Package com.telelogic.rhapsody.core

Interface Summary	
IRPAcceptEventAction	The IRPAcceptEventAction interface represents Accept Event Action elements in a statechart or activity diagram.
IRPAcceptTimeEvent	The IRPAcceptTimeEvent interface represents Accept Time Event elements in activity diagrams and statecharts.
IRPAction	The IRPAction interface represents the action defined for a transition in a statechart.
IRPActionBlock	The IRPActionBlock interface represents action blocks in sequence diagrams.
IRPActivityDiagram	The IRPActivityDiagram interface represents activity diagrams in Rational Rhapsody models.
IRPActor	The IRPActor interface represents actors in Rhapsody models.
IRPAnnotation	The IRPAnnotation interface represents the different types of annotations you can add to your model - notes, comments, constraints, and requirements.
IRPApplication	The IRPApplication interface represents the Rhapsody application, and its methods reflect many of the commands that you can access from the Rhapsody menu bar.
IRPArgument	The IRPArgument interface represents an argument of an operation or an event.
IRPASCIIFile	
IRPAssociationClass	The IRPAssociationClass interface represents association classes in Rational Rhapsody models.
IRPAssociationRole	The IRPAssociationRole interface represents the association roles that link objects in communication diagrams.
IRPAttribute	The IRPAttribute interface represents attributes of a class, and global variables.
IRPAXViewCtrl	
IRPBaseExternalCodeGeneratorTool	
IRPCallOperation	The IRPCallOperation interface represents call operation elements in activity diagrams.
IRPClass	The IRPClass interface represents classes in Rational Rhapsody models.

Interface Summary

IRPClassifier	Represents the features shared by elements such as classes, actors, use cases, and types.
IRPClassifierRole	
IRPCodeGenerator	
IRPCodeGenSimplifiersRegistry	
IRPCollaboration	The IRPCollaboration interface represents the capabilities included in sequence diagrams and communications diagrams.
IRPCollaborationDiagram	The IRPCollaborationDiagram interface represents collaboration diagrams in a Rational Rhapsody model.
IRPCollection	The IRPCollection interface contains methods used to store and manipulate collections of various types of elements that you may have in your Rational Rhapsody model.
IRPComment	The IRPComment interface represents comments in a Rational Rhapsody model.
IRPComponent	The IRPComponent interface represents a code generation component defined in a Rhapsody model.
IRPComponentDiagram	The IRPComponentDiagram interface represents component diagrams in Rational Rhapsody models.
IRPComponentInstance	
IRPConditionMark	The IRPConditionMark interface represents condition marks in sequence diagrams.
IRPConfiguration	The IRPConfiguration interface represents a code generation configuration within a specific component.
IRPConnector	The IRPConnector interface represents the characteristics shared by the various types of "connector" elements that can be included in a statechart, such as condition connectors, history connectors, join sync bar connectors, and fork sync bar connectors.
IRPConstraint	The IRPConstraint interface represents constraints in a Rational Rhapsody model.
IRPContextSpecification	The IRPContextSpecification interface represents the exact context of an object in a hierarchy.
IRPControlledFile	The IRPControlledFile interface represents a controlled file in a Rhapsody model.
IRPDependency	The IRPDependency interface represents dependencies in a Rational Rhapsody model.
IRPDeploymentDiagram	The IRPDeploymentDiagram interface represents deployment diagrams in Rational Rhapsody models.
IRPDestructionEvent	The IRPDestructionEvent interface represents destruction events

Interface Summary

	in sequence diagrams.
IRPDiagram	The IRPDiagram interface contains the methods shared by all the interfaces that represent specific types of diagrams.
IRPDiagSynthAPI	
IRPEnumerationLiteral	
IRPEvent	The IRPEvent interface represents events in Rational Rhapsody models.
IRPEventReception	
IRPExecutionOccurrence	
IRPExternalCheckRegistry	
IRPExternalCodeGeneratorInvoker	
IRPExternalIDERegistry	
IRPExternalRoundtripInvoker	
IRPFile	The IRPFile interface represents a file or folder to be generated during code generation.
IRPFileFragment	
IRPFlow	
IRPFlowchart	The IRPFlowchart interface represents activities in Rational Rhapsody models.
IRPFlowItem	The IRPFlowItem interface represents item flows in Rational Rhapsody models.
IRPGeneralization	
IRPGraphEdge	
IRPGraphElement	
IRPGraphicalProperty	
IRPGraphNode	
IRPGuard	
IRPHyperLink	The IRPHyperLink interface represents hyperlinks in Rational Rhapsody models.
IRPImageMap	
IRPInstance	
IRPInstanceSlot	
IRPInstanceSpecification	
IRPInstanceValue	

Interface Summary

	The IRPInstanceValue interface is used in contexts where a single model element must be stored.
IRPIntegrator	
IRPInteractionOccurrence	
IRPInteractionOperand	The IRPInteractionOperand interface represents interaction operands in Rational Rhapsody models.
IRPInteractionOperator	
IRPInterfaceItem	The IRPInterfaceItem interface represents the features shared by operations, events, and event receptions in Rational Rhapsody models.
IRPInternalOEMPlugin	
IRPJavaPlugins	
IRPLink	The IRPLink interface represents links in Rational Rhapsody models.
IRPLiteralSpecification	The IRPLiteralSpecification interface is used in contexts where a single value must be stored.
IRPMatrixLayout	
IRPMatrixView	The IRPMatrixView interface represents Matrix View elements in Rhapsody models.
IRPMessage	
IRPMessagePoint	
IRPModelElement	The IRPModelElement interface represents an element in a Rational Rhapsody model, and its methods reflect the behavior shared by the various types of model elements.
IRPModule	
IRPNode	
IRPObjectModelDiagram	The IRPObjectModelDiagram interface represents object model diagrams in Rational Rhapsody models.
IRPObjectNode	The IRPObjectNode interface represents Object Node elements in activity diagrams.
IRPOperation	The IRPOperation interface represents operations of classes in Rational Rhapsody models.
IRPPowListListener	
IRPPowPaneMgr	
IRPPowTextListener	
IRPPackage	

Interface Summary

	The IRPPackage interface represents packages in Rhapsody models.
IRPPanelDiagram	The IRPPanelDiagram interface represents panel diagrams in Rational Rhapsody models.
IRPPin	The IRPPin interface represents action pins added to actions, or activity parameters added to action blocks, in an activity diagram.
IRPPlugInWindow	
IRPPort	The IRPPort interface represents ports in Rational Rhapsody models.
IRPProfile	The IRPProfile interface represents profiles in Rational Rhapsody models.
IRPProgressBar	
IRPProject	The IRPProject interface represents Rational Rhapsody projects.
IRPRelation	Represents a relationship between two classes.
IRPRequirement	The IRPRequirement interface represents requirements in a Rational Rhapsody model.
IRPRhapsodyServer	
IRPRoundTrip	
IRPSearchManager	
IRPSearchQuery	The IRPSearchQuery interface represents the search criteria objects that are used by IRPSearchManager to carry out searches.
IRPSearchResult	
IRPSelection	
IRPSendAction	The IRPSendAction interface represents Send Action elements in an activity or statechart.
IRPSequenceDiagram	The IRPSequenceDiagram interface represents sequence diagrams in a Rational Rhapsody model.
IRPState	The IRPState interface represents states in a statechart.
IRPStatechart	The IRPStatechart interface represents the statechart elements underlying a statechart.
IRPStatechartDiagram	The IRPStatechartDiagram interface represents statecharts in a Rational Rhapsody model.
IRPStateVertex	The IRPStateVertex interface represents the characteristics that are shared by various statechart elements such as states, join/fork connectors, and condition connectors.
IRPStereotype	The IRPStereotype interface represents sterotypes in Rational Rhapsody models.

Interface Summary

IRPStructureDiagram	The IRPStructureDiagram interface represents structure diagrams in a Rational Rhapsody model.
IRPSwimlane	The IRPSwimlane interface represents swimlanes in an activity diagram.
IRPSysMLPort	The IRPSysMLPort interface represents flowport elements in Rhapsody models.
IRPTableLayout	
IRPTableView	The IRPTableView interface represents Table View elements in Rhapsody models.
IRPTag	The IRPTag interface represents tags in a Rational Rhapsody model.
IRPTemplateInstantiation	
IRPTemplateInstantiationParameter	
IRPTemplateParameter	The IRPTemplateParameter interface represents parameters of a template in Rational Rhapsody models.
IRPTimingDiagram	
IRPTransition	The IRPTransition interface represents transitions in a statechart.
IRPTrigger	The IRPTrigger interface represents the trigger of a transition in a statechart.
IRPType	
IRPUnit	The IRPUnit interface represents model elements that can be saved as separate files.
IRPUseCase	
IRPUseCaseDiagram	The IRPUseCaseDiagram interface represents use case diagrams in a Rational Rhapsody model.
IRPValueSpecification	The interface IRPValueSpecification represents the UML concept of "value specification" and serves as the base interface for IRPContextSpecification, IRPInstanceValue, and IRPLiteralSpecification.
IRPVariable	The IRPVariable interface represents the characteristics shared by model elements such as attributes, variables, and arguments.

Class Summary

HYPNameType	
IRPApplication.AddToModel_Mode	This class holds constant values to be used with addToModelEx method.

Class Summary

IRPGraphElement.ImageLayout	This class contains constant values for use with the method setImageLayout
IRPMatrixLayout.QueryOrElementsList	This class contains constant values for use with the methods setFromElementTypesUseQueryOrElementsList and setToElementTypesUseQueryOrElementsList.
IRPMatrixView.ContentFormat	This class contains values that specify export format
IRPModelElement.OSLCLink	Constant values used with elements of this type
IRPModelElement.OSLCLink.Types	This class contains values that specify OSLC Types
IRPSearchQuery.References	
IRPSearchQuery.References.QuantityOperator	
IRPSearchQuery.References.RelationKind	
IRPSearchQuery.SearchInField	Constant values used with elements of this type
IRPSearchQuery.SubQueriesOperator	
IRPSearchQuery.UnresolvedKind	
IRPSearchQuery.ViewsToSearch	
IRPTableLayout.Column	This class holds constant values to be used with addColumn method.
IRPTableLayout.Column.AnnotationAttribute	Contains values to be used for Property parameter of addColumn method, when AnnotationAttribute is selected for the Type parameter of addColumn method.
IRPTableLayout.Column.DependsOn	Contains the pre-defined values to be used for Property parameter of addColumn method, when DependsOn is selected for the Type parameter of addColumn method.
IRPTableLayout.Column.FlowAttribute	Contains values to be used for Property parameter of addColumn method, when FlowAttribute is selected for the Type parameter of addColumn method.
IRPTableLayout.Column.GeneralAttribute	Contains values to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.
IRPTableLayout.Column.ImplementationCellType	
IRPTableLayout.Column.RelationAttributeFrom	Contains values to be used for Property parameter of addColumn method, when RelationAttributeFrom is selected for the Type parameter of addColumn method.

Class Summary

IRPTableLayout.Column.RelationAttributeTo	Contains values to be used for Property parameter of addColumn method, when RelationAttributeTo is selected for the Type parameter of addColumn method.
IRPTableLayout.Column.RequirementAttribute	Contains values to be used for Property parameter of addColumn method, when RequirementAttribute is selected for the Type parameter of addColumn method.
IRPTableLayout.Column.UserDefinedMethod	Contains values to be used for Property parameter of addColumn method, when USER_DEFINED_METHOD is selected for the Type parameter of addColumn method.
IRPTableLayout.QueryOrElementsList	This class contains constant values for use with the methods setFromElementTypesUseQueryOrElementsList and setToElementTypesUseQueryOrElementsList.
IRPTableView.ContentFormat	This class contains values that specify export format
RhapsodyAppServer	The RhapsodyAppServer class contains methods relating to accessing an instance of Rhapsody.
RhpClassLoader	
RhpUtils	
RPApplicationListener	
RPCodeGeneratorListener	
RPCodeGenSimplifier	
RPExtendedRPClassesFactory	
RPExternalCheck	
RPExternalCodeGenerator	
RPExternalIDEManager	
RPExternalRoundtrip	
RPIntegratorListener	
RPJavaPluginsManager	
RPowPaneMgrEvents	
RPRoundTripListener	
RPRTCListener	
RPSearchListener	
RPUserPlugin	

Class Summary

SearchFindAsEnum	

Exception Summary

RhapsodyRuntimeException	

---

com.telelogic.rhapsody.core
Class HYPNameType

```
java.lang.Object
  └─ com.telelogic.rhapsody.core.HYPNameType
```

public class **HYPNameType**
extends java.lang.Object

Field Summary

static char	**RP_HYP_FREETEXT** show user defined name
static char	**RP_HYP_LABELTEXT** show hyperlink target label
static char	**RP_HYP_NAMETEXT** show hyperlink target name
static char	**RP_HYP_TAGVALUETEXT** show hyperlink target tag value

Constructor Summary

HYPNameType()

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

RP_HYP_FREETEXT

public static final char **RP_HYP_FREETEXT**

> show user defined name

> **See Also:**
> > [Constant Field Values](#)

RP_HYP_NAMETEXT

public static final char **RP_HYP_NAMETEXT**

> show hyperlink target name

> **See Also:**
> > [Constant Field Values](#)

RP_HYP_LABELTEXT

public static final char **RP_HYP_LABELTEXT**

> show hyperlink target label

> **See Also:**
> > [Constant Field Values](#)

RP_HYP_TAGVALUETEXT

public static final char **RP_HYP_TAGVALUETEXT**

> show hyperlink target tag value

> **See Also:**
> > [Constant Field Values](#)

Constructor Detail

HYPNameType

public **HYPNameType**()

---

com.telelogic.rhapsody.core

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPAcceptEventAction

All Superinterfaces:
> IRPModelElement, IRPState, IRPStateVertex

---

public interface **IRPAcceptEventAction**
extends IRPState

The IRPAcceptEventAction interface represents Accept Event Action elements in a statechart or activity diagram. To create an Accept Event Action element, use the method IRPFlowchart.addAcceptEventAction.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
IRPModelElement.OSLCLink

Method Summary

IRPEvent	**getEvent**() Returns the event that the action waits for.
void	**setEvent**(IRPEvent event) Specifies the event that the action should wait for.

Methods inherited from interface com.telelogic.rhapsody.core.**IRPState**
addActivityFinal, addConnector, addInternalTransition, addState, addStaticReaction, addTerminationState, createDefaultTransition, createNestedStatechart, deleteConnector, deleteInternalTransition, deleteStaticReaction, getDefaultTransition, getEntryAction, getExitAction, getFullNameInStatechart, getInheritsFrom, getInternalTransitions, getIsOverridden, getIsReferenceActivity, getItsStatechart, getItsSwimlane, getLogicalStates, getNestedStatechart, getReferenceToActivity, getSendAction, getStateType, getStaticReactions, getSubStates, getSubStateVertices, getTheEntryAction, getTheExitAction, isAnd, isCompound, isLeaf, isRoot, isSendActionState, overrideInheritance, resetEntryActionInheritance, resetExitActionInheritance, setEntryAction, setExitAction, setInternalTransition, setItsSwimlane, setReferenceToActivity, setStateType, setStaticReaction, unoverrideInheritance

Methods inherited from interface com.telelogic.rhapsody.core.**IRPStateVertex**

com.telelogic.rhapsody.core

Methods inherited from interface com.telelogic.rhapsody.core.**IRPStateVertex**
addFlow, addTransition, deleteTransition, getInTransitions, getOutTransitions, getParent, setParent

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getEvent

IRPEvent **getEvent**()

Returns the event that the action waits for.
Returns:
the event that the action waits for

setEvent

void **setEvent**(IRPEvent event)

Specifies the event that the action should wait for.

Parameters:

event - the event that the action should wait for

---

com.telelogic.rhapsody.core
Interface IRPAcceptTimeEvent

All Superinterfaces:
> IRPModelElement, IRPState, IRPStateVertex

---

public interface **IRPAcceptTimeEvent**
extends IRPState

The IRPAcceptTimeEvent interface represents Accept Time Event elements in activity diagrams and statecharts.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

java.lang.String	**getDurationTime**() Returns the duration that was specified for this element.
void	**setDurationTime**(java.lang.String durationTime) Specifies the duration that should be used for this element.

Methods inherited from interface com.telelogic.rhapsody.core.IRPState
addActivityFinal, addConnector, addInternalTransition, addState, addStaticReaction, addTerminationState, createDefaultTransition, createNestedStatechart, deleteConnector, deleteInternalTransition, deleteStaticReaction, getDefaultTransition, getEntryAction, getExitAction, getFullNameInStatechart, getInheritsFrom, getInternalTransitions, getIsOverridden, getIsReferenceActivity, getItsStatechart, getItsSwimlane, getLogicalStates, getNestedStatechart, getReferenceToActivity, getSendAction, getStateType, getStaticReactions, getSubStates, getSubStateVertices, getTheEntryAction, getTheExitAction, isAnd, isCompound, isLeaf, isRoot, isSendActionState, overrideInheritance, resetEntryActionInheritance, resetExitActionInheritance, setEntryAction, setExitAction, setInternalTransition, setItsSwimlane, setReferenceToActivity, setStateType, setStaticReaction, unoverrideInheritance

Methods inherited from interface com.telelogic.rhapsody.core.IRPStateVertex

Methods inherited from interface com.telelogic.rhapsody.core.IRPStateVertex
addFlow, addTransition, deleteTransition, getInTransitions, getOutTransitions, getParent, setParent

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getDurationTime

```
java.lang.String getDurationTime()
```

Returns the duration that was specified for this element.

Returns:

the duration that was specified for this element

setDurationTime

```
void setDurationTime(java.lang.String durationTime)
```

Specifies the duration that should be used for this element.

Parameters:
> `durationTime` - the duration that should be used for this element

---

com.telelogic.rhapsody.core

com.telelogic.rhapsody.core
Interface IRPAction

All Superinterfaces:
> IRPModelElement

All Known Subinterfaces:
> IRPSendAction

---

public interface **IRPAction**
extends IRPModelElement

The IRPAction interface represents the action defined for a transition in a statechart.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

java.lang.String	**getBody**() Gets the code defined as the action for the transition.
void	**setBody**(java.lang.String body) Used to specify the code that serves as the action for the transition.

