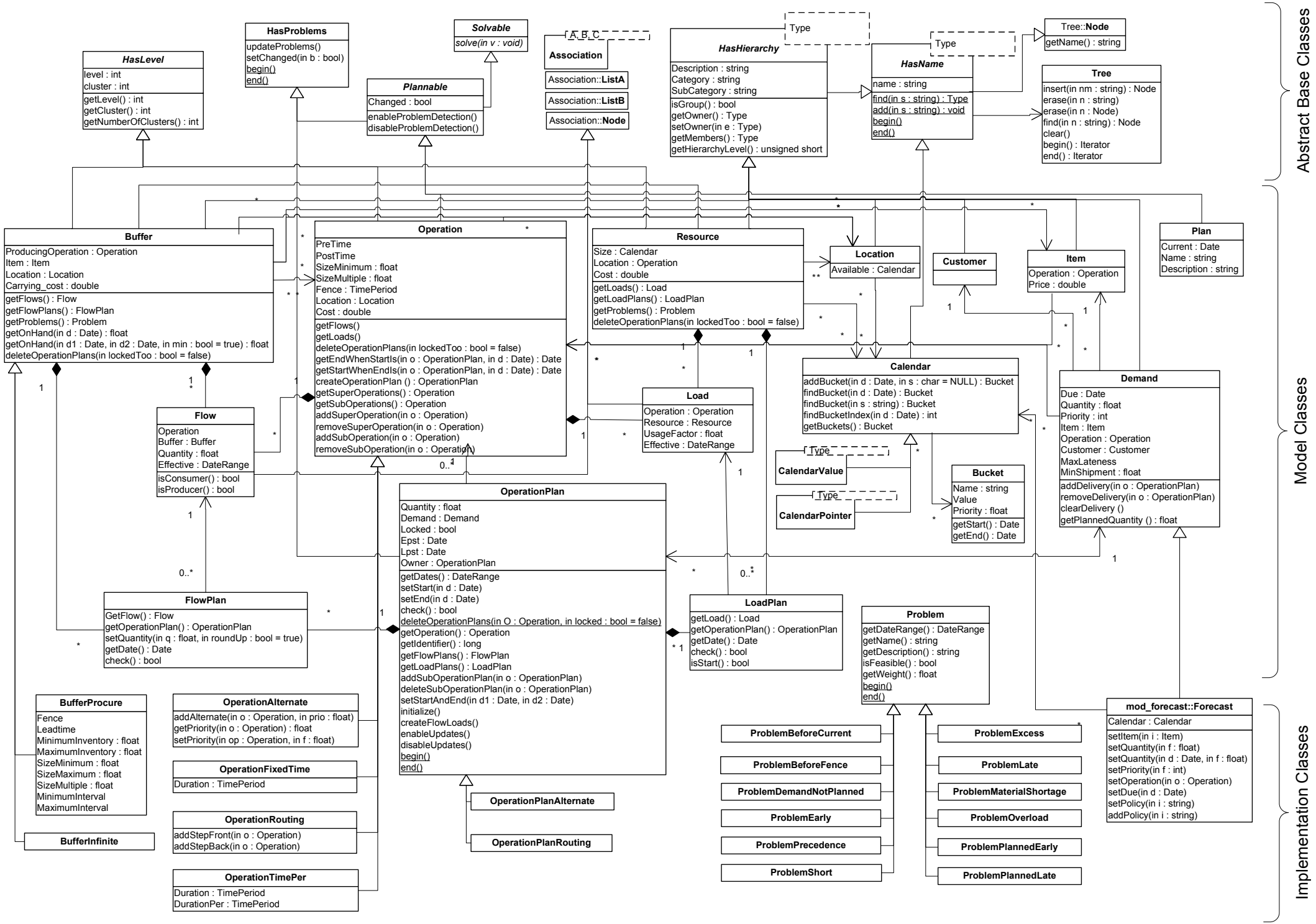

Command
verbose : bool
execute()
undo()
getDescription()
undoable() : bool
getType()
CommandSave
FileName : string
HeaderStart : string
HeaderAtts : string
Content : string
CommandSavePlan
FileName : string
CommandLoadLibrary
LibraryName : string
CommandReadXMLString
Data : string
CommandReadXMLFile
FileName : string
CommandList
AbortOnError : bool
MaxParallel : int
add(in c : Command)
empty() : bool
size() : unsigned int
CommandCreateOperationPlan
getOperationPlan() : OperationPlan
CommandDeleteOperationPlan
CommandMoveOperationPlan
CommandErase
CommandPlanSize
Date
Date()
Date(in s : string)
Date(in l : long)
Date(in year, in mon, in day, in hr, in min, in sec)
string() : string
setFormat(in f : string)
getFormat() : string
now() : Date
operator<(in b : Date) : bool
operator<=(in b : Date) : bool
operator>(in b : Date) : bool
operator>=(in b : Date) : bool
operator==(in b : Date) : bool
operator!=(in b : Date) : bool
operator!() : bool
operator+(in t : TimePeriod) : Date
operator-(in t : TimePeriod) : Date
operator+=(in t : TimePeriod)
operator-=(in t : TimePeriod)
operator-(in d : Date) : TimePeriod
isFinite() : bool
operator<<(inout o : ostream)
DateRange
Start : Date
End : Date
Separator : string
DateRange()
DateRange(in s : Date, in e : Date)
DateRange(in s : DateRange)
duration() : TimePeriod
setStartAndEnd(in s : Date, in e : Date)
operator+=(in t : TimePeriod)
operator-=(in t : TimePeriod)
operator=(in t : DateRange)
string() : string
intersect(in d : DateRange) : bool
within(in d : Date) : bool
operator<<(inout o : ostream)
TimePeriod
TimePeriod(in l : long = 0)
long() : long
string() : string
parse(in s : string)
operator<(in t : TimePeriod) : bool
operator>(in t : TimePeriod) : bool
operator<=(in t : TimePeriod) : bool
operator>=(in t : TimePeriod) : bool
operator==(in t : TimePeriod) : bool
operator!=(in t : TimePeriod) : bool
operator!() : bool
operator+=(in t : TimePeriod)
operator-=(in t : TimePeriod)
operator<<(inout o : ostream)
Keyword
Keyword(in c : string)
getName() : string
hash(in c : char *) : unsigned int
stringStartElement() : string
stringEndElement() : string
stringElement() : string
stringAttribute() : string
find(in t : string) : Keyword
getXMLName() : char *
Object
Hidden : bool
getType() : MetaData
writeElement(in x : XMLOutput, in t : Keyword, in m : Mode = DEFAULT)
beginElement(in x : XMLInput, in e : XMLElement)
endElement(in x : XMLInput, in e : XMLElement)
getSize()() : short
«enumeration»
Action
+ADD = 0
+CHANGE = 1
+ADD_CHANGE = 2
+REMOVE = 3
PythonInterpreter
initialize()
execute(in cmd : char *)
registerGlobalMethod(in ...)
getModule() : PyObject
getPythonEncoding() : char *
PythonType
evalException()
PythonType(in size : int)
type_object() : PyTypeObject
addMethod(in ...)
setName(in n : string)
setBase() : PythonType
setDoc(in n : string)
supportdealloc(in deallocatefunc)
supportgetattro()
