org.OpenMI.Standard interface specification v.1.4.0

September 2007 © The OpenMI Association URL: www.openmi.org

OpenMI

data definitions

What

«interface»
IQuantity
+ «property» ID() : string
+ «property» Description() : string
+ «property» ValueType() : ValueType
+ «property» Dimension() : IDimension
+ «property» Unit() : IUnit

«interface»
IUnit
+ «property» ID() : string
+ «property» Description() : string
+ «property» ConversionFactorToSI() : double
+ «property» OffSetToSI() : double

Values

«interface»
IValueSet
+ «property» Count() : int
+ IsValid(elementIndex :int) : bool

«enumeration»
ValueType
+ Scalar: int = 1
+ Vector: int = 2

«interface»
IDimension
+ Equals(otherDimension :IDimension) : bool
+ GetPower(baseQuantity :DimensionBase) : double

«interface»
IScalarSet
+ GetScalar(elementIndex :int) : double

«interface»
IVectorSet
+ GetVector(elementIndex :int) : IVector

«enumeration»
DimensionBase
+ Length: int = 0
+ Mass: int = 1
+ Time: int = 2
+ ElectricCurrent: int = 3
+ Temperature: int = 4
+ AmountOfSubstance: int = 5
+ LuminousIntensity: int = 6
+ Currency: int = 7
+ NUM_BASE_DIMENSIONS: int

«interface»
IVector
+ «property» XComponent() : double
+ «property» YComponent() : double
+ «property» ZComponent() : double

When

«interface»
ITime

«interface»
ITimeSpan
+ «property» Start() : ITimeStamp
+ «property» End() : ITimeStamp

«interface»
ITimeStamp
+ «property» ModifiedJulianDay() : double

Where

«interface»
IElementSet
+ «property» ID() : string
+ «property» Description() : string
+ «property» SpatialReference() : ISpatialReference
+ «property» ElementType() : ElementType
+ «property» ElementCount() : int
+ «property» Version() : int
+ GetElementIndex(elementID :string) : int
+ GetElementID(elementIndex :int) : string
+ GetVertexCount(elementIndex :int) : int
+ GetFaceCount(elementIndex :int) : int
+ GetFaceVertexIndices(elementIndex :int, faceIndex :int) : int[]
+ GetXCoordinate(elementIndex :int, vertexIndex :int) : double
+ GetYCoordinate(elementIndex :int, vertexIndex :int) : double
+ GetZCoordinate(elementIndex :int, vertexIndex :int) : double

«enumeration»
ElementType
+ IDBased: int = 0
+ XYPoint: int = 1
+ XYLine: int = 2
+ XYPolyLine: int = 3
+ XYPolygon: int = 4
+ XYZPoint: int = 5
+ XYZLine: int = 6
+ XYZPolyLine: int = 7
+ XYZPolygon: int = 8
+ XYZPolyhedron: int = 9

«interface»
ISpatialReference
+ «property» ID() : string

«interface»
IArgument
+ «property» Key() : string
+ «property» Value() : string
+ «property» ReadOnly() : bool
+ «property» Description() : string

How

«interface»
IDataOperation
+ «property» ID() : string
+ «property» ArgumentCount() : int
+ GetArgument(argumentIndex :int) : IArgument
+ Initialize(properties :IArgument[]) : void
+ IsValid(inputExchangeItem :IInputExchangeItem, outputExchangeItem :IOutputExchangeItem, SelectedDataOperations :IDataOperation[]) : bool

meta data to express what can be exchanged

«interface»
IExchangeItem
+ «property» Quantity() : IQuantity
+ «property» ElementSet() : IElementSet

«interface»
IInputExchangeItem

«interface»
IOutputExchangeItem
+ «property» DataOperationCount() : int
+ GetDataOperation(dataOperationIndex :int) : IDataOperation

messaging definitions

«interface»
IListener
+ OnEvent(Event :IEvent) : void
+ GetAcceptedEventTypeCount() : int
+ GetAcceptedEventType(acceptedEventTypeIndex :int) : EventType