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks,

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getBody

```
java.lang.String getBody()
```

> Gets the code defined as the action for the transition.
> **Returns:**
> > the code defined as the action for the transition.

setBody

```
void setBody(java.lang.String body)
```

> Used to specify the code that serves as the action for the transition.
> **Parameters:**
> > body - The code that should be used as the action for the transition.

com.telelogic.rhapsody.core
Interface IRPActionBlock

All Superinterfaces:
> IRPMessage, IRPModelElement

---

public interface **IRPActionBlock**
extends **IRPMessage**

The IRPActionBlock interface represents action blocks in sequence diagrams.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

Methods inherited from interface com.telelogic.rhapsody.core.IRPMessage
addSourceExecutionOccurrence, addTargetExecutionOccurrence, getActualParameterList, getCommunicationConnection, getCondition, getDurationConstraint, getDurationObservation, getFlowPort, getFormalInterfaceItem, getFormalType, getInvariant, getMessageType, getPort, getReturnValue, getSequenceNumber, getSignature, getSource, getSourceExecutionOccurrence, getTarget, getTargetExecutionOccurrence, getTimeConstraint, getTimeObservation, getTimerValue, reroute, setActualParameterList, setDurationConstraint, setDurationObservation, setFlowPort, setFormalInterfaceItem, setFormalType, setInvariant, setPort, setReturnValue, setTimeConstraint, setTimeObservation, setTimerValue

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal,

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

---

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | **METHOD**   DETAIL: FIELD | CONSTR | **METHOD**

com.telelogic.rhapsody.core
Interface IRPActivityDiagram

All Superinterfaces:
 IRPDiagram, IRPModelElement, IRPStatechartDiagram, IRPUnit

---

public interface **IRPActivityDiagram**
extends IRPStatechartDiagram

The IRPActivityDiagram interface represents activity diagrams in Rational Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPFlowchart	**getFlowchart**() Returns the IRPFlowchart object underlying the activity diagram.

Methods inherited from interface com.telelogic.rhapsody.core.IRPStatechartDiagram
addAndLine, createGraphics, getStatechart

Methods inherited from interface com.telelogic.rhapsody.core.IRPDiagram
addFreeShapeByType, addImage, addNewEdgeByType, addNewEdgeForElement, addNewNodeByType, addNewNodeForElement, addTextBox, closeDiagram, completeRelations, getCorrespondingGraphicElements, getElementsInDiagram, getGraphicalElements, getLastVisualizationModifiedTime, getPicture, getPictureAs, getPictureAsDividedMetafiles, getPictureEx, getPicturesWithImageMap, isOpen, isShowDiagramFrame, openDiagram, openDiagramView, populateDiagram, removeGraphElements, setShowDiagramFrame

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit
copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference,

Methods inherited from interface com.telelogic.rhapsody.core.**IRPUnit**
referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getFlowchart

IRPFlowchart **getFlowchart**()

> Returns the IRPFlowchart object underlying the activity diagram.
> **Returns:**
> > the IRPFlowchart object underlying the activity diagram

---

com.telelogic.rhapsody.core
Interface IRPActor

All Superinterfaces:
> IRPClassifier, IRPModelElement, IRPUnit

```
public interface IRPActor
```
extends IRPClassifier

The IRPActor interface represents actors in Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

int	**getIsBehaviorOverriden**() Checks whether an actor does not inherit the behavior defined in the statechart of its base class.
void	**setIsBehaviorOverriden**(int isBehaviorOverriden) Specifies whether an actor should inherit the behavior defined in the statechart of its base class.

Methods inherited from interface com.telelogic.rhapsody.core.IRPClassifier
addActivityDiagram, addAttribute, addFlowItems, addFlows, addGeneralization, addOperation, addRelation, addRelationTo, addStatechart, addUnidirectionalRelation, addUnidirectionalRelationTo, deleteAttribute, deleteFlowItems, deleteFlows, deleteGeneralization, deleteOperation, deleteRelation, findAttribute, findBaseClassifier, findDerivedClassifier, findGeneralization, findInterfaceItem, findNestedClassifier, findNestedClassifierRecursive, findRelation, findTrigger, getActivityDiagram, getAttributes, getAttributesIncludingBases, getBaseClassifiers, getBehavioralDiagrams, getDerivedClassifiers, getFlowItems, getFlows, getGeneralizations, getInterfaceItems, getInterfaceItemsIncludingBases, getLinks, getNestedClassifiers, getOperations, getPorts, getRelations, getRelationsIncludingBases, getSequenceDiagrams, getSourceArtifacts, getStatechart

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

Methods inherited from interface com.telelogic.rhapsody.core.**IRPUnit**
copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getIsBehaviorOverriden

```
int getIsBehaviorOverriden()
```

Checks whether an actor does not inherit the behavior defined in the statechart of its base class. When you create a statechart for an actor, by default it inherits the behavior defined in the statechart of its base class. However, Rhapsody allows you to specify that the actor should not inherit this behavior. This operation checks whether this option has been exercised for the current actor.

Returns:

indication of whether or not the actor inherits the behavior specified in the statechart of its base class. 1 means that it does not inherit this behavior, 0 means that it does inherit the

behavior defined in the statechart of the base class.

setIsBehaviorOverriden

```
void setIsBehaviorOverriden(int isBehaviorOverriden)
```

Specifies whether an actor should inherit the behavior defined in the statechart of its base class. When you create a statechart for an actor, by default it inherits the behavior defined in the statechart of its base class. However, Rhapsody allows you to specify that the actor should not inherit this behavior.

Parameters:

isBehaviorOverriden - use 1 if you do not want the actor to inherit the behavior defined in the statechart of its base class. Use 0 if you want the actor to inherit this behavior.

com.telelogic.rhapsody.core
Interface IRPAnnotation

All Superinterfaces:
> IRPModelElement, IRPUnit

All Known Subinterfaces:
> IRPComment, IRPConstraint, IRPRequirement

```
public interface IRPAnnotation
```
extends IRPUnit

The IRPAnnotation interface represents the different types of annotations you can add to your model - notes, comments, constraints, and requirements.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

void	**addAnchor**(IRPModelElement target) Adds an anchor from the annotation to the specified model element.
IRPCollection	**getAnchoredByMe**() Gets the list of model elements that are anchored to the annotation.
java.lang.String	**getBody**() Gets the text of the specification for the annotation.
java.lang.String	**getSpecification**() Gets the text of the specification for the annotation.
java.lang.String	**getSpecificationRTF**() Returns the specification of the annotation in RTF format.
int	**isSpecificationRTF**() Checks whether the specification is in RTF format
void	**removeAnchor**(IRPModelElement target) Removes the anchor to the specified model element.

Method Summary

void	**setBody**(java.lang.String body) Adds a specification to the annotation.
void	**setSpecification**(java.lang.String specification) Adds a specification to the annotation.
void	**setSpecificationRTF**(java.lang.String specificationRTF) Specifies RTF string to use for the specification of the annotation.

Methods inherited from interface com.telelogic.rhapsody.core.**IRPUnit**

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addAnchor

void **addAnchor**([IRPModelElement](#) target)

> Adds an anchor from the annotation to the specified model element.
> **Parameters:**
> > target - the model element the annotation should be anchored to

getAnchoredByMe

[IRPCollection](#) **getAnchoredByMe**()

> Gets the list of model elements that are anchored to the annotation.
> **Returns:**
> > the model elements that are anchored to the annotation.

getBody

java.lang.String **getBody**()

> Gets the text of the specification for the annotation.
> **Returns:**
> > the text of the specification for the annotation

getSpecification

java.lang.String **getSpecification**()

> Gets the text of the specification for the annotation.
> **Returns:**
> > the text of the specification for the annotation

getSpecificationRTF

java.lang.String **getSpecificationRTF**()

> Returns the specification of the annotation in RTF format.
> **Returns:**
> > the specification of the annotation in RTF format

isSpecificationRTF

int **isSpecificationRTF**()

> Checks whether the specification is in RTF format

Returns:
1 if the specification is in RTF format, 0 otherwise

---

removeAnchor

void **removeAnchor**(IRPModelElement target)

Removes the anchor to the specified model element.
Parameters:
target - the model element for which the anchor should be removed

---

setBody

void **setBody**(java.lang.String body)

Adds a specification to the annotation.
Parameters:
body - the text to use as the specification

---

setSpecification

void **setSpecification**(java.lang.String specification)

Adds a specification to the annotation.
Parameters:
specification - the text to use as the specification

---

setSpecificationRTF

void **setSpecificationRTF**(java.lang.String specificationRTF)

Specifies RTF string to use for the specification of the annotation.
Parameters:
specificationRTF - the RTF string to use for the specification of the annotation

---

Package   **Class**   [Use](Use)   Tree   Serialized   Deprecated   Index   Help

PREV CLASS **NEXT CLASS**   **FRAMES**   **NO FRAMES**   **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**　　　　　　　　**FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD　　　DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Class IRPApplication.AddToModel_Mode

```
java.lang.Object
   └─ com.telelogic.rhapsody.core.IRPApplication.AddToModel_Mode
```

Enclosing interface:
>   IRPApplication

---

public static final class **IRPApplication.AddToModel_Mode**
extends java.lang.Object

This class holds constant values to be used with addToModelEx method.

Field Summary

static int	**AS_REFERENCE** A reference to the unit should be added to the model (unit cannot be modified).
static int	**AS_UNIT_WITH_COPY** The unit should be added to the model and its file should be copied to the project directory.
static int	**AS_UNIT_WITHOUT_COPY** The unit should be added to the model as an editable unit, but its file should not be copied to the project directory.

Constructor Summary

IRPApplication.AddToModel_Mode()	

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

AS_UNIT_WITH_COPY

`public static final int **AS_UNIT_WITH_COPY**`

> The unit should be added to the model and its file should be copied to the project directory.
>
> **See Also:**
> > Constant Field Values

AS_UNIT_WITHOUT_COPY

`public static final int **AS_UNIT_WITHOUT_COPY**`

> The unit should be added to the model as an editable unit, but its file should not be copied to the project directory.
>
> **See Also:**
> > Constant Field Values

AS_REFERENCE

`public static final int **AS_REFERENCE**`

> A reference to the unit should be added to the model (unit cannot be modified).
>
> **See Also:**
> > Constant Field Values

Constructor Detail

IRPApplication.AddToModel_Mode

`public **IRPApplication.AddToModel_Mode**()`

[Package](#) **Class** **[Use](#)** **[Tree](#)** **[Serialized](#)** **[Deprecated](#)** **[Index](#)** **[Help](#)**

[PREV CLASS](#) **[NEXT CLASS](#)**   **[FRAMES](#)** **[NO FRAMES](#)** **[All Classes](#)**

SUMMARY: [NESTED](#) | FIELD | CONSTR | [METHOD](#)   DETAIL: FIELD | CONSTR | [METHOD](#)

com.telelogic.rhapsody.core
Interface IRPApplication

public interface **IRPApplication**

The IRPApplication interface represents the Rhapsody application, and its methods reflect many of the commands that you can access from the Rhapsody menu bar.

Nested Class Summary

static class	**[IRPApplication.AddToModel_Mode](#)** This class holds constant values to be used with addToModelEx method.

Method Summary

[IRPProject](#)	**[activeProject](#)**() Returns an IRPProject object representing the project currently open in Rhapsody
[IRPUnit](#)	**[addProfileToModel](#)**(java.lang.String profileName) addProfileToModel
void	**[addSelectedToFavorites](#)**() Adds the currently selected item to the Favorites list.
void	**[addToModel](#)**(java.lang.String filename, int withDescendant) add To Model
void	**[addToModelByReference](#)**(java.lang.String filename) add To Model by reference
void	**[addToModelEx](#)**(java.lang.String filename, int addToModelMode, int addSubUnits, int addDependents) Adds a unit to the model.
void	**[addToModelFromURL](#)**(java.lang.String purl) add To Model From URL
void	**[allowBrowserRefresh](#)**(int shouldRefresh) allowBrowserRefresh
void	**[allowGERefresh](#)**(int shouldRefresh) allowGERefresh
void	

	Method Summary
	applyNewTermsProfile(java.lang.String profileName) Called to apply a NewTerms Profile to the active project
void	**arcCheckOut**(java.lang.String filename, java.lang.String label, int isLocked, int isRecursive) archive Check Out
void	**bringWindowToTop**() bring window to top
void	**build**() Builds an application using the active component and configuration.
void	**buildEntireProject**() buildEntireProject
void	**buildWithDependencies**() buildWithDependencies
int	**canRedo**() Check if Redo action is available
int	**canUndo**() Check if Undo action is available
void	**checkIn**(java.lang.String unitName, java.lang.String label, int isLocked, int isRecursive, java.lang.String description) check In
void	**checkModel**() check model
void	**checkOut**(java.lang.String unitName, java.lang.String label, int isLocked, int isRecursive) check Out
void	**clean**() clean
void	**clearOutputWindow**(java.lang.String title) clear output window
void	**closeAllAnimatedSequenceDiagrams**(int withSave) Close All Animated Sequence diagrams without save
void	**compareSequenceDiagram**([IRPSequenceDiagram](#) leftDiagram, [IRPSequenceDiagram](#) rightDiagram) Compares the two sequence diagrams specified as parameters.
void	**connectToArchive**(java.lang.String archivePath) connect To Archive
void	**connectToImportedModel**(java.lang.String serverURL, java.lang.String projectAreaName, java.lang.String streamName)

Method Summary

	Used to connect to an externally-managed model.
void	**connectToTarget**(java.lang.String targetName) connectToTarget
int	**createAndInsertProject**(java.lang.String projectLocation, java.lang.String projectName) Create a new project and insert it into current workspace
void	**createDomainFromProfile**(IRPProfile profileArg, java.lang.String serverURL) Create Domain from Profile After Login
IRPCollection	**createNewCollection**() creates a new Rhapsody collection object
void	**createNewProject**(java.lang.String projectLocation, java.lang.String projectName) Creates a new Rhapsody project
void	**dbgCheckComIn**(short i) For debug - check communication in
short	**dbgCheckComOut**() For debug - check communication out
void	**deferredAddToModel**(java.lang.String filename, int withDescendants, java.lang.String origPrjId, int eraseDir) Add Rhapsody unit to current project
int	**deleteProjectFromList**(java.lang.String projectName) Delete specified project from current workspace
void	**disconnectFromTarget**() disconnectFromTarget
void	**dMRefreshRecursive**(IRPUnit pUnit) dMRefreshRecursive
void	**dMSyncAndRefresh**(IRPProject projectArg, int sync, int refresh) dMSyncAndRefresh
void	**downloadToTarget**(java.lang.String filename) downloadToTarget
void	**endUndoTransaction**() end undo transaction
void	**enterAnimationCommand**(java.lang.String command) enter Animation Command
java.lang.String	**errorMessage**() Returns error message for last method called.
void	**executeCommand**(java.lang.String commandType, IRPCollection pCommandInitialization,

Method Summary

	[IRPCollection](#) pCommandResult) method execute command
int	**executeCommandLine**(java.lang.String commandLine) Execute command line
void	**executeTransformationSequence**(java.lang.String transformationSeque int showTransformedModelPackage) executeTransformationSequence
java.lang.String	**expandStringKeywords**(java.lang.String theString) expand environment-variable keywords in the provided string
java.lang.String	**fixpack**() Get Rhapsody fixpack
void	**forceOutput2Console**(int val) Force output to system console
void	**forceRoundtrip**() forceRoundtrip
void	**forceRoundtripElements**([IRPCollection](#) elements) method forceRoundtripElements
void	**generate**() Generates code for the entire project, using the active component and configuration.
void	**generateElements**([IRPCollection](#) elements) method generateElements
void	**generateEntireProject**() generateEntireProject
void	**generateMainAndMakeFiles**() Generate Main and Make Files
void	**generateWithDependencies**() generateWithDependencies
java.lang.String	**getApplicationConnectionString**() getApplicationConnectionString
void	**getApplicationName**(java.lang.String applicationName, java.lang.String productRCPName) Get application name
int	**getApplicationStatus**() getApplicationStatus
java.lang.String	**getBuildNo**() get property BuildNo
[IRPCodeGenSimplifiersRegistry](#)	**getCodeGenSimplifiersRegistry**() get the code generation simplifiers registry

Method Summary

IRPDiagram	**getDiagramOfSelectedElement**() get diagram of selected element
IRPDiagSynthAPI	**getDiagSynthAPI**(java.lang.String clientName) for internal use
int	**getDMBoolProperty**(java.lang.String pKey) Returns the value of the specified Design Manager boolean property
java.lang.String	**getDMModelWorkspaceFolder**() Returns the path of the workspace folder used for the DM model.
java.lang.String	**getDMProperty**(java.lang.String pKey) Returns the value of the specified Design Manager property
java.lang.String	**getErrorMessage**() Returns error message for last method called.
java.lang.String	**getExecutableFolder**() getExecutableFolder
IRPExternalCheckRegistry	**getExternalCheckerRegistry**() get the External Checker registry
IRPExternalIDERegistry	**getExternalIDERegistry**(java.lang.String clientID) get the External IDE registry
IRPExternalRoundtripInvoker	**getExternalRoundtripInvoker**() getExternalRoundtripInvoker
java.lang.String	**getIniFileParameterValue**(java.lang.String sectionName, java.lang.String paramName) getIniFileParameterValue
java.lang.String	**getInterfaceName**() get property interfaceName
int	**getIsHiddenUI**() get property isHiddenUI
int	**getIsLoadOnDemand**() get property isLoadOnDemand
java.lang.String	**getLanguage**() get property Language
IRPCollection	**getListOfFactoryProperties**() get list of factory properties
IRPCollection	**getListOfSelectedElements**() get list of selected elements
IRPCollection	**getListOfSiteProperties**() get list of site properties
java.lang.String	