supportsetattro()
supportcompare()
supportiter()
supportcall()
supportstr()
supportcreate(in createfunc)
typeReady(in i : PyObject)
operator ==(in o : PythonType) : bool
MetaClass
factoryMethod
type : string
typetag
printClasses()
findClass() : MetaClass
registerClass(in ct : char *, in nm : char *, in fm : void, in def : bool)
decodeAction() : <unspecified>
MetaCategory
classes : MetaClass[]
group : string
grouptag : Keyword
getCategories()
findCategory(in : char *) : MetaCategory
findCategory(in : int) : MetaCategory
registerCategory(in name : char *, in group : char *, in ctrl : void)
findClass(in : char *) : MetaClass
findClass(in : int) : MetaClass
Timer
Timer()
restart()
elapsed() : float
operator<<(inout o : ostream)
Environment
HomeDirectory : string
Processors : int
LogFile : string
searchFile(in s : string) : string
NonCopyable
DataException
LogicException
RuntimeException
Solver
Solve(in P : Plannable, in V : void)
MRPSolver
constraints : short
isConstrained() : bool
isMaterialConstrained() : bool
isCapacityConstrained() : bool
isLeadtimeConstrained() : bool
isFenceConstrained() : bool
mod_lp_solver::LPSolver
calendar : Calendar
mod_forecast::ForecastSolver
XMLOutput
HeaderStart : string
HeaderAtts : string
XMLEscape(in i : char *) : string
BeginObject(in t : Keyword)
BeginObject(in t1 : Keyword, in t2 : Keyword, in d : Date)
BeginObject(in t1 : Keyword, in t2 : Keyword, in s : string)
BeginObject(in t1 : Keyword, in t2 : Keyword, in l : long)
BeginObject(in t1 : Keyword, in t2 : Keyword, in s1 : string, in t3 : Keyword, in s2 : string)
BeginObject(in t1 : Keyword, in t2 : Keyword, in s1 : unsigned long, in t3 : Keyword, in s2 : string)
EndObject()
getCurrentObject() : Object
getPreviousObject() : Object
writeElement(in t : Keyword, in o : Object, in m : Mode = DEFAULT)
writeElementWithHeader(in t : Keyword, in o : Object)
writeString(in t : Keyword, in i : char *)
writeString(in t : Keyword, in i : string)
writeString(in t : Keyword, in i : int)
writeString(in t : Keyword, in i : double)
writeString(in t : Keyword, in i : bool)
writeString(in t : Keyword, in i : Date)
writeString(in t : Keyword, in i : DateRange)
writeString(in t : Keyword, in i : TimePeriod)
writeString(in t : Keyword, in i : Object)
writeString(in t1 : Keyword, in t2 : Keyword, in i : Date)
writeString(in t1 : Keyword, in t2 : Keyword, in i : string)
writeString(in t1 : Keyword, in t2 : Keyword, in i : long)
writeString(in t1 : Keyword, in t2 : Keyword, in v1 : string, in t3 : Keyword, in v2 : string)
XMLOutputFile
XMLOutputFile(in str : string)
XMLOutputString
XMLOutputString(in str : string)
Tree
insert(in nm : string) : Node
erase(in n : string)
erase(in n : Node)
find(in n : string) : Node
clear()
begin() : Iterator
end() : Iterator
Tree::Node
getName() : string
Tree::Iterator
iterator()
iterator() : Node
++() : Iterator
--() : Iterator
Mutex
lock()
unlock()
ScopeMutexLock