«interface»
IEvent
+ «property» Type() : EventType
+ «property» Description() : string
+ «property» Sender() : ILinkableComponent
+ «property» SimulationTime() : ITimeStamp
+ GetAttribute(key :string) : object

«enumeration»
EventType
+ Warning: int = 0
+ Informative: int = 1
+ ValueOutOfRange: int = 2
+ GlobalProgress: int = 3
+ TimeStepProgres: int = 4
+ DataChanged: int = 5
+ TargetBeforeGetValuesCall: int = 6
+ SourceAfterGetValuesCall: int = 7
+ SourceBeforeGetValuesReturn: int = 8
+ TargetAfterGetValuesReturn: int = 9
+ Other: int = 10
+ NUM_OF_EVENT_TYPES: int = 2005

specification what will be exchanged and how

«interface»
ILink
+ «property» ID() : string
+ «property» Description() : string
+ «property» SourceComponent() : ILinkableComponent
+ «property» TargetComponent() : ILinkableComponent
+ «property» TargetQuantity() : IQuantity
+ «property» TargetElementSet() : IElementSet
+ «property» SourceElementSet() : IElementSet
+ «property» SourceQuantity() : IQuantity
+ «property» DataOperationsCount() : int
+ GetDataOperation(dataOperationIndex :int) : IDataOperation

component interfaces for generic component access

«interface»
IPublisher
+ Subscribe(listener :IListener, eventType :EventType) : void
+ UnSubscribe(listener :IListener, eventType :EventType) : void
+ SendEvent(Event :IEvent) : void
+ GetPublishedEventTypeCount() : int
+ GetPublishedEventType(providedEventTypeIndex :int) : EventType

optional extensions

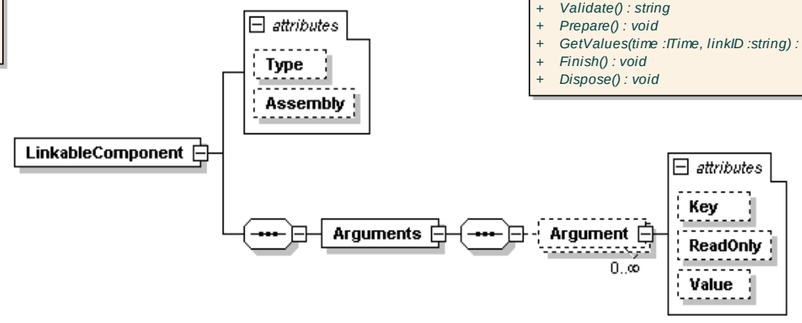
«interface»
IDiscreteTimes
+ HasDiscreteTimes(quantity :IQuantity, elementSet :IElementSet) : bool
+ GetDiscreteTimesCount(quantity :IQuantity, elementSet :IElementSet) : int
+ GetDiscreteTime(quantity :IQuantity, elementSet :IElementSet, discreteTimeIndex :int) : ITime

obligatory interface

«interface»
IManageState
+ KeepCurrentState() : string
+ RestoreState(stateID :string) : void
+ ClearState(stateID :string) : void

«interface»
ILinkableComponent
+ «property» ComponentID() : string
+ «property» ComponentDescription() : string
+ «property» ModelID() : string
+ «property» ModelDescription() : string
+ «property» InputExchangeItemCount() : int
+ «property» OutputExchangeItemCount() : int
+ «property» TimeHorizon() : ITimeSpan
+ «property» EarliestInputTime() : ITimeStamp
+ Initialize(properties :IArgument[]) : void
+ GetInputExchangeItem(inputExchangeItemIndex :int) : IInputExchangeItem
+ GetOutputExchangeItem(outputExchangeItemIndex :int) : IOutputExchangeItem
+ AddLink(link :ILink) : void
+ RemoveLink(linkID :string) : void
+ Validate() : string
+ Prepare() : void
+ GetValues(time :ITime, linkID :string) : IValueSet
+ Finish() : void
+ Dispose() : void

starting point: the OMI-file



OpenMI