Method Summary

	getLocaleName() Returns the locale for the version of Rhapsody running.
IRPModelElement	**getModelElementFromSource**(java.lang.String sourceData, int isSourceDataIsfileName, int lineNumber) Find model element from source code
java.lang.String	**getOMROOT**() get property OMROOT
java.lang.String	**getOutputWindowText**() Returns the text displayed in the output window.
IRPowPaneMgr	**getOWPaneMgr**(java.lang.String clientID) For internal use only.
IRPPlugInWindow	**getPlugInWindow**(int nPlugInID, int nWindowID, int nCreateNew) PlugIn window factory
IRPCollection	**getProjects**() get property projects
long	**getRhapsodyHandleErrorFunction**() getRhapsodyHandleErrorFunction
long	**getRhapsodyHandleErrorFunctionLong**() getRhapsodyHandleErrorFunctionLong
IRPSearchManager	**getSearchManager**() get Rhapsody search manager
IRPModelElement	**getSelectedElement**() get selected element
IRPCollection	**getSelectedGraphElements**() get selected graph elements
IRPSelection	**getSelection**() Get Selection Interface
java.lang.String	**getSerialNo**() get property SerialNo
IRPCodeGenerator	**getTheCodeGeneratorInterface**() get codegeneration interface
IRPExternalCodeGeneratorInvoker	**getTheExternalCodeGeneratorInvoker**() get external code generator invoker
IRPIntegrator	**getTheIntegratorInterface**() get integrator interface
IRPJavaPlugins	**getTheJavaPluginsInterface**() getTheJavaPluginsInterface
IRPRoundTrip	

Method Summary

	getTheRoundtripInterface() get roundtrip interface
java.lang.String	**getToolSet**() get property ToolSet
void	**highlightByHandle**(java.lang.String strHandle) highlight by handle
void	**highLightElement**(IRPModelElement val) highlight element
void	**importClasses**() import Classes
void	**importSyncSimulinkBlock2**(IRPModelElement simulinkBlock, java.lang.String matlabExePath, java.lang.String simMdlFile, java.lang.String simSrcFiles, java.lang.String sampleTime) Imports a Simulink model into a Rhapsody model.
void	**importTlb**(java.lang.String pPath) import tlb
IRPProject	**insertProject**(java.lang.String filename) Insert existing project into current workspace
IRPProject	**insertProjectFromDesignManager**(java.lang.String userName, java.lang.String passwd, java.lang.String serverURL, java.lang.String projectAreaName, java.lang.String streamName, java.lang.String modelName) insertProjectFromDesignManager
void	**invokePluginsMethod**(java.lang.String methodName) invoke Plugins Method
void	**invokeRPE**() Opens the IBM Rational Rhapsody Report Generator wizard.
int	**isRhapsodyCL**() Is RhapsodyCL
int	**isRhapsodyFileType**(java.lang.String extension) Check if specified extension corresponds to any Rhapsody unit type
int	**loginToDesignManagerWithAlias**(java.lang.String serverURL, java.lang.String alias) loginToDesignManagerWithAlias
int	**loginToDesignManagerWithCertificate**(java.lang.String serverURL, java.lang.String certificateLocation, java.lang.String password) loginToDesignManagerWithCertificate
int	**loginToDesignManagerWithUsername**(java.lang.String serverURL, java.lang.String userName, java.lang.String password)

Method Summary

	loginToDesignManagerWithUsername
void	**make**() make
void	**mergeElements**([IRPModelElement](#) left, [IRPModelElement](#) right) mergeElements
[IRPProject](#)	**newProjectOnDesignManager**(java.lang.String userName, java.lang.String password, java.lang.String serverURL, java.lang.String projectAreaName, java.lang.String streamName, java.lang.String modelName) Creates a new project on the Design Manager server.
void	**notifyFileChanged**(java.lang.String filename, int contentChanged) File change notification
[IRPAXViewCtrl](#)	**openActiveXView**(java.lang.String viewType, [IRPCollection](#) pViewInitialization, [IRPCollection](#) pExtra) Open ActivexView
void	**openAdvancedSearchAndReplaceDialog**() Open Advanced Search and Replace dialog
[IRPAXViewCtrl](#)	**openDiagramView**([IRPDiagram](#) diagram) method OpenDiagramView
void	**openFileList**(java.lang.String filename) method openFileList
[IRPProject](#)	**openProject**(java.lang.String filename) Opens an existing Rhapsody project
[IRPProject](#)	**openProjectFromDesignManager**(java.lang.String userName, java.lang.String password, java.lang.String serverURL, java.lang.String projectAreaName, java.lang.String streamName, java.lang.String modelName) openProjectFromDesignManager
[IRPProject](#)	**openProjectFromDesignManagerAfterLogin**(java.lang.String serverURL, java.lang.String projectAreaName, java.lang.String streamName, java.lang.String modelName) openProjectFromDesignManagerAfterLogin
void	**openProjectFromURL**(java.lang.String purl) open Project From URL
[IRPProject](#)	**openProjectWithLastSession**(java.lang.String filename) open project with last session
[IRPProject](#)	**openProjectWithoutSubUnits**(java.lang.String filename) open project without subunits
void	**quit**() quit application
void	

Method Summary

	rebuild() rebuild
void	**rebuildEntireProject**() rebuildEntireProject
void	**rebuildWithDependencies**() rebuildWithDependencies
int	**redo**() Perform Redo
void	**refactorSelectedOperation**(java.lang.String newName) Changes the name of the currently-selected operation and updates any references to the operation accordingly.
void	**refreshAllViews**() refresh all views
void	**regenerate**() regenerate
void	**regenerateElements**(IRPCollection elements) method regenerateElements
void	**regenerateEntireProject**() regenerateEntireProject
void	**regenerateWithDependencies**() regenerateWithDependencies
void	**registerCOMClient**(int processID, java.lang.String clientFilename, int magicNumber) register COM client
void	**report**(java.lang.String format, java.lang.String outputFileName) report
void	**rhpCheckinLicense**(java.lang.String feature) checkin license
java.lang.String	**rhpCheckoutLicense**(java.lang.String feature) checkout license
void	**roundtrip**() roundtrip
void	**roundtripElements**(IRPCollection elements) method roundtripElements
void	**runApplication**() Runs the application that was built for the project
void	**runHelper**(java.lang.String helperName) runHelper

	Method Summary
void	**runHelperWithParameters**(java.lang.String helperName, java.lang.String params) runHelperWithParameters
void	**saveAll**() method saveAll
void	**selectGraphElements**(IRPCollection graphElements) Selects multiple elements in the most recently opened diagram.
void	**selectModelElements**(IRPCollection modelElements) Selects multiple items in the model browser.
void	**setApplicationStatus**(int nStatus) setApplicationStatus
void	**setComponent**(java.lang.String component) set Component
void	**setConfiguration**(java.lang.String configuration) set Configuration
void	**setDMBoolProperty**(java.lang.String pKey, int val) Sets the value of the specified boolean Design Manager property
void	**setDMProperty**(java.lang.String pKey, java.lang.String val) Sets the value of the specified Design Manager property
void	**setHiddenUI**(boolean pVal) set property hiddenUI
void	**setIsLoadOnDemand**(int isLoadOnDemand) set property isLoadOnDemand
void	**setLanguage**(java.lang.String language) set property Language
void	**setLog**(java.lang.String logFile) set log file
void	**setToolSet**(java.lang.String toolSet) set property ToolSet
void	**setUpdateRecentFileList**(int shouldUpdate) setUpdateRecentFileList
void	**startUndoTransaction**() start undo transaction
void	**syncBuild**() syncBuild
void	**terminateApplication**() Terminate the Application
int	**undo**() Perform Undo

Method Summary

void	**unloadFromTarget**() unloadFromTarget
void	**unregisterCOMClient**(int processID, java.lang.String clientFilename, int magicNumber) unregister COM client
java.lang.String	**version**() Get Rhapsody version
java.lang.String	**versionNumberLong**() Get Rhapsody versionNumberLong
void	**writeToOutputWindow**(java.lang.String title, java.lang.String outputStr) Writes text to Rhapsody's Output window.

Method Detail

executeCommand

```
void executeCommand(java.lang.String commandType,
                    IRPCollection pCommandInitialization,
                    IRPCollection pCommandResult)
```

method execute command
Throws:
          RhapsodyRuntimeException

---

getPlugInWindow

```
IRPPlugInWindow getPlugInWindow(int nPlugInID,
                                int nWindowID,
                                int nCreateNew)
```

PlugIn window factory
Throws:
          RhapsodyRuntimeException

---

openActiveXView

```
IRPAXViewCtrl openActiveXView(java.lang.String viewType,
                              IRPCollection pViewInitialization,
                              IRPCollection pExtra)
```

Open ActivexView

Throws:
>   RhapsodyRuntimeException

---

openDiagramView

IRPAXViewCtrl **openDiagramView**(IRPDiagram diagram)

>   method OpenDiagramView
>   **Throws:**
>   RhapsodyRuntimeException

---

rhpCheckinLicense

void **rhpCheckinLicense**(java.lang.String feature)

>   checkin license
>   **Throws:**
>   RhapsodyRuntimeException

---

rhpCheckoutLicense

java.lang.String **rhpCheckoutLicense**(java.lang.String feature)

>   checkout license
>   **Throws:**
>   RhapsodyRuntimeException

---

activeProject

IRPProject **activeProject**()

>   Returns an IRPProject object representing the project currently open in Rhapsody
>   **Returns:**
>   IRPProject object that represents the project currently open in Rhapsody

---

addProfileToModel

IRPUnit **addProfileToModel**(java.lang.String profileName)

>   addProfileToModel
>   **Throws:**
>   RhapsodyRuntimeException

---

addSelectedToFavorites

void **addSelectedToFavorites**()

Adds the currently selected item to the Favorites list.

addToModel

void **addToModel**(java.lang.String filename,
                int withDescendant)

add To Model
Throws:
    RhapsodyRuntimeException

addToModelByReference

void **addToModelByReference**(java.lang.String filename)

add To Model by reference
Throws:
    RhapsodyRuntimeException

addToModelEx

void **addToModelEx**(java.lang.String filename,
                  int addToModelMode,
                  int addSubUnits,
                  int addDependents)

Adds a unit to the model.
Parameters:
    filename - the full path to the file to add to the model
    addToModelMode - how the unit should be added to the model - see
    IRPApplication.AddToModel_Mode for the available values
    addSubUnits - use 1 if you want to also add the sub-units of the unit, 0 otherwise (this
    parameter is ignored if the addToModelMode parameter equals
    IRPApplication.AddToModel_Mode.AS_UNIT_WITHOUT_COPY)
    addDependents - use 1 if you want to also add the units that elements in the specified unit are
    dependent upon, 0 otherwise (this parameter is ignored if the addToModelMode parameter
    equals IRPApplication.AddToModel_Mode.AS_UNIT_WITHOUT_COPY)

addToModelFromURL

void **addToModelFromURL**(java.lang.String purl)

add To Model From URL

Throws:
> [RhapsodyRuntimeException](#)

allowBrowserRefresh

void **allowBrowserRefresh**(int shouldRefresh)

> allowBrowserRefresh
> **Throws:**
> > [RhapsodyRuntimeException](#)

allowGERefresh

void **allowGERefresh**(int shouldRefresh)

> allowGERefresh
> **Throws:**
> > [RhapsodyRuntimeException](#)

applyNewTermsProfile

void **applyNewTermsProfile**(java.lang.String profileName)

> Called to apply a NewTerms Profile to the active project
> **Throws:**
> > [RhapsodyRuntimeException](#)

arcCheckOut

void **arcCheckOut**(java.lang.String filename,
                java.lang.String label,
                int isLocked,
                int isRecursive)

> archive Check Out
> **Throws:**
> > [RhapsodyRuntimeException](#)

bringWindowToTop

void **bringWindowToTop**()

> bring window to top
> **Throws:**
> > [RhapsodyRuntimeException](#)

build

void **build**()

       Builds an application using the active component and configuration. Use
       IRPProject.setActiveComponent and IRPProject.setActiveConfiguration to change the active
       component and configuration if necessary before calling the build method.

buildEntireProject

void **buildEntireProject**()

       buildEntireProject
       Throws:
              RhapsodyRuntimeException

buildWithDependencies

void **buildWithDependencies**()

       buildWithDependencies
       Throws:
               RhapsodyRuntimeException

canRedo

int **canRedo**()

       Check if Redo action is available
       Throws:
               RhapsodyRuntimeException

canUndo

int **canUndo**()

       Check if Undo action is available
       Throws:
               RhapsodyRuntimeException

checkIn

```
void checkIn(java.lang.String unitName,
          java.lang.String label,
          int isLocked,
          int isRecursive,
```

```
            java.lang.String description)
```

      check In
Throws:
         RhapsodyRuntimeException

---

checkModel

```
void checkModel()
```

      check model
Throws:
         RhapsodyRuntimeException

---

checkOut

```
void checkOut(java.lang.String unitName,
             java.lang.String label,
             int isLocked,
             int isRecursive)
```

      check Out
Throws:
         RhapsodyRuntimeException

---

clean

```
void clean()
```

      clean
Throws:
         RhapsodyRuntimeException

---

clearOutputWindow

```
void clearOutputWindow(java.lang.String title)
```

      clear output window
Throws:
         RhapsodyRuntimeException

---

closeAllAnimatedSequenceDiagrams

```
void closeAllAnimatedSequenceDiagrams(int withSave)
```

      Close All Animated Sequence diagrams without save

Throws:
> RhapsodyRuntimeException

compareSequenceDiagram

void **compareSequenceDiagram**(IRPSequenceDiagram leftDiagram,
                              IRPSequenceDiagram rightDiagram)

> Compares the two sequence diagrams specified as parameters. Corresponds to the Sequence Diagram Compare option in the Tools menu.
> **Parameters:**
>> leftDiagram - the first diagram to use for the comparison
>> rightDiagram - the second diagram to use for the comparison

connectToArchive

void **connectToArchive**(java.lang.String archivePath)

> connect To Archive
> **Throws:**
>> RhapsodyRuntimeException

connectToImportedModel

void **connectToImportedModel**(java.lang.String serverURL,
                              java.lang.String projectAreaName,
                              java.lang.String streamName)

> Used to connect to an externally-managed model. The method assumes you have already logged in to the relevant server, using one of the loginToDesignManager* methods in IRPApplication.
> **Parameters:**
>> serverURL - - the URL of the Design Manager server
>> projectAreaName - - the name of the Project Area where the model is located
>> streamName - the name of the Stream where the model is located

connectToTarget

void **connectToTarget**(java.lang.String targetName)

> connectToTarget
> **Throws:**
>> RhapsodyRuntimeException

createAndInsertProject

```
int createAndInsertProject(java.lang.String projectLocation,
                           java.lang.String projectName)
```

Create a new project and insert it into current workspace
Throws:
   RhapsodyRuntimeException

---

createDomainFromProfile

```
void createDomainFromProfile(IRPProfile profileArg,
                             java.lang.String serverURL)
```

Create Domain from Profile After Login
Throws:
   RhapsodyRuntimeException

---

createNewCollection

```
IRPCollection createNewCollection()
```

creates a new Rhapsody collection object
Returns:
   IRPCollection object that represents the collection that was created. After creating a
   collection, you can add items to it by calling IRPCollection.addItem or by calling
   IRPCollection.setSize and then IRPCollection.setModelElement.

---

createNewProject

```
void createNewProject(java.lang.String projectLocation,
                      java.lang.String projectName)
```

Creates a new Rhapsody project
Parameters:
   projectLocation - the directory where the project should be saved, for example,
   "l:\\temp_sample_code"
   projectName - the name to use for the project. This will be the name used for the .rpy file.

---

dMRefreshRecursive

```
void dMRefreshRecursive(IRPUnit pUnit)
```

dMRefreshRecursive
Throws:
   RhapsodyRuntimeException

---

dMSyncAndRefresh

```
void dMSyncAndRefresh(IRPProject projectArg,
                      int sync,
                      int refresh)
```

dMSyncAndRefresh
Throws:
     RhapsodyRuntimeException

---

dbgCheckComIn

```
void dbgCheckComIn(short i)
```

For debug - check communication in
Throws:
     RhapsodyRuntimeException

---

dbgCheckComOut

```
short dbgCheckComOut()
```

For debug - check communication out
Throws:
     RhapsodyRuntimeException

---

deferredAddToModel

```
void deferredAddToModel(java.lang.String filename,
                        int withDescendants,
                        java.lang.String origPrjId,
                        int eraseDir)
```

Add Rhapsody unit to current project
Throws:
     RhapsodyRuntimeException

---

deleteProjectFromList

```
int deleteProjectFromList(java.lang.String projectName)
```

Delete specified project from current workspace
Throws:
     RhapsodyRuntimeException

---

disconnectFromTarget

void **disconnectFromTarget**()

>   disconnectFromTarget
>   **Throws:**
>>   RhapsodyRuntimeException

---

downloadToTarget

void **downloadToTarget**(java.lang.String filename)

>   downloadToTarget
>   **Throws:**
>>   RhapsodyRuntimeException

---

endUndoTransaction

void **endUndoTransaction**()

>   end undo transaction
>   **Throws:**
>>   RhapsodyRuntimeException

---

enterAnimationCommand

void **enterAnimationCommand**(java.lang.String command)

>   enter Animation Command
>   **Throws:**
>>   RhapsodyRuntimeException

---

errorMessage

java.lang.String **errorMessage**()

>   Returns error message for last method called. If the last method completed successfully, then this
>   method returns an empty string. To get the correct error message for a method, errorMessage() must
>   be called immediately after the method is called.
>   **Returns:**
>>   the error message for the last method called

---

executeCommandLine

int **executeCommandLine**(java.lang.String commandLine)

> Execute command line
> **Throws:**
>> RhapsodyRuntimeException

---

executeTransformationSequence

void **executeTransformationSequence**(java.lang.String transformationSequence,
                                    int showTransformedModelPackage)

> executeTransformationSequence
> **Throws:**
>> RhapsodyRuntimeException

---

expandStringKeywords

java.lang.String **expandStringKeywords**(java.lang.String theString)

> expand environment-variable keywords in the provided string
> **Throws:**
>> RhapsodyRuntimeException

---

fixpack

java.lang.String **fixpack**()

> Get Rhapsody fixpack
> **Throws:**
>> RhapsodyRuntimeException

---

forceOutput2Console

void **forceOutput2Console**(int val)

> Force output to system console
> **Throws:**
>> RhapsodyRuntimeException

---

forceRoundtrip

void **forceRoundtrip**()

> forceRoundtrip

Throws:
>   RhapsodyRuntimeException

---

forceRoundtripElements

void **forceRoundtripElements**(IRPCollection elements)

>   method forceRoundtripElements
>   **Throws:**
>   RhapsodyRuntimeException

---

generate

void **generate**()

>   Generates code for the entire project, using the active component and configuration. Use IRPProject.setActiveComponent and IRPProject.setActiveConfiguration to change the active component and configuration if necessary before calling the generate method.