ScopeMutexLock(in M : Mutex)
«enumeration»
Mode
+DEFAULT = 0
+REFERENCE = 1
+NOHEADER = 2
+FULL = 3
XMLInput
readTo(in s : Object)
shutdown()
ignoreElement()
topObject() : Object
prevObject() : Object
parse(in s : Object)
parse(in i : istream, in s : Object)
captureData()
isObjectEnd() : bool
XMLInputFile
FileName : string
XMLInputFile(in c : string)
XMLInputFile()
parse(in s : Object)
XMLInputString
Data : string
XMLInputString()
XMLInputString(in d : string)
parse(in in : Object)
XMLElement
Init(in c : char *)
AddData(in c : char *, in i : int)
getData() : char *
getName() : char *
Attrib(in c : char *) : char *
AttribIs(in attrib : char *, in val : char *) : bool
IsA(in t : Keyword) : bool
getLong() : long
getBool() : bool
getDate() : Date
getTimeperiod() : TimePeriod
getInt() : int
getString() : string
getFloat() : float
getDouble() : double
operator >>(in l : long)
operator >>(in b : bool)
operator >>(in d : Date)
operator >>(in t : TimePeriod)
operator >>(in i : int)
operator >>(in s : string)
operator >>(in f : float)
operator >>(in d : double)


Abstract Base Classes
Model Classes
Implementation Classes
HasLevel
level : int
cluster : int
getLevel() : int
getCluster() : int
getNumberOfClusters() : int
HasProblems
updateProblems()
setChanged(in b : bool)
begin()
end()
Solvable
solve(in v : void)
Plannable
Changed : bool
enableProblemDetection()
disableProblemDetection()
A, B, C
Association
Association::ListA
Association::ListB
Association::Node
Type
HasHierarchy
Description : string
Category : string
SubCategory : string
isGroup() : bool
getOwner() : Type
setOwner(in e : Type)
getMembers() : Type
getHierarchyLevel() : unsigned short
Type
HasName
name : string
find(in s : string) : Type
add(in s : string) : void
begin()
end()
Tree::Node
getName() : string
Tree
insert(in nm : string) : Node
erase(in n : string)
erase(in n : Node)
find(in n : string) : Node
clear()
begin() : Iterator
end() : Iterator
Buffer
ProducingOperation : Operation
Item : Item
Location : Location
Carrying_cost : double
getFlows() : Flow
getFlowPlans() : FlowPlan
getProblems() : Problem
getOnHand(in d : Date) : float
getOnHand(in d1 : Date, in d2 : Date, in min : bool = true) : float
deleteOperationPlans(in lockedToo : bool = false)
Operation
PreTime
PostTime
SizeMinimum : float
SizeMultiple : float
Fence : TimePeriod
Location : Location
Cost : double
getFlows()
getLoads()
deleteOperationPlans(in lockedToo : bool = false)
getEndWhenStartIs(in o : OperationPlan, in d : Date) : Date
getStartWhenEndIs(in o : OperationPlan, in d : Date) : Date
createOperationPlan () : OperationPlan
getSuperOperations() : Operation
getSubOperations() : Operation
addSuperOperation(in o : Operation)
removeSuperOperation(in o : Operation)
addSubOperation(in o : Operation)
removeSubOperation(in o : Operation)
Resource
Size : Calendar
Location : Operation
Cost : double
getLoads() : Load
getLoadPlans() : LoadPlan
getProblems() : Problem