---

generateElements

void **generateElements**(IRPCollection elements)

>   method generateElements
>   **Throws:**
>   RhapsodyRuntimeException

---

generateEntireProject

void **generateEntireProject**()

>   generateEntireProject
>   **Throws:**
>   RhapsodyRuntimeException

---

generateMainAndMakeFiles

void **generateMainAndMakeFiles**()

>   Generate Main and Make Files
>   **Throws:**
>   RhapsodyRuntimeException

---

generateWithDependencies

void **generateWithDependencies**()

> generateWithDependencies
> **Throws:**
> > RhapsodyRuntimeException

---

getApplicationConnectionString

java.lang.String **getApplicationConnectionString**()

> getApplicationConnectionString
> **Throws:**
> > RhapsodyRuntimeException

---

getApplicationName

void **getApplicationName**(java.lang.String applicationName,
                        java.lang.String productRCPName)

> Get application name
> **Throws:**
> > RhapsodyRuntimeException

---

getApplicationStatus

int **getApplicationStatus**()

> getApplicationStatus
> **Throws:**
> > RhapsodyRuntimeException

---

getBuildNo

java.lang.String **getBuildNo**()

> get property BuildNo
> **Throws:**
> > RhapsodyRuntimeException

---

getCodeGenSimplifiersRegistry

IRPCodeGenSimplifiersRegistry **getCodeGenSimplifiersRegistry**()

> get the code generation simplifiers registry

Throws:
> [RhapsodyRuntimeException](#)

---

getDMBoolProperty

int **getDMBoolProperty**(java.lang.String pKey)

> Returns the value of the specified Design Manager boolean property
> **Parameters:**
>> pKey - the name of the boolean property whose value should be returned
>
> **Returns:**
>> the value of the specified Design Manager boolean property - 0 if the value of the property is
>> False, 1 if the value of the property is True

---

getDMModelWorkspaceFolder

java.lang.String **getDMModelWorkspaceFolder**()

> Returns the path of the workspace folder used for the DM model.
> **Returns:**
>> the path of the workspace folder used for the DM model

---

getDMProperty

java.lang.String **getDMProperty**(java.lang.String pKey)

> Returns the value of the specified Design Manager property
> **Parameters:**
>> pKey - the name of the property whose value should be returned
>
> **Returns:**
>> the value of the specified Design Manager property

---

getDiagSynthAPI

[IRPDiagSynthAPI](#) **getDiagSynthAPI**(java.lang.String clientName)

> for internal use
> **Throws:**
>> [RhapsodyRuntimeException](#)

---

getDiagramOfSelectedElement

[IRPDiagram](#) **getDiagramOfSelectedElement**()

> get diagram of selected element

getCodeGenSimplifiersRegistry 57

Throws:
[RhapsodyRuntimeException](#)

getErrorMessage

`java.lang.String` **getErrorMessage**()

Returns error message for last method called. If the last method completed successfully, then this method returns an empty string. To get the correct error message for a method, errorMessage() must be called immediately after the method is called.

Returns:
the error message for the last method called

getExecutableFolder

`java.lang.String` **getExecutableFolder**()

getExecutableFolder
Throws:
[RhapsodyRuntimeException](#)

getExternalCheckerRegistry

[IRPExternalCheckRegistry](#) **getExternalCheckerRegistry**()

get the External Checker registry
Throws:
[RhapsodyRuntimeException](#)

getExternalIDERegistry

[IRPExternalIDERegistry](#) **getExternalIDERegistry**(java.lang.String clientID)

get the External IDE registry
Throws:
[RhapsodyRuntimeException](#)

getExternalRoundtripInvoker

[IRPExternalRoundtripInvoker](#) **getExternalRoundtripInvoker**()

getExternalRoundtripInvoker
Throws:
[RhapsodyRuntimeException](#)

getIniFileParameterValue

```
java.lang.String getIniFileParameterValue(java.lang.String sectionName,
                                           java.lang.String paramName)
```

getIniFileParameterValue
Throws:
> RhapsodyRuntimeException

getInterfaceName

```
java.lang.String getInterfaceName()
```

get property interfaceName
Throws:
> RhapsodyRuntimeException

getIsHiddenUI

```
int getIsHiddenUI()
```

get property isHiddenUI
Throws:
> RhapsodyRuntimeException

getIsLoadOnDemand

```
int getIsLoadOnDemand()
```

get property isLoadOnDemand
Throws:
> RhapsodyRuntimeException

getLanguage

```
java.lang.String getLanguage()
```

get property Language
Throws:
> RhapsodyRuntimeException

getListOfFactoryProperties

```
IRPCollection getListOfFactoryProperties()
```

get list of factory properties

Throws:
      [RhapsodyRuntimeException](#)

---

getListOfSelectedElements

[IRPCollection](#) **getListOfSelectedElements**()

get list of selected elements
Throws:
      [RhapsodyRuntimeException](#)

---

getListOfSiteProperties

[IRPCollection](#) **getListOfSiteProperties**()

get list of site properties
Throws:
      [RhapsodyRuntimeException](#)

---

getLocaleName

java.lang.String **getLocaleName**()

Returns the locale for the version of Rhapsody running.
Returns:
      the locale of the version of Rhapsody running, for example, EN for the English version or JA
      for the Japanese version

---

getModelElementFromSource

[IRPModelElement](#) **getModelElementFromSource**(java.lang.String sourceData,
                                    int isSourceDataIsfileName,
                                    int lineNumber)

Find model element from source code
Throws:
      [RhapsodyRuntimeException](#)

---

getOMROOT

java.lang.String **getOMROOT**()

get property OMROOT
Throws:
      [RhapsodyRuntimeException](#)

---

getOWPaneMgr

[IRPowPaneMgr](#) **getOWPaneMgr**(java.lang.String clientID)

> For internal use only.

getOutputWindowText

java.lang.String **getOutputWindowText**()

> Returns the text displayed in the output window.
> **Returns:**
> > the text displayed in the output window

getProjects

[IRPCollection](#) **getProjects**()

> get property projects
> **Throws:**
> > [RhapsodyRuntimeException](#)

getRhapsodyHandleErrorFunction

long **getRhapsodyHandleErrorFunction**()

> getRhapsodyHandleErrorFunction
> **Throws:**
> > [RhapsodyRuntimeException](#)

getRhapsodyHandleErrorFunctionLong

long **getRhapsodyHandleErrorFunctionLong**()

> getRhapsodyHandleErrorFunctionLong
> **Throws:**
> > [RhapsodyRuntimeException](#)

getSearchManager

[IRPSearchManager](#) **getSearchManager**()

> get Rhapsody search manager
> **Throws:**
> > [RhapsodyRuntimeException](#)

getSelectedElement

IRPModelElement **getSelectedElement**()

get selected element
Throws:
RhapsodyRuntimeException

getSelectedGraphElements

IRPCollection **getSelectedGraphElements**()

get selected graph elements
Throws:
RhapsodyRuntimeException

getSelection

IRPSelection **getSelection**()

Get Selection Interface
Throws:
RhapsodyRuntimeException

getSerialNo

java.lang.String **getSerialNo**()

get property SerialNo
Throws:
RhapsodyRuntimeException

getTheCodeGeneratorInterface

IRPCodeGenerator **getTheCodeGeneratorInterface**()

get codegeneration interface
Throws:
RhapsodyRuntimeException

getTheExternalCodeGeneratorInvoker

[IRPExternalCodeGeneratorInvoker](#) **getTheExternalCodeGeneratorInvoker**()

> get external code generator invoker
> **Throws:**
> > [RhapsodyRuntimeException](#)

getTheIntegratorInterface

[IRPIntegrator](#) **getTheIntegratorInterface**()

> get integrator interface
> **Throws:**
> > [RhapsodyRuntimeException](#)

getTheJavaPluginsInterface

[IRPJavaPlugins](#) **getTheJavaPluginsInterface**()

> getTheJavaPluginsInterface
> **Throws:**
> > [RhapsodyRuntimeException](#)

getTheRoundtripInterface

[IRPRoundTrip](#) **getTheRoundtripInterface**()

> get roundtrip interface
> **Throws:**
> > [RhapsodyRuntimeException](#)

getToolSet

java.lang.String **getToolSet**()

> get property ToolSet
> **Throws:**
> > [RhapsodyRuntimeException](#)

highLightElement

void **highLightElement**([IRPModelElement](#) val)

> highlight element
> **Throws:**

getTheExternalCodeGeneratorInvoker 63

RhapsodyRuntimeException

---

highlightByHandle

void **highlightByHandle**(java.lang.String strHandle)

> highlight by handle
> **Throws:**
>> RhapsodyRuntimeException

---

importClasses

void **importClasses**()

> import Classes
> **Throws:**
>> RhapsodyRuntimeException

---

importSyncSimulinkBlock2

void **importSyncSimulinkBlock2**(IRPModelElement simulinkBlock,
                                  java.lang.String matlabExePath,
                                  java.lang.String simMdlFile,
                                  java.lang.String simSrcFiles,
                                  java.lang.String sampleTime)

> Imports a Simulink model into a Rhapsody model.
> **Parameters:**
>> simulinkBlock - the SimulinkBlock element that you created in your model (Object with
>> SimulinkBlock stereotype applied to it)
>> matlabExePath - the full path to the Matlab executable
>> simMdlFile - the full path to the Simulink model file
>> simSrcFiles - the full path for each of the .cpp files generated for the Simulink model
>> (except for ert_main.cpp). If there is one then more source file, the paths should be separated
>> by a semi-colon.
>> sampleTime - the interval (in milliseconds) at which Rational Rhapsody should activate the
>> Simulink engine

---

importTlb

void **importTlb**(java.lang.String pPath)

> import tlb
> **Throws:**
>> RhapsodyRuntimeException

---

insertProject

[IRPProject](#) **insertProject**(java.lang.String filename)

>
>   Insert existing project into current workspace
>   **Throws:**
>   [RhapsodyRuntimeException](#)

---

insertProjectFromDesignManager

[IRPProject](#) **insertProjectFromDesignManager**(java.lang.String userName,
                                                    java.lang.String passwd,
                                                    java.lang.String serverURL,
                                                    java.lang.String projectAreaName,
                                                    java.lang.String streamName,
                                                    java.lang.String modelName)

>
>   insertProjectFromDesignManager
>   **Throws:**
>   [RhapsodyRuntimeException](#)

---

invokePluginsMethod

void **invokePluginsMethod**(java.lang.String methodName)

>
>   invoke Plugins Method
>   **Throws:**
>   [RhapsodyRuntimeException](#)

---

invokeRPE

void **invokeRPE**()

>
>   Opens the IBM Rational Rhapsody Report Generator wizard.

---

isRhapsodyCL

int **isRhapsodyCL**()

>
>   Is RhapsodyCL
>   **Throws:**
>   [RhapsodyRuntimeException](#)

---

isRhapsodyFileType

```
int isRhapsodyFileType(java.lang.String extension)
```

Check if specified extension corresponds to any Rhapsody unit type
Throws:
> RhapsodyRuntimeException

loginToDesignManagerWithAlias

```
int loginToDesignManagerWithAlias(java.lang.String serverURL,
                                  java.lang.String alias)
```

loginToDesignManagerWithAlias
Throws:
> RhapsodyRuntimeException

loginToDesignManagerWithCertificate

```
int loginToDesignManagerWithCertificate(java.lang.String serverURL,
                                        java.lang.String certificateLocation,
                                        java.lang.String password)
```

loginToDesignManagerWithCertificate
Throws:
> RhapsodyRuntimeException

loginToDesignManagerWithUsername

```
int loginToDesignManagerWithUsername(java.lang.String serverURL,
                                     java.lang.String userName,
                                     java.lang.String password)
```

loginToDesignManagerWithUsername
Throws:
> RhapsodyRuntimeException

make

```
void make()
```

make
Throws:
> RhapsodyRuntimeException

mergeElements

```
void mergeElements(IRPModelElement left,
                   IRPModelElement right)
```

mergeElements
Throws:
        RhapsodyRuntimeException

newProjectOnDesignManager

```
IRPProject newProjectOnDesignManager(java.lang.String userName,
                                     java.lang.String password,
                                     java.lang.String serverURL,
                                     java.lang.String projectAreaName,
                                     java.lang.String streamName,
                                     java.lang.String modelName)
```

Creates a new project on the Design Manager server.
Parameters:
        userName - the username to use to log in to the DM server
        password - the password to use to log in to the DM server
        serverURL - the URL of the DM server
        projectAreaName - the project area where the project should be created
        streamName - the stream in which the project should be created
        modelName - the name to use for the project
Returns:
        the project that was created

notifyFileChanged

```
void notifyFileChanged(java.lang.String filename,
                       int contentChanged)
```

File change notification
Throws:
        RhapsodyRuntimeException

openAdvancedSearchAndReplaceDialog

```
void openAdvancedSearchAndReplaceDialog()
```

Open Advanced Search and Replace dialog
Throws:
        RhapsodyRuntimeException

openFileList

void **openFileList**(java.lang.String filename)

> method openFileList
> **Throws:**
> > RhapsodyRuntimeException

openProject

IRPProject **openProject**(java.lang.String filename)

> Opens an existing Rhapsody project
> **Parameters:**
> > filename - the name of the .rpy file, including the full path, for example,
> > "l:\\temp_sample_code\\Class_Tricks.rpy"
> **Returns:**
> > IRPProject object that represents the Rhapsody project

openProjectFromDesignManager

IRPProject **openProjectFromDesignManager**(java.lang.String userName,
                                                java.lang.String password,
                                                java.lang.String serverURL,
                                                java.lang.String projectAreaName,
                                                java.lang.String streamName,
                                                java.lang.String modelName)

> openProjectFromDesignManager
> **Throws:**
> > RhapsodyRuntimeException

openProjectFromDesignManagerAfterLogin

IRPProject **openProjectFromDesignManagerAfterLogin**(java.lang.String serverURL,
                                                java.lang.String projectAreaName,
                                                java.lang.String streamName,
                                                java.lang.String modelName)

> openProjectFromDesignManagerAfterLogin
> **Throws:**
> > RhapsodyRuntimeException

openProjectFromURL

void **openProjectFromURL**(java.lang.String purl)

> open Project From URL

openFileList   68

Throws:
    [RhapsodyRuntimeException](#)

---

openProjectWithLastSession

[IRPProject](#) **openProjectWithLastSession**(java.lang.String filename)

open project with last session
Throws:
    [RhapsodyRuntimeException](#)

---

openProjectWithoutSubUnits

[IRPProject](#) **openProjectWithoutSubUnits**(java.lang.String filename)

open project without subunits
Throws:
    [RhapsodyRuntimeException](#)

---

quit

void **quit**()

quit application
Throws:
    [RhapsodyRuntimeException](#)

---

rebuild

void **rebuild**()

rebuild
Throws:
    [RhapsodyRuntimeException](#)

---

rebuildEntireProject

void **rebuildEntireProject**()

rebuildEntireProject
Throws:
    [RhapsodyRuntimeException](#)

---

rebuildWithDependencies

void **rebuildWithDependencies**()

> rebuildWithDependencies
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

redo

int **redo**()

> Perform Redo
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

refactorSelectedOperation

void **refactorSelectedOperation**(java.lang.String newName)

> Changes the name of the currently-selected operation and updates any references to the operation accordingly. Corresponds to the Refactor option in the pop-up menu for operations.
> **Parameters:**
> > newName - the new name to use for the operation

---

refreshAllViews

void **refreshAllViews**()

> refresh all views
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

regenerate

void **regenerate**()

> regenerate
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

regenerateElements

void **regenerateElements**([IRPCollection](#) elements)

method regenerateElements
Throws:
> [RhapsodyRuntimeException](#)

regenerateEntireProject

void **regenerateEntireProject**()

> regenerateEntireProject
> **Throws:**
>> [RhapsodyRuntimeException](#)

regenerateWithDependencies

void **regenerateWithDependencies**()

> regenerateWithDependencies
> **Throws:**
>> [RhapsodyRuntimeException](#)

registerCOMClient

void **registerCOMClient**(int processID,
                        java.lang.String clientFilename,
                        int magicNumber)

> register COM client
> **Throws:**
>> [RhapsodyRuntimeException](#)

report

void **report**(java.lang.String format,
            java.lang.String outputFileName)

> report
> **Throws:**
>> [RhapsodyRuntimeException](#)

roundtrip

void **roundtrip**()

> roundtrip
> **Throws:**
>> [RhapsodyRuntimeException](#)

roundtripElements

void **roundtripElements**([IRPCollection](#) elements)

>   method roundtripElements
>   **Throws:**
>   [RhapsodyRuntimeException](#)

runApplication

void **runApplication**()

>   Runs the application that was built for the project

runHelper

void **runHelper**(java.lang.String helperName)

>   runHelper
>   **Throws:**
>   [RhapsodyRuntimeException](#)

runHelperWithParameters

void **runHelperWithParameters**(java.lang.String helperName,
                          java.lang.String params)

>   runHelperWithParameters
>   **Throws:**
>   [RhapsodyRuntimeException](#)

saveAll

void **saveAll**()

>   method saveAll
>   **Throws:**
>   [RhapsodyRuntimeException](#)

selectGraphElements

void **selectGraphElements**([IRPCollection](#) graphElements)

Selects multiple elements in the most recently opened diagram.
Parameters:
>   graphElements - collection of the graphical elements that should be selected

---

selectModelElements

void **selectModelElements**([IRPCollection](#) modelElements)

Selects multiple items in the model browser.
Parameters:
>   modelElements - collection of the model elements that should be selected

---

setApplicationStatus

void **setApplicationStatus**(int nStatus)

setApplicationStatus
Throws:
>   [RhapsodyRuntimeException](#)

---

setComponent

void **setComponent**(java.lang.String component)

set Component
Throws:
>   [RhapsodyRuntimeException](#)

---

setConfiguration

void **setConfiguration**(java.lang.String configuration)

set Configuration
Throws:
>   [RhapsodyRuntimeException](#)

---

setDMBoolProperty

void **setDMBoolProperty**(java.lang.String pKey,
                          int val)

Sets the value of the specified boolean Design Manager property
Parameters:
>   pKey - the name of the boolean property whose value should be set
>   val - the value to use for the property - use 0 for False, 1 for True

---

selectGraphElements   73

setDMProperty

```
void setDMProperty(java.lang.String pKey,
                   java.lang.String val)
```

> Sets the value of the specified Design Manager property
> **Parameters:**
> > pKey - the name of the property whose value should be set
> > val - the value to use for the property

setHiddenUI

```
void setHiddenUI(boolean pVal)
```

> set property hiddenUI
> **Throws:**
> > RhapsodyRuntimeException

setIsLoadOnDemand

```
void setIsLoadOnDemand(int isLoadOnDemand)
```

> set property isLoadOnDemand
> **Throws:**
> > RhapsodyRuntimeException

setLanguage

```
void setLanguage(java.lang.String language)
```

> set property Language
> **Throws:**
> > RhapsodyRuntimeException

setLog

```
void setLog(java.lang.String logFile)
```

> set log file
> **Throws:**
> > RhapsodyRuntimeException

setToolSet

void **setToolSet**(java.lang.String toolSet)

> set property ToolSet
> **Throws:**
> > RhapsodyRuntimeException

---

setUpdateRecentFileList

void **setUpdateRecentFileList**(int shouldUpdate)

> setUpdateRecentFileList
> **Throws:**
> > RhapsodyRuntimeException

---

startUndoTransaction

void **startUndoTransaction**()

> start undo transaction
> **Throws:**
> > RhapsodyRuntimeException

---

syncBuild

void **syncBuild**()

> syncBuild
> **Throws:**
> > RhapsodyRuntimeException

---

terminateApplication

void **terminateApplication**()

> Terminate the Application
> **Throws:**
> > RhapsodyRuntimeException

---

undo

int **undo**()

> Perform Undo
> **Throws:**

unloadFromTarget

void **unloadFromTarget**()

> unloadFromTarget
> **Throws:**
> > RhapsodyRuntimeException

unregisterCOMClient

void **unregisterCOMClient**(int processID,
                            java.lang.String clientFilename,
                            int magicNumber)

> unregister COM client
> **Throws:**
> > RhapsodyRuntimeException

version

java.lang.String **version**()

> Get Rhapsody version
> **Throws:**
> > RhapsodyRuntimeException

versionNumberLong

java.lang.String **versionNumberLong**()

> Get Rhapsody versionNumberLong
> **Throws:**
> > RhapsodyRuntimeException

writeToOutputWindow

void **writeToOutputWindow**(java.lang.String title,
                            java.lang.String outputStr)

> Writes text to Rhapsody's Output window.
> **Parameters:**
> > title - the name of the tab to which the text should be written. The possible values are Log,
> > Build, Configuration Management, Animation.
> > outputStr - the text to display in the Output window

com.telelogic.rhapsody.core
Interface IRPArgument

All Superinterfaces:
> IRPModelElement, IRPUnit, IRPVariable

```
public interface IRPArgument
```
extends <u>IRPVariable</u>

The IRPArgument interface represents an argument of an operation or an event.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
<u>IRPModelElement.OSLCLink</u>

Method Summary

java.lang.String	**getArgumentDirection**() Returns the direction of the argument (In, Out, or InOut).
void	**setArgumentDirection**(java.lang.String argumentDirection) Sets the direction of the argument.

Methods inherited from interface com.telelogic.rhapsody.core.IRPVariable
<u>addElementDefaultValue</u>, <u>addStringDefaultValue</u>, <u>getDeclaration</u>, <u>getDefaultValue</u>, <u>getType</u>, <u>getValueSpecifications</u>, <u>setDeclaration</u>, <u>setDefaultValue</u>, <u>setType</u>, <u>setTypeDeclaration</u>

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit
<u>copyToAnotherProject</u>, <u>getAddToModelMode</u>, <u>getCMHeader</u>, <u>getCMState</u>, <u>getCurrentDirectory</u>, <u>getFilename</u>, <u>getIncludeInNextLoad</u>, <u>getIsStub</u>, <u>getLanguage</u>, <u>getLastModifiedTime</u>, <u>getNestedSaveUnits</u>, <u>getNestedSaveUnitsCount</u>, <u>getStructureDiagrams</u>, <u>isReadOnly</u>, <u>isReferenceUnit</u>, <u>isSeparateSaveUnit</u>, <u>load</u>, <u>moveToAnotherProjectLeaveAReference</u>, <u>referenceToAnotherProject</u>, <u>save</u>, <u>setCMHeader</u>, <u>setFilename</u>, <u>setIncludeInNextLoad</u>, <u>setLanguage</u>, <u>setReadOnly</u>, <u>setSeparateSaveUnit</u>, <u>setUnitPath</u>, <u>unload</u>

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getArgumentDirection

java.lang.String **getArgumentDirection**()

> Returns the direction of the argument (In, Out, or InOut).
> **Returns:**
> > the direction of the argument

setArgumentDirection

void **setArgumentDirection**(java.lang.String argumentDirection)

> Sets the direction of the argument.
> **Parameters:**
> > argumentDirection - the direction to use for the argument. The valid strings are In, Out, and InOut.

com.telelogic.rhapsody.core

Package Class Use Tree Serialized Deprecated Index Help

PREV CLASS NEXT CLASS   **FRAMES NO FRAMES   All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

 com.telelogic.rhapsody.core
Interface IRPASCIIFile

```
public interface IRPASCIIFile
```

Method Summary

void	**close**() close file
java.lang.String	**getInterfaceName**() get property interfaceName
void	**open**(java.lang.String filename) open file
void	**write**(java.lang.String data) write to file

Method Detail

close

```
void close()
```

close file
Throws:
RhapsodyRuntimeException

getInterfaceName

```
java.lang.String getInterfaceName()
```

get property interfaceName
Throws:
RhapsodyRuntimeException

open

```
void open(java.lang.String filename)
```

> open file
> **Throws:**
> > RhapsodyRuntimeException

write

```
void write(java.lang.String data)
```

> write to file
> **Throws:**
> > RhapsodyRuntimeException

Package | **Class** | **Use** | **Tree** | **Serialized** | **Deprecated** | **Index** | **Help**

PREV CLASS | **NEXT CLASS**   **FRAMES** | **NO FRAMES** | **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

com.telelogic.rhapsody.core
Interface IRPAssociationClass

All Superinterfaces:
      IRPClass, IRPClassifier, IRPModelElement, IRPUnit

```
public interface IRPAssociationClass
```
extends IRPClass

The IRPAssociationClass interface represents association classes in Rational Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPRelation	**getEnd1**() Gets the relation represented by the first end of the association class.
IRPRelation	**getEnd2**() Gets the relation represented by the second end of the association class.
int	**getIsClass**() Checks whether the element is an association class or an association element.
void	**setIsClass**(int isClass) Specifies whether the element should be an association class or an association element.

Methods inherited from interface com.telelogic.rhapsody.core.IRPClass
addClass, addConstructor, addDestructor, addEventReception, addLink, addLinkToPartViaPort, addReception, addSuperclass, addTriggeredOperation, addType, deleteClass, deleteConstructor, deleteDestructor, deleteEventReception, deleteReception, deleteSuperclass, deleteType, getIsAbstract, getIsActive, getIsBehaviorOverriden, getIsComposite, getIsFinal, getIsReactive, setIsAbstract, setIsActive, setIsBehaviorOverriden, setIsFinal

Methods inherited from interface com.telelogic.rhapsody.core.IRPClassifier

Methods inherited from interface com.telelogic.rhapsody.core.IRPClassifier

addActivityDiagram, addAttribute, addFlowItems, addFlows, addGeneralization, addOperation, addRelation, addRelationTo, addStatechart, addUnidirectionalRelation, addUnidirectionalRelationTo, deleteAttribute, deleteFlowItems, deleteFlows, deleteGeneralization, deleteOperation, deleteRelation, findAttribute, findBaseClassifier, findDerivedClassifier, findGeneralization, findInterfaceItem, findNestedClassifier, findNestedClassifierRecursive, findRelation, findTrigger, getActivityDiagram, getAttributes, getAttributesIncludingBases, getBaseClassifiers, getBehavioralDiagrams, getDerivedClassifiers, getFlowItems, getFlows, getGeneralizations, getInterfaceItems, getInterfaceItemsIncludingBases, getLinks, getNestedClassifiers, getOperations, getPorts, getRelations, getRelationsIncludingBases, getSequenceDiagrams, getSourceArtifacts, getStatechart

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getEnd1

[IRPRelation](#) **getEnd1**()

> Gets the relation represented by the first end of the association class.
> **Returns:**
>> the relation represented by the first end of the association class

---

getEnd2

[IRPRelation](#) **getEnd2**()

> Gets the relation represented by the second end of the association class.
> **Returns:**
>> the relation represented by the second end of the association class

---

getIsClass

int **getIsClass**()

> Checks whether the element is an association class or an association element.
> **Returns:**
>> 1 if the element is an association class, 0 if it is an association element

---

setIsClass

void **setIsClass**(int isClass)

> Specifies whether the element should be an association class or an association element.
> **Parameters:**
>> isClass - Use 1 to specify that the element should be an association class. Use 0 to specify that the element should be an association element.

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPAssociationRole

All Superinterfaces:
> IRPModelElement

---

public interface **IRPAssociationRole**
extends IRPModelElement

The IRPAssociationRole interface represents the association roles that link objects in communication diagrams.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPCollection	**getClassifierRoles**() Returns a collection of the classifier roles that are linked by the association role.
IRPCollection	**getFormalRelations**() Returns a collection of IRPRelation objects, representing the association ends of the association role.
java.lang.String	**getRoleType**() For internal use only.

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram,

setIsClass   86

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getClassifierRoles

IRPCollection **getClassifierRoles**()

> Returns a collection of the classifier roles that are linked by the association role.
> **Returns:**
> > the classifier roles that are linked by the association role

getFormalRelations

IRPCollection **getFormalRelations**()

> Returns a collection of IRPRelation objects, representing the association ends of the association role.
> **Returns:**
> > the association ends of the association role

getRoleType

java.lang.String **getRoleType**()

> For internal use only.

Package   **Class**   **Use**   **Tree**   **Serialized**   **Deprecated**   **Index**   **Help**

PREV CLASS   **NEXT CLASS**   **FRAMES**   **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.coreInterface IRPAssociationRole   87

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | <u>METHOD</u>   DETAIL: FIELD | CONSTR | <u>METHOD</u>

com.telelogic.rhapsody.core
Interface IRPAttribute

All Superinterfaces:
> IRPModelElement, IRPUnit, IRPVariable

---

public interface **IRPAttribute**
extends IRPVariable

The IRPAttribute interface represents attributes of a class, and global variables. To create a new attribute, use the method IRPClassifier.addAttribute. To create a new variable, use the method IRPPackage.addGlobalVariable.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

int	**getIsConstant**() Checks whether the attribute was defined as constant.
int	**getIsOrdered**() For attributes with multiplicity greater than one, checks whether the order of the items was specified as significant.
int	**getIsReference**() Checks whether the attribute was defined as a pointer.
int	**getIsStatic**() Checks whether the attribute was defined as static.
java.lang.String	**getMultiplicity**() Gets the multiplicity specified for the attribute.
java.lang.String	**getVisibility**() Gets the visibility specified for the attribute.
void	**setIsConstant**(int isConstant) Specifies whether an attribute should be defined as constant.

Method Summary

void	**setIsOrdered**(int isOrdered) For attributes with multiplicity greater than one, this method is used to specify whether the attribute should be defined as ordered, meaning that the order of the items is significant.
void	**setIsReference**(int isReference) Specifies whether an attribute should be defined as a pointer.
void	**setIsStatic**(int isStatic) Specifies whether an attribute should be defined as static.
void	**setMultiplicity**(java.lang.String multiplicity) Specifies the multiplicity for the attribute.
void	**setVisibility**(java.lang.String visibility) Specifies the visibility of the operation.

Methods inherited from interface com.telelogic.rhapsody.core.IRPVariable

addElementDefaultValue, addStringDefaultValue, getDeclaration, getDefaultValue, getType, getValueSpecifications, setDeclaration, setDefaultValue, setType, setTypeDeclaration

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser,

com.telelogic.rhapsody.core

Method Detail

getIsConstant

int **getIsConstant**()

Checks whether the attribute was defined as constant.
Returns:
1 if the attribute was defined as constant, 0 otherwise

getIsOrdered

int **getIsOrdered**()

For attributes with multiplicity greater than one, checks whether the order of the items was specified as significant.
Returns:
1 if the attribute was defined as ordered, 0 otherwise

getIsReference

int **getIsReference**()

Checks whether the attribute was defined as a pointer.
Returns:
1 if the attribute was defined as pointer, 0 otherwise

getIsStatic

int **getIsStatic**()

Checks whether the attribute was defined as static.
Returns:
1 if the attribute was defined as static, 0 otherwise

getMultiplicity

`java.lang.String **getMultiplicity**()`

Gets the multiplicity specified for the attribute.

Returns:

the multiplicity specified for the attribute

---

getVisibility

`java.lang.String **getVisibility**()`

Gets the visibility specified for the attribute.

Returns:

the visibility specified for the attribute

---

setIsConstant

`void **setIsConstant**(int isConstant)`

Specifies whether an attribute should be defined as constant.

Parameters:

isConstant - Use 1 to specify that the attribute should be defined as constant. Use 0 to specify that the attribute should not be defined as constant.

---

setIsOrdered

`void **setIsOrdered**(int isOrdered)`

For attributes with multiplicity greater than one, this method is used to specify whether the attribute should be defined as ordered, meaning that the order of the items is significant.

Parameters:

isOrdered - Use 1 to specify that the attribute should be defined as ordered. Use 0 to specify that the attribute should not be defined as ordered.

---

setIsReference

`void **setIsReference**(int isReference)`

Specifies whether an attribute should be defined as a pointer.

Parameters:

isReference - Use 1 to specify that the attribute should be defined as a pointer. Use 0 to specify that the attribute should not be defined as a pointer.

---

setIsStatic

void **setIsStatic**(int isStatic)

Specifies whether an attribute should be defined as static.
Parameters:
isStatic - Use 1 to specify that the attribute should be defined as static. Use 0 to specify that the attribute should not be defined as static.

setMultiplicity

void **setMultiplicity**(java.lang.String multiplicity)

Specifies the multiplicity for the attribute.
Parameters:
multiplicity - the multiplicity to use for the attribute. You can use strings such as "1" or "14" to specify a specific number, or you can use one of the values listed in the Features dialog for attributes: "0,1", "*", or "1..*". If you are setting the multiplicity to a value greater than one, use the setIsOrdered method to specify whether the order of the items is significant.

setVisibility

void **setVisibility**(java.lang.String visibility)

Specifies the visibility of the operation.
Parameters:
visibility - the visibility to use for the operation. The possible values are "public", "private", and "protected". For C# projects, you can also use the value "project" for internal operations, and the value "projectOrProtected" for protected internal operations

com.telelogic.rhapsody.core
Interface IRPAXViewCtrl

```
public interface IRPAXViewCtrl
```

Method Summary

void	**doCommand**(long commandID) Execute command by command id
void	**executeCommand**(java.lang.String commandType, IRPCollection pCommandInitialization, IRPCollection pCommandResult) Execute command
java.lang.String	**getInterfaceName**() get property interfaceName

Method Detail

doCommand

```
void doCommand(long commandID)
```

> Execute command by command id
> **Throws:**
>   RhapsodyRuntimeException

---

executeCommand

```
void executeCommand(java.lang.String commandType,
                    IRPCollection pCommandInitialization,
                    IRPCollection pCommandResult)
```

> Execute command
> **Throws:**
>   RhapsodyRuntimeException

---

getInterfaceName

`java.lang.String` **`getInterfaceName`**`()`

get property interfaceName

Throws:

[RhapsodyRuntimeException](#)

[Package](#)	**Class**	**[Use](#)**	**[Tree](#)**	**[Serialized](#)**	**[Deprecated](#)**	**[Index](#)**	**[Help](#)**

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPBaseExternalCodeGeneratorTool

All Known Subinterfaces:
>   IRPCodeGenSimplifiersRegistry, IRPExternalCodeGeneratorInvoker

---

public interface **IRPBaseExternalCodeGeneratorTool**

Method Summary

void	**advanceCodeGenProgressBar**() method advanceCodeGenProgressBar
int	**shouldAbortCodeGeneration**() method shouldAbortCodeGeneration
void	**writeCodeGenMessage**(java.lang.String msg) method writeCodeGenMessage

Method Detail

advanceCodeGenProgressBar

void **advanceCodeGenProgressBar**()

>   method advanceCodeGenProgressBar
>   **Throws:**
> >   RhapsodyRuntimeException

---

shouldAbortCodeGeneration

int **shouldAbortCodeGeneration**()

>   method shouldAbortCodeGeneration
>   **Throws:**
> >   RhapsodyRuntimeException

---

writeCodeGenMessage

void **writeCodeGenMessage**(java.lang.String msg)

> method writeCodeGenMessage
> **Throws:**
>> RhapsodyRuntimeException

com.telelogic.rhapsody.core
Interface IRPCallOperation

All Superinterfaces:
> IRPModelElement, IRPState, IRPStateVertex

---

public interface **IRPCallOperation**
extends IRPState

The IRPCallOperation interface represents call operation elements in activity diagrams.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPInterfaceItem	**getOperation**() Returns the operation specified for this call operation element.
IRPRelation	**getTarget**() Returns the target specified for this call operation element.
void	**setOperation**(IRPInterfaceItem operation) Specifies the operation to use for this call operation element.
void	**setTarget**(IRPRelation target) Specifies the target to use for this call operation element.