deleteOperationPlans(in lockedToo : bool = false)
Location
Available : Calendar
Customer
Item
Operation : Operation
Price : double
Plan
Current : Date
Name : string
Description : string
Flow
Operation
Buffer : Buffer
Quantity : float
Effective : DateRange
isConsumer() : bool
isProducer() : bool
Load
Operation : Operation
Resource : Resource
UsageFactor : float
Effective : DateRange
Calendar
addBucket(in d : Date, in s : char = NULL) : Bucket
findBucket(in d : Date) : Bucket
findBucket(in s : string) : Bucket
findBucketIndex(in d : Date) : int
getBuckets() : Bucket
Demand
Due : Date
Quantity : float
Priority : int
Item : Item
Operation : Operation
Customer : Customer
MaxLateness
MinShipment : float
addDelivery(in o : OperationPlan)
removeDelivery(in o : OperationPlan)
clearDelivery ()
getPlannedQuantity () : float
Type
CalendarValue
Type
CalendarPointer
Bucket
Name : string
Value
Priority : float
getStart() : Date
getEnd() : Date
OperationPlan
Quantity : float
Demand : Demand
Locked : bool
Epst : Date
Lpst : Date
Owner : OperationPlan
getDates() : DateRange
setStart(in d : Date)
setEnd(in d : Date)
check() : bool
deleteOperationPlans(in O : Operation, in locked : bool = false)
getOperation() : Operation
getIdentifier() : long
getFlowPlans() : FlowPlan
getLoadPlans() : LoadPlan
addSubOperationPlan(in o : OperationPlan)
deleteSubOperationPlan(in o : OperationPlan)
setStartAndEnd(in d1 : Date, in d2 : Date)
initialize()
createFlowLoads()
enableUpdates()
disableUpdates()
begin()
end()
FlowPlan
GetFlow() : Flow
getOperationPlan() : OperationPlan
setQuantity(in q : float, in roundUp : bool = true)
getDate() : Date
check() : bool
LoadPlan
getLoad() : Load
getOperationPlan() : OperationPlan
getDate() : Date
check() : bool
isStart() : bool
Problem
getDateRange() : DateRange
getName() : string
getDescription() : string
isFeasible() : bool
getWeight() : float
begin()
end()
mod_forecast::Forecast
Calendar : Calendar
setItem(in i : Item)
setQuantity(in f : float)
setQuantity(in d : Date, in f : float)
setPriority(in f : int)
setOperation(in o : Operation)
setDue(in d : Date)
setPolicy(in i : string)
addPolicy(in i : string)
BufferProcure
Fence
Leadtime
MinimumInventory : float
MaximumInventory : float
SizeMinimum : float
SizeMaximum : float
SizeMultiple : float
MinimumInterval
MaximumInterval
BufferInfinite
OperationAlternate
addAlternate(in o : Operation, in prio : float)
getPriority(in o : Operation) : float
setPriority(in op : Operation, in f : float)
OperationFixedTime
Duration : TimePeriod
OperationRouting
addStepFront(in o : Operation)
addStepBack(in o : Operation)
OperationTimePer
Duration : TimePeriod
DurationPer : TimePeriod
OperationPlanAlternate
OperationPlanRouting
ProblemBeforeCurrent
ProblemBeforeFence
ProblemDemandNotPlanned
ProblemEarly
ProblemPrecedence
ProblemShort
ProblemExcess
ProblemLate
ProblemMaterialShortage
ProblemOverload
ProblemPlannedEarly
ProblemPlannedLate