Methods inherited from interface com.telelogic.rhapsody.core.IRPState
addActivityFinal, addConnector, addInternalTransition, addState, addStaticReaction, addTerminationState, createDefaultTransition, createNestedStatechart, deleteConnector, deleteInternalTransition, deleteStaticReaction, getDefaultTransition, getEntryAction, getExitAction, getFullNameInStatechart, getInheritsFrom, getInternalTransitions, getIsOverridden, getIsReferenceActivity, getItsStatechart, getItsSwimlane, getLogicalStates, getNestedStatechart, getReferenceToActivity, getSendAction, getStateType, getStaticReactions, getSubStates, getSubStateVertices, getTheEntryAction, getTheExitAction, isAnd, isCompound, isLeaf, isRoot, isSendActionState, overrideInheritance, resetEntryActionInheritance, resetExitActionInheritance, setEntryAction, setExitAction, setInternalTransition, setItsSwimlane,

Methods inherited from interface com.telelogic.rhapsody.core.**IRPState**
setReferenceToActivity, setStateType, setStaticReaction, unoverrideInheritance

Methods inherited from interface com.telelogic.rhapsody.core.**IRPStateVertex**
addFlow, addTransition, deleteTransition, getInTransitions, getOutTransitions, getParent, setParent

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getOperation

IRPInterfaceItem **getOperation**()

> Returns the operation specified for this call operation element.
> **Returns:**
> > the operation specified for this call operation element

getTarget

<u>IRPRelation</u> **getTarget**()

>> Returns the target specified for this call operation element.
>> **Returns:**
>>> the target specified for this call operation element

---

setOperation

void **setOperation**(<u>IRPInterfaceItem</u> operation)

>> Specifies the operation to use for this call operation element.
>> **Parameters:**
>>> operation - the operation to use for this call operation element

---

setTarget

void **setTarget**(<u>IRPRelation</u> target)

>> Specifies the target to use for this call operation element.
>> **Parameters:**
>>> target - the target to use for this call operation element

---

Package   **Class**   **Use**   **Tree**   **Serialized**   **Deprecated**   **Index**   **Help**

PREV CLASS   **NEXT CLASS**   **FRAMES**   **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | <u>METHOD</u>   DETAIL: FIELD | CONSTR | <u>METHOD</u>

com.telelogic.rhapsody.core
Interface IRPClass

All Superinterfaces:
>   IRPClassifier, IRPModelElement, IRPUnit

All Known Subinterfaces:
>   IRPAssociationClass, IRPFlowchart, IRPStatechart

```
public interface IRPClass
```
extends IRPClassifier

The IRPClass interface represents classes in Rational Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPClass	**addClass**(java.lang.String name) Adds a class to the current class.
IRPOperation	**addConstructor**(java.lang.String argumentsData) Adds a constructor for the current class.
IRPOperation	**addDestructor**() Adds a destructor for the current class.
IRPEventReception	**addEventReception**(java.lang.String name) Adds an event reception to the current class.
IRPLink	**addLink**(IRPInstance fromPart, IRPInstance toPart, IRPRelation assoc, IRPPort fromPort, IRPPort toPort) This method is used to create a link between two parts belonging to a class.
IRPLink	**addLinkToPartViaPort**(IRPInstance toPart, IRPInstance partPort, IRPInstance classPort, IRPRelation assoc) This method is used to create a delegation connector between a class and one of its parts.

Method Summary

IRPEventReception	**addReception**(java.lang.String name) Adds a reception to the current class.
void	**addSuperclass**(IRPClass superClass) Specifies a base class that the current class should inherit from.
IRPOperation	**addTriggeredOperation**(java.lang.String name) Adds a new triggered operation to the current class.
IRPType	**addType**(java.lang.String name) Adds a new type to the current class.
void	**deleteClass**(java.lang.String name) Deletes the specified class from the current class.
void	**deleteConstructor**(IRPOperation constructor) Deletes the specified constructor from the current class.
void	**deleteDestructor**() Deletes the destructor for the class.
void	**deleteEventReception**(IRPEventReception pVal) Deletes the specified event reception.
void	**deleteReception**(IRPEventReception pVal) Deletes the specified reception from the current class.
void	**deleteSuperclass**(IRPClass superClass) Removes the inheritance relationship with the specified base class.
void	**deleteType**(java.lang.String name) Deletes the specified type from the current class.
int	**getIsAbstract**() Checks whether the class is an abstract class.
int	**getIsActive**() Checks whether the class was defined as "active", meaning that during execution it runs on its own thread.
int	**getIsBehaviorOverriden**() Checks whether a class does not inherit the behavior defined in the statechart of its base class.
int	**getIsComposite**() Checks whether the class is a composite class.
int	**getIsFinal**() Checks whether the class is a final class.
int	**getIsReactive**() Checks whether the class is a reactive class, meaning that a statechart or an activity diagram has been created for the class so that it reacts to events.
void	**setIsAbstract**(int isAbstract) Specifies that the class should be abstract.

Method Summary

void	**setIsActive**(int isActive) Specifies that the class should be defined as "active", meaning that during execution it runs on its own thread.
void	**setIsBehaviorOverriden**(int isBehaviorOverriden) Specifies whether a class should inherit the behavior defined in the statechart of its base class.
void	**setIsFinal**(int newVal) Specifies that the class should be a final class.

Methods inherited from interface com.telelogic.rhapsody.core.IRPClassifier

addActivityDiagram, addAttribute, addFlowItems, addFlows, addGeneralization, addOperation, addRelation, addRelationTo, addStatechart, addUnidirectionalRelation, addUnidirectionalRelationTo, deleteAttribute, deleteFlowItems, deleteFlows, deleteGeneralization, deleteOperation, deleteRelation, findAttribute, findBaseClassifier, findDerivedClassifier, findGeneralization, findInterfaceItem, findNestedClassifier, findNestedClassifierRecursive, findRelation, findTrigger, getActivityDiagram, getAttributes, getAttributesIncludingBases, getBaseClassifiers, getBehavioralDiagrams, getDerivedClassifiers, getFlowItems, getFlows, getGeneralizations, getInterfaceItems, getInterfaceItemsIncludingBases, getLinks, getNestedClassifiers, getOperations, getPorts, getRelations, getRelationsIncludingBases, getSequenceDiagrams, getSourceArtifacts, getStatechart

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML,

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addClass

IRPClass **addClass**(java.lang.String name)

> Adds a class to the current class.
> **Parameters:**
> > name - the name to use for the new class
> **Returns:**
> > the new class created

addConstructor

IRPOperation **addConstructor**(java.lang.String argumentsData)

> Adds a constructor for the current class.
> **Parameters:**
> > argumentsData - the name and types of the arguments for the constructor. The string should use the format "name1,type1,name2,type2", for example "a,int,b,int". For a constructor that does not take arguments, use an empty string ("").
> **Returns:**
> > the constructor created

addDestructor

IRPOperation **addDestructor**()

> Adds a destructor for the current class.
> **Returns:**
> > the destructor created

addEventReception

<u>IRPEventReception</u> **addEventReception**(java.lang.String name)

Adds an event reception to the current class. It is preferable that you use the operation IRPClass.addReception instead.

Parameters:
name - the name to use for the new event reception
Returns:
the event reception created

addLink

<u>IRPLink</u> **addLink**(<u>IRPInstance</u> fromPart,
                 <u>IRPInstance</u> toPart,
                 <u>IRPRelation</u> assoc,
                 <u>IRPPort</u> fromPort,
                 <u>IRPPort</u> toPort)

This method is used to create a link between two parts belonging to a class. In addition to specifying the two parts, you must specify the association that the link should represent, or, alternatively, the two ports that should be used for the link. If you provide the two ports as arguments, you should use Null for the association argument. Similarly, if you specify an association, you should use Null for the two port arguments. Note that if you are not specifying the two ports, you must provide an association as an argument even if there is only one relevant association.

Parameters:
fromPart - the "from" part for the link
toPart - the "to" part for the link
assoc - the association that the link should represent
fromPort - the "from" port for the link
toPort - the "to" port for the link
Returns:
the link created

addLinkToPartViaPort

<u>IRPLink</u> **addLinkToPartViaPort**(<u>IRPInstance</u> toPart,
                           <u>IRPInstance</u> partPort,
                           <u>IRPInstance</u> classPort,
                           <u>IRPRelation</u> assoc)

This method is used to create a delegation connector between a class and one of its parts. In addition to specifying the part to use, you must specify the association that the link should represent, or, alternatively, the two ports that should be used for the link. If you provide the two ports as arguments, you should use Null for the association argument. Similarly, if you specify an association, you should use Null for the two port arguments. Note that if you are not specifying the two ports, you must provide an association as an argument even if there is only one relevant association.

Parameters:
toPart - the part that should be linked to

> partPort - the port to use on the part
>
> classPort - the port to use on the class
>
> assoc - the association that the link should represent

Returns:

> the link created

---

addReception

<u>IRPEventReception</u> **addReception**(java.lang.String name)

> Adds a reception to the current class.
>
> **Parameters:**
>
> > name - the name to use for the new reception
>
> **Returns:**
>
> > the reception created

---

addSuperclass

void **addSuperclass**(<u>IRPClass</u> superClass)

> Specifies a base class that the current class should inherit from.
>
> **Parameters:**
>
> > superClass - the name of the class that should serve as the base class

---

addTriggeredOperation

<u>IRPOperation</u> **addTriggeredOperation**(java.lang.String name)

> Adds a new triggered operation to the current class.
>
> **Parameters:**
>
> > name - the name to use for the new triggered operation
>
> **Returns:**
>
> > the triggered operation created

---

addType

<u>IRPType</u> **addType**(java.lang.String name)

> Adds a new type to the current class.
>
> **Parameters:**
>
> > name - the name to use for the new type
>
> **Returns:**
>
> > the type created

---

deleteClass

void **deleteClass**(java.lang.String name)

Deletes the specified class from the current class.
Parameters:
   name - the name of the class that should be deleted

deleteConstructor

void **deleteConstructor**([IRPOperation](#) constructor)

Deletes the specified constructor from the current class.
Parameters:
   constructor - the constructor that should be deleted. Note that this parameter is of type IRPOperation.

deleteDestructor

void **deleteDestructor**()

Deletes the destructor for the class.

deleteEventReception

void **deleteEventReception**([IRPEventReception](#) pVal)

Deletes the specified event reception. It is preferable that you use the operation IRPClass.deleteReception instead.
Parameters:
   pVal - the reception that should be deleted

deleteReception

void **deleteReception**([IRPEventReception](#) pVal)

Deletes the specified reception from the current class.
Parameters:
   pVal - the reception that should be deleted

deleteSuperclass

void **deleteSuperclass**([IRPClass](#) superClass)

Removes the inheritance relationship with the specified base class.

Parameters:

`superClass` - the base class of the current class.

deleteType

void **deleteType**(java.lang.String name)

Deletes the specified type from the current class.
Parameters:

`name` - the name of the type that should be deleted

getIsAbstract

int **getIsAbstract**()

Checks whether the class is an abstract class.
Returns:

indication of whether the class is abstract - 1 if the class is abstract, 0 if not

getIsActive

int **getIsActive**()

Checks whether the class was defined as "active", meaning that during execution it runs on its own thread.
Returns:

indication of whether the class was defined as "active". 1 means that the class is "active", 0 means that the class was defined as "sequential".

getIsBehaviorOverriden

int **getIsBehaviorOverriden**()

Checks whether a class does not inherit the behavior defined in the statechart of its base class. When you create a statechart for a class, by default it inherits the behavior defined in the statechart of its base class. However, Rhapsody allows you to specify that the class should not inherit this behavior. This operation checks whether this option has been exercised for the current class.
Returns:

indication of whether or not the class inherits the behavior specified in the statechart of its base class. 1 means that it does not inherit this behavior, 0 means that it does inherit the behavior defined in the statechart of the base class.

getIsComposite

int **getIsComposite**()

>
> Checks whether the class is a composite class.
> **Returns:**
>> indication of whether the class is a composite class. 1 means that the class is a composite class, 0 means it is not.

getIsFinal

int **getIsFinal**()

>
> Checks whether the class is a final class. Relevant only for Java classes.
> **Returns:**
>> indication of whether the class is a final class - 1 if the class is final, 0 if not

getIsReactive

int **getIsReactive**()

>
> Checks whether the class is a reactive class, meaning that a statechart or an activity diagram has been created for the class so that it reacts to events.
> **Returns:**
>> indication of whether the class is a reactive class. 1 means that the class is a reactive class, 0 means it is not.

setIsAbstract

void **setIsAbstract**(int isAbstract)

>
> Specifies that the class should be abstract.
> **Parameters:**
>> isAbstract - use 1 to specify that the class should be an abstract class, use 0 to specify that it should not be abstract

setIsActive

void **setIsActive**(int isActive)

>
> Specifies that the class should be defined as "active", meaning that during execution it runs on its own thread.
> **Parameters:**
>> isActive - 1 means that the class will be defined as "active", 0 means that the class will be defined as "sequential"

setIsBehaviorOverriden

void **setIsBehaviorOverriden**(int isBehaviorOverriden)

> Specifies whether a class should inherit the behavior defined in the statechart of its base class. When you create a statechart for a class, by default it inherits the behavior defined in the statechart of its base class. However, Rhapsody allows you to specify that the class should not inherit this behavior.
> **Parameters:**
>> isBehaviorOverriden - use 1 if you do not want the class to inherit the behavior defined in the statechart of its base class. Use 0 if you want the class to inherit this behavior.

setIsFinal

void **setIsFinal**(int newVal)

> Specifies that the class should be a final class. Relevant only for Java classes.
> **Parameters:**
>> newVal - use 1 to specify that the class should be a final class, use 0 to specify that it should not be final

 com.telelogic.rhapsody.core
Interface IRPClassifier

All Superinterfaces:
> IRPModelElement, IRPUnit

All Known Subinterfaces:
> IRPActor, IRPAssociationClass, IRPClass, IRPEvent, IRPEventReception, IRPFlowchart, IRPFlowItem, IRPInterfaceItem, IRPNode, IRPOperation, IRPStatechart, IRPStereotype, IRPType, IRPUseCase

---

public interface **IRPClassifier**
extends IRPUnit

Represents the features shared by elements such as classes, actors, use cases, and types.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPFlowchart	**addActivityDiagram**() Creates a new activity diagram.
IRPAttribute	**addAttribute**(java.lang.String name) Adds a new attribute to the classifier.
IRPFlowItem	**addFlowItems**(java.lang.String name) Adds a new item flow to the classifier.
IRPFlow	**addFlows**(java.lang.String name) Adds a new flow to the classifier.
void	**addGeneralization**(IRPClassifier pVal) Adds a generalization relationship between the classifier and the classifier specified as a parameter.
IRPOperation	**addOperation**(java.lang.String name) Adds a new operation.

---

setIsFinal   111

Method Summary

IRPRelation	**addRelation**(java.lang.String otherClassName, java.lang.String otherClassPackageName, java.lang.String roleName1, java.lang.String linkType1, java.lang.String multiplicity1, java.lang.String roleName2, java.lang.String linkType2, java.lang.String multiplicity2, java.lang.String linkName) Adds a new association to the classifier.
IRPRelation	**addRelationTo**(IRPClassifier otherClassifier, java.lang.String roleName1, java.lang.String linkType1, java.lang.String multiplicity1, java.lang.String roleName2, java.lang.String linkType2, java.lang.String multiplicity2, java.lang.String linkName) Adds a new association to the classifier.
IRPStatechart	**addStatechart**() Creates a new statechart.
IRPRelation	**addUnidirectionalRelation**(java.lang.String otherClassName, java.lang.String otherClassPackageName, java.lang.String roleName, java.lang.String linkType, java.lang.String multiplicity, java.lang.String linkName) Adds a new directed association to the classifier.
IRPRelation	**addUnidirectionalRelationTo**(IRPClassifier otherClassifier, java.lang.String roleName, java.lang.String linkType, java.lang.String multiplicity, java.lang.String linkName) Adds a new directed association to the classifier.
void	**deleteAttribute**(IRPAttribute attribute) Deletes the specified attribute.
void	**deleteFlowItems**(IRPFlowItem pItem) Deletes the specified item flow.
void	**deleteFlows**(IRPFlow pFlow) Deletes the specified flow.
void	**deleteGeneralization**(IRPClassifier superClass) Deletes the generalization relationship between the classifier and the classifier specified as a parameter.
void	**deleteOperation**(IRPOperation operation) Deletes the specified operation.
void	**deleteRelation**(IRPRelation relation) Deletes the specified relation.
IRPAttribute	**findAttribute**(java.lang.String newVal) Returns the attribute with the name specified.
IRPClassifier	**findBaseClassifier**(java.lang.String newVal) Returns the base classifier with the specified name.
IRPClassifier	**findDerivedClassifier**(java.lang.String newVal) Returns the derived classifier with the specified name.
IRPGeneralization	

Method Summary

	findGeneralization(java.lang.String newVal) Returns the element representing the generalization relationship between this classifier and the classifier whose name was specified as a parameter.
IRPInterfaceItem	**findInterfaceItem**(java.lang.String signature) Gets the operation or event reception that matches the signature provided.
IRPClassifier	**findNestedClassifier**(java.lang.String newVal) Searches for the nested classifier with the name specified.
IRPModelElement	**findNestedClassifierRecursive**(java.lang.String newVal) Searches recursively for the classifier with the name specified.
IRPRelation	**findRelation**(java.lang.String newVal) Returns the association whose name was specified as a parameter.
IRPInterfaceItem	**findTrigger**(java.lang.String name) Returns the trigger with the specified name in the classifier's statechart.
IRPFlowchart	**getActivityDiagram**() This method should no longer be used because Rational Rhapsody now allows you to define more than one statechart and activity diagram for a class.
IRPCollection	**getAttributes**() Returns a collection of all the classifier's attributes.
IRPCollection	**getAttributesIncludingBases**() Returns a collection of all the classifier's attributes, including those it inherits from its base classifiers.
IRPCollection	**getBaseClassifiers**() Returns a collection of the classifiers that server as base classifiers for this classifier.
IRPCollection	**getBehavioralDiagrams**() Returns a collection of all the statecharts and activities defined for the classifier.
IRPCollection	**getDerivedClassifiers**() Returns a collection of all the classifiers derived from this classifier.
IRPCollection	**getFlowItems**() Returns a collection of all the classifier's item flows.
IRPCollection	**getFlows**() Returns a collection of the classifier's flows.
IRPCollection	**getGeneralizations**() Returns a collection of all the classifier's generalization relationships.
IRPCollection	**getInterfaceItems**() Returns a collection of the classifier's elements of type IRPInterfaceItem (such as operations, triggered operations, and event receptions).
IRPCollection	**getInterfaceItemsIncludingBases**() Returns a collection of the classifier's elements of type IRPInterfaceItem (such as operations, triggered operations, and event receptions), including those it inherits from

Method Summary

	its base classifier.
IRPCollection	**getLinks**() Returns a collection of all the classifier's link relationships.
IRPCollection	**getNestedClassifiers**() Returns a collection of all the classifiers nested below the current classifier.
IRPCollection	**getOperations**() Returns a collection of all the classifier's operations.
IRPCollection	**getPorts**() Returns a collection of all the classifier's ports.
IRPCollection	**getRelations**() Returns a collection of all the classifier's associations.
IRPCollection	**getRelationsIncludingBases**() Returns a collection of all the classifier's associations, including those it inherits from its base classifier.
IRPCollection	**getSequenceDiagrams**() Returns a collection of the classifier's sequence diagrams.
IRPCollection	**getSourceArtifacts**() Gets the source artifacts for the classifier.
IRPStatechart	**getStatechart**() This method should no longer be used because Rational Rhapsody now allows you to define more than one statechart and activity diagram for a class.

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern,

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addActivityDiagram

IRPFlowchart **addActivityDiagram**()

> Creates a new activity diagram.
> **Returns:**
> > the activity diagram that was created

---

addAttribute

IRPAttribute **addAttribute**(java.lang.String name)

> Adds a new attribute to the classifier.
> **Parameters:**
> > name - the name to use for the new attribute
> **Returns:**
> > the attribute that was created

---

addFlowItems

IRPFlowItem **addFlowItems**(java.lang.String name)

> Adds a new item flow to the classifier.
> **Parameters:**
> > name - the name to use for the new item flow
> **Returns:**
> > the item flow that was created

---

addFlows

IRPFlow **addFlows**(java.lang.String name)

Adds a new flow to the classifier.
Parameters:
name - the name to use for the new flow
Returns:
the flow that was created

addGeneralization

void **addGeneralization**(IRPClassifier pVal)

Adds a generalization relationship between the classifier and the classifier specified as a parameter.
For example:

```
convertibleClass.addGeneralization(carClass);
```

Parameters:
pVal - the classifier that should serve as the base classifier for this classifier

addOperation

IRPOperation **addOperation**(java.lang.String name)

Adds a new operation.
Parameters:
name - the name to use for the new operation
Returns:
the operation that was created

addRelation

IRPRelation **addRelation**(java.lang.String otherClassName,
                      java.lang.String otherClassPackageName,
                      java.lang.String roleName1,
                      java.lang.String linkType1,
                      java.lang.String multiplicity1,
                      java.lang.String roleName2,
                      java.lang.String linkType2,
                      java.lang.String multiplicity2,
                      java.lang.String linkName)

Adds a new association to the classifier.
Parameters:
otherClassName - the name of the classifier that the current classifier should be associated
with

otherClassPackageName - that name of the package that contains the classifier that the current classifier should be associated with

roleName1 - the role name to use for the association end near the other classifier

linkType1 - used in conjunction with the parameter linkType2 to determine the type of association to create. The strings that can be used for this parameter are Association, Aggregation and Composition (parameter is case-sensitive). To create a simple association, use Association for each of the linkType parameters. To create an aggregation relationship, use Association for one of the linkType parameters and Aggregation for the other parameter. To create a composition relationship, use Association for one of the linkType parameters and use Composition for the other parameter.

multiplicity1 - the multiplicity to use for the association end near the other classifier. You can use strings such as "1" or "14" to specify a specific number, or you can use one of the values listed in the Features dialog for attributes: "0,1", "*", or "1..*".

roleName2 - the role name to use for the association end near the current classifier

linkType2 - used in conjunction with the parameter linkType1 to determine the type of association to create. The strings that can be used for this parameter are Association, Aggregation and Composition (parameter is case-sensitive). To create a simple association, use Association for each of the linkType parameters. To create an aggregation relationship, use Association for one of the linkType parameters and Aggregation for the other parameter. To create a composition relationship, use Association for one of the linkType parameters and use Composition for the other parameter.

multiplicity2 - the multiplicity to use for the association end near the current classifier. You can use strings such as "1" or "14" to specify a specific number, or you can use one of the values listed in the Features dialog for attributes: "0,1", "*", or "1..*".

linkName - if you want to create an association class, use this parameter to specify the name of the class. If you do not want to create an association class, use an empty string as the value of this parameter.

Returns:

the association that was created

---

addRelationTo

```
IRPRelation addRelationTo(IRPClassifier otherClassifier,
                    java.lang.String roleName1,
                    java.lang.String linkType1,
                    java.lang.String multiplicity1,
                    java.lang.String roleName2,
                    java.lang.String linkType2,
                    java.lang.String multiplicity2,
                    java.lang.String linkName)
```

Adds a new association to the classifier.

Parameters:

otherClassifier - the classifier that the current classifier should be associated with

roleName1 - the role name to use for the association end near the other classifier

linkType1 - used in conjunction with the parameter linkType2 to determine the type of association to create. The strings that can be used for this parameter are Association, Aggregation and Composition (parameter is case-sensitive). To create a simple association, use Association for each of the linkType parameters. To create an aggregation relationship, use Association for one of the linkType parameters and Aggregation for the other parameter. To create a composition relationship, use Association for one of the linkType parameters and

use Composition for the other parameter.

`multiplicity1` - the multiplicity to use for the association end near the other classifier. You can use strings such as "1" or "14" to specify a specific number, or you can use one of the values listed in the Features dialog for attributes: "0,1", "*", or "1..*".

`roleName2` - the role name to use for the association end near the current classifier

`linkType2` - used in conjunction with the parameter linkType1 to determine the type of association to create. The strings that can be used for this parameter are Association, Aggregation and Composition (parameter is case-sensitive). To create a simple association, use Association for each of the linkType parameters. To create an aggregation relationship, use Association for one of the linkType parameters and Aggregation for the other parameter. To create a composition relationship, use Association for one of the linkType parameters and use Composition for the other parameter.

`multiplicity2` - the multiplicity to use for the association end near the current classifier. You can use strings such as "1" or "14" to specify a specific number, or you can use one of the values listed in the Features dialog for attributes: "0,1", "*", or "1..*".

`linkName` - if you want to create an association class, use this parameter to specify the name of the class. If you do not want to create an association class, use an empty string as the value of this parameter.

Returns:

the association that was created

addStatechart

[IRPStatechart](#) **addStatechart**()

Creates a new statechart.
Returns:

the statechart that was created

addUnidirectionalRelation

[IRPRelation](#) **addUnidirectionalRelation**(java.lang.String otherClassName,
                                 java.lang.String otherClassPackageName,
                                 java.lang.String roleName,
                                 java.lang.String linkType,
                                 java.lang.String multiplicity,
                                 java.lang.String linkName)

Adds a new directed association to the classifier.
Parameters:

`otherClassName` - the name of the classifier that the current classifier should be associated with

`otherClassPackageName` - that name of the package that contains the classifier that the current classifier should be associated with

`roleName` - the role name to use for the association end

`linkType` - used to determine the type of association to create. The strings that can be used for this parameter are Association, Aggregation and Composition (parameter is case-sensitive).

multiplicity - the multiplicity to use for the association end. You can use strings such as "1" or "14" to specify a specific number, or you can use one of the values listed in the Features dialog for attributes: "0,1", "*", or "1..*".

linkName - if you want to create an association class, use this parameter to specify the name of the class. If you do not want to create an association class, use an empty string as the value of this parameter.

Returns:

the association that was created

addUnidirectionalRelationTo

IRPRelation **addUnidirectionalRelationTo**(IRPClassifier otherClassifier,
                                      java.lang.String roleName,
                                      java.lang.String linkType,
                                      java.lang.String multiplicity,
                                      java.lang.String linkName)

Adds a new directed association to the classifier.

Parameters:

otherClassifier - the classifier that the current classifier should be associated with

roleName - the role name to use for the association end

linkType - used to determine the type of association to create. The strings that can be used for this parameter are Association, Aggregation and Composition (parameter is case-sensitive).

multiplicity - the multiplicity to use for the association end. You can use strings such as "1" or "14" to specify a specific number, or you can use one of the values listed in the Features dialog for attributes: "0,1", "*", or "1..*".

linkName - if you want to create an association class, use this parameter to specify the name of the class. If you do not want to create an association class, use an empty string as the value of this parameter.

Returns:

the association that was created

deleteAttribute

void **deleteAttribute**(IRPAttribute attribute)

Deletes the specified attribute.

Parameters:

attribute - the attribute that should be deleted

deleteFlowItems

void **deleteFlowItems**(IRPFlowItem pItem)

Deletes the specified item flow.

Parameters:

pItem - the item flow that should be deleted

deleteFlows

void **deleteFlows**([IRPFlow](#) pFlow)

> Deletes the specified flow.
> **Parameters:**
> > pFlow - the flow that should be deleted

deleteGeneralization

void **deleteGeneralization**([IRPClassifier](#) superClass)

> Deletes the generalization relationship between the classifier and the classifier specified as a parameter.
> **Parameters:**
> > superClass - the classifier whose generalization relationship with this classifier should be deleted

deleteOperation

void **deleteOperation**([IRPOperation](#) operation)

> Deletes the specified operation.
> **Parameters:**
> > operation - the operation that should be deleted

deleteRelation

void **deleteRelation**([IRPRelation](#) relation)

> Deletes the specified relation.
> **Parameters:**
> > relation - the relation that should be deleted

findAttribute

[IRPAttribute](#) **findAttribute**(java.lang.String newVal)

> Returns the attribute with the name specified.
> **Parameters:**
> > newVal - the name of the attribute that should be returned
> **Returns:**
> > the attribute with the name specified

findBaseClassifier

[IRPClassifier](#) **findBaseClassifier**(java.lang.String newVal)

Returns the base classifier with the specified name.
Parameters:
newVal - the name of the base classifier that should be returned
Returns:
the base classifier with the specified name

findDerivedClassifier

[IRPClassifier](#) **findDerivedClassifier**(java.lang.String newVal)

Returns the derived classifier with the specified name.
Parameters:
newVal - the name of the derived classifier that should be returned
Returns:
the derived classifier with the specified name

findGeneralization

[IRPGeneralization](#) **findGeneralization**(java.lang.String newVal)

Returns the element representing the generalization relationship between this classifier and the classifier whose name was specified as a parameter.
Parameters:
newVal - the name of the classifier whose generalization relationship should be returned
Returns:
the element representing the generalization relationship between this classifier and the classifier whose name was specified as a parameter

findInterfaceItem

[IRPInterfaceItem](#) **findInterfaceItem**(java.lang.String signature)

Gets the operation or event reception that matches the signature provided.
Parameters:
signature - the signature of the operation or event reception. The string you provide should consist of the operation name followed by parentheses containing a comma-delimited list of the types of the parameters, for example, "runEngine(int,int)".
Returns:
the operation or event reception

findNestedClassifier

[IRPClassifier](#) **findNestedClassifier**(java.lang.String newVal)

Searches for the nested classifier with the name specified. This method only searches the first level of elements below the current classifier. To search all of the levels below the current classifier, use the method findNestedClassifierRecursive.

Parameters:

newVal - the name of the classifier to search for

Returns:

the classifier with the name that was specified

findNestedClassifierRecursive

[IRPModelElement](#) **findNestedClassifierRecursive**(java.lang.String newVal)

Searches recursively for the classifier with the name specified. This method searches all of the levels below the current classifier. To search only the first level of elements below the current classifier, use the method findNestedClassifier.

Parameters:

newVal - the name of the classifier to search for

Returns:

the classifier that was specified. Note that the classifier is returned as an object of type IRPModelElement. So you will usually want to use casting, for example: IRPClassifier classifierToSearchFor = (IRPClassifier)stillsCamera.findNestedClassifierRecursive("nested_1_next_level");

findRelation

[IRPRelation](#) **findRelation**(java.lang.String newVal)

Returns the association whose name was specified as a parameter.

Parameters:

newVal - the name of the association that should be returned

Returns:

the association whose name was specified as a parameter

findTrigger

[IRPInterfaceItem](#) **findTrigger**(java.lang.String name)

Returns the trigger with the specified name in the classifier's statechart.

Parameters:

name - the name of the trigger to find

Returns:

the trigger with the specified name in the classifier's statechart

getActivityDiagram

[IRPFlowchart](#) **getActivityDiagram**()

> This method should no longer be used because Rational Rhapsody now allows you to define more than one statechart and activity diagram for a class. Use the method getBehavioralDiagrams instead.

getAttributes

[IRPCollection](#) **getAttributes**()

> Returns a collection of all the classifier's attributes.
> **Returns:**
> > all the classifier's attributes

getAttributesIncludingBases

[IRPCollection](#) **getAttributesIncludingBases**()

> Returns a collection of all the classifier's attributes, including those it inherits from its base classifiers.
> **Returns:**
> > all of the classifier's attributes, including those it inherits from its base classifiers

getBaseClassifiers

[IRPCollection](#) **getBaseClassifiers**()

> Returns a collection of the classifiers that server as base classifiers for this classifier.
> **Returns:**
> > all the classifiers that serve as base classifiers for this classifier

getBehavioralDiagrams

[IRPCollection](#) **getBehavioralDiagrams**()

> Returns a collection of all the statecharts and activities defined for the classifier. The collection that is returned consists of elements of type IRPStatechart.
> **Returns:**
> > all of the statecharts and activities defined for the classifier

getDerivedClassifiers

[IRPCollection](#) **getDerivedClassifiers**()

> Returns a collection of all the classifiers derived from this classifier.

Returns:
>   all the classifiers derived from this classifier

getFlowItems

[IRPCollection](#) **getFlowItems**()

>   Returns a collection of all the classifier's item flows.
>   **Returns:**
>   >   all of the classifier's item flows

getFlows

[IRPCollection](#) **getFlows**()

>   Returns a collection of the classifier's flows.
>   **Returns:**
>   >   all of the classifier's flows

getGeneralizations

[IRPCollection](#) **getGeneralizations**()

>   Returns a collection of all the classifier's generalization relationships.
>   **Returns:**
>   >   all of the classifier's generalization relationships

getInterfaceItems

[IRPCollection](#) **getInterfaceItems**()

>   Returns a collection of the classifier's elements of type IRPInterfaceItem (such as operations,
>   triggered operations, and event receptions).
>   **Returns:**
>   >   all of the classifier's elements of type IRPInterfaceItem

getInterfaceItemsIncludingBases

[IRPCollection](#) **getInterfaceItemsIncludingBases**()

>   Returns a collection of the classifier's elements of type IRPInterfaceItem (such as operations,
>   triggered operations, and event receptions), including those it inherits from its base classifier.
>   **Returns:**
>   >   all of the classifier's elements of type IRPInterfaceItem, including those it inherits from its
>   >   base classifier

getLinks

[IRPCollection](#) **getLinks**()

Returns a collection of all the classifier's link relationships.
Returns:
all of the classifier's link relationships

getNestedClassifiers

[IRPCollection](#) **getNestedClassifiers**()

Returns a collection of all the classifiers nested below the current classifier. Note that this method is not recursive - it only returns the classifiers at the level directly below the current classifier.
Returns:
all of the classifiers nested below the current classifier

getOperations

[IRPCollection](#) **getOperations**()

Returns a collection of all the classifier's operations.
Returns:
all the classifier's operations

getPorts

[IRPCollection](#) **getPorts**()

Returns a collection of all the classifier's ports.
Returns:
all of the classifier's ports

getRelations

[IRPCollection](#) **getRelations**()

Returns a collection of all the classifier's associations.
Returns:
all of the classifier's associations

getRelationsIncludingBases

IRPCollection **getRelationsIncludingBases**()

Returns a collection of all the classifier's associations, including those it inherits from its base classifier.

Returns:

all of the classifier's associations, including those it inherits from its base classifier

getSequenceDiagrams

IRPCollection **getSequenceDiagrams**()

Returns a collection of the classifier's sequence diagrams.

Returns:

all of the classifier's sequence diagrams

getSourceArtifacts

IRPCollection **getSourceArtifacts**()

Gets the source artifacts for the classifier.

Returns:

the source artifacts for the classifier, as a collection of IRPFile objects

getStatechart

IRPStatechart **getStatechart**()

This method should no longer be used because Rational Rhapsody now allows you to define more than one statechart and activity diagram for a class. Use the method getBehavioralDiagrams instead.

com.telelogic.rhapsody.core
Interface IRPClassifierRole

All Superinterfaces:
IRPModelElement

---

public interface **IRPClassifierRole**
extends IRPModelElement

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPClassifier	**getFormalClassifier**() get property formalClassifier
IRPInstance	**getFormalInstance**() get property formalInstance
IRPSequenceDiagram	**getReferencedSequenceDiagram**() get property referencedSequenceDiagram
IRPCollection	**getReferencingClassifierRolesRecursively**() method getReferencingClassifierRolesRecursively
java.lang.String	**getRoleType**() get property roleType
void	**setFormalClassifier**(IRPClassifier formalClassifier) set property formalClassifier
void	**setFormalInstance**(IRPInstance formalInstance) set property formalInstance
void	**setReferencedSequenceDiagram**(IRPSequenceDiagram referencedSequenceDiagram) set property referencedSequenceDiagram

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

---

getStatechart   127

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getFormalClassifier

IRPClassifier **getFormalClassifier**()

get property formalClassifier
Throws:
RhapsodyRuntimeException

getFormalInstance

IRPInstance **getFormalInstance**()

get property formalInstance
Throws:
RhapsodyRuntimeException

getReferencedSequenceDiagram

[IRPSequenceDiagram](#) **getReferencedSequenceDiagram**()

> get property referencedSequenceDiagram
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

getReferencingClassifierRolesRecursively

[IRPCollection](#) **getReferencingClassifierRolesRecursively**()

> method getReferencingClassifierRolesRecursively
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

getRoleType

java.lang.String **getRoleType**()

> get property roleType
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

setFormalClassifier

void **setFormalClassifier**([IRPClassifier](#) formalClassifier)

> set property formalClassifier
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

setFormalInstance

void **setFormalInstance**([IRPInstance](#) formalInstance)

> set property formalInstance
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

setReferencedSequenceDiagram

void **setReferencedSequenceDiagram**([IRPSequenceDiagram](#) referencedSequenceDiagram)

> set property referencedSequenceDiagram
> **Throws:**

com.telelogic.rhapsody.core

[RhapsodyRuntimeException](#)

 com.telelogic.rhapsody.core
Interface IRPCodeGenerator

```
public interface IRPCodeGenerator
```

Method Summary

<u>IRPCollection</u>	**getCodeAnnotations**(<u>IRPModelElement</u> element, int bSpecFile) method to get generated code file names
<u>IRPCollection</u>	**getGeneratedFileNames**(<u>IRPModelElement</u> element) method to get generated code file names
java.lang.String	**getInterfaceName**() get property interfaceName

Method Detail

getCodeAnnotations

```
IRPCollection getCodeAnnotations(IRPModelElement element,
                                 int bSpecFile)
```

method to get generated code file names
Throws:
<u>RhapsodyRuntimeException</u>

---

getGeneratedFileNames

```
IRPCollection getGeneratedFileNames(IRPModelElement element)
```

method to get generated code file names
Throws:
<u>RhapsodyRuntimeException</u>

---

getInterfaceName

`java.lang.String` **`getInterfaceName`**`()`

get property interfaceName
Throws:
[RhapsodyRuntimeException](#)

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

com.telelogic.rhapsody.core
Interface IRPCodeGenSimplifiersRegistry

All Superinterfaces:
> IRPBaseExternalCodeGeneratorTool

---

public interface **IRPCodeGenSimplifiersRegistry**
extends IRPBaseExternalCodeGeneratorTool

Method Summary

java.lang.String	**getInterfaceName**() get property interfaceName
void	**notifySimplificationDone**() method notifySimplificationDone

Methods inherited from interface com.telelogic.rhapsody.core.IRPBaseExternalCodeGeneratorTool
advanceCodeGenProgressBar, shouldAbortCodeGeneration, writeCodeGenMessage

Method Detail

getInterfaceName

java.lang.String **getInterfaceName**()

> get property interfaceName
> **Throws:**
> > RhapsodyRuntimeException

---

notifySimplificationDone

void **notifySimplificationDone**()

> method notifySimplificationDone
> **Throws:**
> > RhapsodyRuntimeException

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

 com.telelogic.rhapsody.core
Interface IRPCollaboration

All Superinterfaces:
> IRPModelElement

All Known Subinterfaces:
> IRPInteractionOperand

---

public interface **IRPCollaboration**
extends IRPModelElement

The IRPCollaboration interface represents the capabilities included in sequence diagrams and communications diagrams. To get the IRPCollaboration object underlying a sequence diagram or a communication diagram, use the methods IRPSequenceDiagram.getLogicalCollaboration() and IRPCollaborationDiagram.getLogicalCollaboration().

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPActionBlock	**addActionBlock**(IRPClassifierRole classifier) Adds a new action block to the specified classifier.
IRPMessage	**addCancelledTimeout**(IRPClassifierRole receiver) Adds a cancelled timeout to the specified instance line.
IRPClassifierRole	**addClassifierRole**(java.lang.String newVal, IRPClassifier cls) Adds an instance line to a sequence diagram.
IRPClassifierRole	**addClassifierRoleByName**(java.lang.String newVal, java.lang.String classFullPath) method addClassifierRoleByName
IRPClassifierRole	**addClassifierRoleForInstance**(IRPInstance inst) method addClassifierRoleForInstance
IRPMessage	**addConditionMark**(IRPClassifierRole classifier) Adds a condition mark to the specified instance line.

Method Summary

IRPMessage	**addCtor**(*IRPInterfaceItem* interItem, java.lang.String actualParamList, *IRPClassifierRole* sender, *IRPClassifierRole* receiver) Adds a Create Arrow to a sequence diagram.
IRPMessage	**addDataFlow**(*IRPSysMLPort* flowPort, java.lang.String value, *IRPClassifierRole* sender, *IRPClassifierRole* receiver) method addDataFlow
IRPMessage	**addDestructionEvent**(*IRPClassifierRole* classifier) Adds a destruction event to the specified lifeline.
IRPMessage	**addDtor**(*IRPInterfaceItem* interItem, java.lang.String actualParamList, *IRPClassifierRole* sender, *IRPClassifierRole* receiver) Adds a Destroy Arrow to a sequence diagram.
IRPMessage	**addDurationConstraint**(java.lang.String durationConstraint, *IRPMessage* startState, *IRPMessage* endState) Adds a Duration Constraint to the specified state invariants.
IRPMessage	**addDurationObservation**(java.lang.String durationObservation, *IRPMessage* startState, *IRPMessage* endState) Adds a Duration Observation to the specified states invariants.
IRPMessage	**addFoundMessage**(*IRPClassifierRole* receiver) Adds a Found Message to the specified lifeline.
IRPInteractionOccurrence	**addInteractionOccurrence**() Adds an interaction occurrence.
IRPInteractionOperator	**addInteractionOperator**() Adds an interaction operator to a sequence diagram.
IRPMessage	**addLostMessage**(*IRPClassifierRole* sender) Adds a Lost Message to the specified lifeline.
IRPMessage	**addMessage**(*IRPInterfaceItem* interItem, java.lang.String actualParamList, *IRPClassifierRole* sender, *IRPClassifierRole* receiver) Adds a message to a sequence diagram.
IRPMessage	**addReplyMessage**(*IRPInterfaceItem* interItem, java.lang.String actualParamList, *IRPClassifierRole* sender, *IRPClassifierRole* receiver) method addReplyMessage
IRPMessage	**addStateInvariant**(java.lang.String invariant, *IRPClassifierRole* classifier) Adds a State Invariant to the specified lifeline.
IRPClassifierRole	**addSystemBorder**() Adds a System Border element to a sequence diagram.
IRPMessage	**addTimeConstraint**(java.lang.String timeConstraint, *IRPMessage* state) Adds a Time Constraint to the specified state invariant.

Method Summary

[IRPMessage](#)	**addTimeInterval**([IRPClassifierRole](#) receiver) Adds a Time Interval to the specified lifeline.
[IRPMessage](#)	**addTimeObservation**(java.lang.String timeObservation, [IRPMessage](#) state) Adds a Time Observation to the specified state invariant.
[IRPMessage](#)	**addTimeout**([IRPInterfaceItem](#) interItem, java.lang.String actualParamList, [IRPClassifierRole](#) sender, [IRPClassifierRole](#) receiver) Adds a timeout to a sequence diagram.
[IRPSequenceDiagram](#)	**generateSequence**(java.lang.String newVal, [IRPPackage](#) owner) Generates a sequence diagram from the content of the IRPCollaboration object.
java.lang.String	**getActivationCondition**() get property activationCondition
java.lang.String	**getActivationMode**() get property activationMode
[IRPMessage](#)	**getActivator**([IRPMessage](#) msg) method getActivator
[IRPCollection](#)	**getAssociations**() get property associations
[IRPCollection](#)	**getClassifier**() Returns a collection of all the instance lines in the sequence diagram.
[IRPCollection](#)	**getConcurrentGroup**([IRPMessage](#) msg) method getConcurrentGroup
[IRPCollection](#)	**getExecutionOccurrences**() Returns a collection of all the Execution Occurrences in the diagram.
[IRPCollection](#)	**getInteractionOccurrences**() Returns a collection of all the interaction occurrences in the sequence diagram.
[IRPCollection](#)	**getInteractionOperators**() Returns a collection of all the interaction operators in the sequence diagram.
[IRPCollection](#)	**getMessagePoints**() Returns all the message points along the specified instance line.
[IRPCollection](#)	**getMessagePoints**([IRPClassifierRole](#) classifier) Returns all the message points along the specified instance line.
[IRPCollection](#)	**getMessages**() Returns a collection of all the messages in the sequence diagram.
java.lang.String	**getMode**() get property mode

Method Summary

IRPMessage	**getPredecessor**(IRPMessage msg) Returns the message that precedes the specified message.
IRPMessage	**getSuccessor**(IRPMessage msg) Returns the message that follows the specified message.

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addActionBlock

IRPActionBlock **addActionBlock**(IRPClassifierRole classifier)

      Adds a new action block to the specified classifier.
      Parameters:
            classifier - the classifier to which the action block should be added
      Returns:
            the action block that was created

addCancelledTimeout

IRPMessage **addCancelledTimeout**(IRPClassifierRole receiver)

> Adds a cancelled timeout to the specified instance line.
> **Parameters:**
> > receiver - the instance line that the cancelled timeout should be added to
> **Returns:**
> > the cancelled timeout that was added

addClassifierRole

IRPClassifierRole **addClassifierRole**(java.lang.String newVal,
                                        IRPClassifier cls)

> Adds an instance line to a sequence diagram.
> **Parameters:**
> > newVal - the name to use for the new instance line
> > cls - the class that the instance line is based on
> **Returns:**
> > the instance line that was added

addClassifierRoleByName

IRPClassifierRole **addClassifierRoleByName**(java.lang.String newVal,
                                              java.lang.String classFullPath)

> method addClassifierRoleByName
> **Throws:**
> > RhapsodyRuntimeException

addClassifierRoleForInstance

IRPClassifierRole **addClassifierRoleForInstance**(IRPInstance inst)

> method addClassifierRoleForInstance
> **Throws:**
> > RhapsodyRuntimeException

addConditionMark

IRPMessage **addConditionMark**(IRPClassifierRole classifier)

> Adds a condition mark to the specified instance line.
> **Parameters:**
> > classifier - the instance line to which the condition mark should be added
> **Returns:**

the condition mark that was created

addCtor

```
IRPMessage addCtor(IRPInterfaceItem interItem,
                   java.lang.String actualParamList,
                   IRPClassifierRole sender,
                   IRPClassifierRole receiver)
```

Adds a Create Arrow to a sequence diagram.

Parameters:

interItem - the constructor for the object to be created

actualParamList - string representing the arguments to pass to the constructor. The string provided should be a comma-separated list of arguments

sender - the instance line at which the Create Arrow begins

receiver - the instance line at which the Create Arrow ends (representing the object to be created)

Returns:

the Create Arrow that was added to the diagram

addDataFlow

```
IRPMessage addDataFlow(IRPSysMLPort flowPort,
                       java.lang.String value,
                       IRPClassifierRole sender,
                       IRPClassifierRole receiver)
```

method addDataFlow

Throws:

RhapsodyRuntimeException

addDestructionEvent

```
IRPMessage addDestructionEvent(IRPClassifierRole classifier)
```

Adds a destruction event to the specified lifeline.

Parameters:

classifier - the lifeline that the destruction event should be added to

Returns:

the destruction event that was created

addDtor

```
IRPMessage addDtor(IRPInterfaceItem interItem,
                   java.lang.String actualParamList,
                   IRPClassifierRole sender,
                   IRPClassifierRole receiver)
```

Adds a Destroy Arrow to a sequence diagram.

Parameters:

> `interItem` - the destructor for the object to be destroyed
>
> `actualParamList` - since destructors do not take arguments, use an empty string "" for this parameter
>
> `sender` - the instance line at which the Destroy Arrow begins
>
> `receiver` - the instance line at which the Destroy Arrow ends (representing the object to be destroyed)

Returns:

> the Destroy Arrow that was added to the diagram

---

addDurationConstraint

[IRPMessage](#) **addDurationConstraint**(java.lang.String durationConstraint,
                                  [IRPMessage](#) startState,
                                  [IRPMessage](#) endState)

Adds a Duration Constraint to the specified state invariants.

Parameters:

> `durationConstraint` - the text to display above the new Duration Constraint
>
> `startState` - the state invariant at which the new Duration Constraint should begin
>
> `endState` - the state invariant at which the new Duration Constraint should end. If you want the Duration Constraint to cover only a single state invariant, specify the same state invariant for both the startState parameter and the endState parameter

Returns:

> the Duration Constraint that was created

---

addDurationObservation

[IRPMessage](#) **addDurationObservation**(java.lang.String durationObservation,
                                   [IRPMessage](#) startState,
                                   [IRPMessage](#) endState)

Adds a Duration Observation to the specified states invariants.

Parameters:

> `durationObservation` - the text to display above the new Duration Observation
>
> `startState` - the state invariant at which the new Duration Observation should begin
>
> `endState` - the state invariant at which the new Duration Observation should end. If you want the Duration Observation to cover only a single state invariant, specify the same state invariant for both the startState parameter and the endState parameter

Returns:

> the Duration Observation that was created

---

addFoundMessage

[IRPMessage](#) **addFoundMessage**([IRPClassifierRole](#) receiver)

Adds a Found Message to the specified lifeline.

addDtor   141

Parameters:
> receiver - the lifeline that the Found Message should be added to

Returns:
> the Found Message that was created

---

addInteractionOccurrence

IRPInteractionOccurrence **addInteractionOccurrence**()

> Adds an interaction occurrence.
>
> **Returns:**
> > the interaction occurrence that was created

---

addInteractionOperator

IRPInteractionOperator **addInteractionOperator**()

> Adds an interaction operator to a sequence diagram.
>
> **Returns:**
> > the interaction operator that was added

---

addLostMessage

IRPMessage **addLostMessage**(IRPClassifierRole sender)

> Adds a Lost Message to the specified lifeline.
>
> **Parameters:**
> > sender - the lifeline that the Lost Message should be added to
>
> **Returns:**
> > the Lost Message that was created

---

addMessage

IRPMessage **addMessage**(IRPInterfaceItem interItem,
                         java.lang.String actualParamList,
                         IRPClassifierRole sender,
                         IRPClassifierRole receiver)

> Adds a message to a sequence diagram.
>
> **Parameters:**
> > interItem - the operation call represented by the message
> >
> > actualParamList - the arguments to pass to the operation. If the operation does not take any arguments, use an empty string "" for this parameter
> >
> > sender - the instance line sending the message
> >
> > receiver - the instance line receiving the message
>
> **Returns:**

the message that was added to the diagram

---

addReplyMessage

IRPMessage **addReplyMessage**(IRPInterfaceItem interItem,
                            java.lang.String actualParamList,
                            IRPClassifierRole sender,
                            IRPClassifierRole receiver)

method addReplyMessage
Throws:
        RhapsodyRuntimeException

---

addStateInvariant

IRPMessage **addStateInvariant**(java.lang.String invariant,
                              IRPClassifierRole classifier)

Adds a State Invariant to the specified lifeline.
Parameters:
        invariant - the text to use for the Invariant field of the new State Invariant
        classifier - the lifeline that the State Invariant should be added to
Returns:
        the State Invariant that was created

---

addSystemBorder

IRPClassifierRole **addSystemBorder**()

Adds a System Border element to a sequence diagram.
Returns:
        the System Border element that was added

---

addTimeConstraint

IRPMessage **addTimeConstraint**(java.lang.String timeConstraint,
                              IRPMessage state)

Adds a Time Constraint to the specified state invariant.
Parameters:
        timeConstraint - the text to display for the new Time Constraint
        state - the state invariant to which the new Time Constraint should be added
Returns:
        the Time Constraint that was created

---

addTimeInterval

IRPMessage **addTimeInterval**(IRPClassifierRole receiver)

> Adds a Time Interval to the specified lifeline.
> **Parameters:**
>> receiver - the lifeline that the Time Interval should be added to
> **Returns:**
>> the Time Interval that was created

---

addTimeObservation

IRPMessage **addTimeObservation**(java.lang.String timeObservation,
                                 IRPMessage state)

> Adds a Time Observation to the specified state invariant.
> **Parameters:**
>> timeObservation - the text to display for the new Time Observation
>> state - the state invariant to which the new Time Observation should be added
> **Returns:**
>> the Time Observation that was created

---

addTimeout

IRPMessage **addTimeout**(IRPInterfaceItem interItem,
                        java.lang.String actualParamList,
                        IRPClassifierRole sender,
                        IRPClassifierRole receiver)

> Adds a timeout to a sequence diagram.
> **Parameters:**
>> interItem - use null for this parameter
>> actualParamList - duration of timeout in milliseconds
>> sender - the instance line that the timeout should be added to
>> receiver - use null for this parameter
> **Returns:**
>> the timeout created

---

generateSequence

IRPSequenceDiagram **generateSequence**(java.lang.String newVal,
                                       IRPPackage owner)

> Generates a sequence diagram from the content of the IRPCollaboration object.
> **Parameters:**
>> newVal - the name to give to the new diagram
>> owner - the package to which the new diagram should belong
> **Returns:**

the sequence diagram created

getActivationCondition

java.lang.String **getActivationCondition**()

> get property activationCondition
> **Throws:**
>> RhapsodyRuntimeException

getActivationMode

java.lang.String **getActivationMode**()

> get property activationMode
> **Throws:**
>> RhapsodyRuntimeException

getActivator

IRPMessage **getActivator**(IRPMessage msg)

> method getActivator
> **Throws:**
>> RhapsodyRuntimeException

getAssociations

IRPCollection **getAssociations**()

> get property associations
> **Throws:**
>> RhapsodyRuntimeException

getClassifier

IRPCollection **getClassifier**()

> Returns a collection of all the instance lines in the sequence diagram.
> **Returns:**
>> all the instance lines in the sequence diagram

getConcurrentGroup

IRPCollection **getConcurrentGroup**(IRPMessage msg)

method getConcurrentGroup
Throws:
RhapsodyRuntimeException

getExecutionOccurrences

IRPCollection **getExecutionOccurrences**()

Returns a collection of all the Execution Occurrences in the diagram.
Returns:
all the Execution Occurrences in the diagram

getInteractionOccurrences

IRPCollection **getInteractionOccurrences**()

Returns a collection of all the interaction occurrences in the sequence diagram.
Returns:
all the interaction occurrences in the sequence diagram

getInteractionOperators

IRPCollection **getInteractionOperators**()

Returns a collection of all the interaction operators in the sequence diagram.
Returns:
all the interaction operators in the sequence diagram

getMessagePoints

IRPCollection **getMessagePoints**(IRPClassifierRole classifier)

Returns all the message points along the specified instance line.
Returns:
a collection of IRPMessagePoint objects, representing all the message points along the
specified instance line (in the correct order)

getMessagePoints

IRPCollection **getMessagePoints**()

Returns all the message points along the specified instance line.

Returns:

a collection of IRPMessagePoint objects, representing all the message points along the specified instance line (in the correct order)

getMessages

IRPCollection **getMessages**()

Returns a collection of all the messages in the sequence diagram.

Returns:

all the messages in the sequence diagram

getMode

java.lang.String **getMode**()

get property mode

Throws:

RhapsodyRuntimeException

getPredecessor

IRPMessage **getPredecessor**(IRPMessage msg)

Returns the message that precedes the specified message.

Parameters:

msg - the message whose predecessor has to be found

Returns:

the message that precedes the specified message

getSuccessor

IRPMessage **getSuccessor**(IRPMessage msg)

Returns the message that follows the specified message.

Parameters:

msg - the message whose successor has to be found

Returns:

the message that follows the specified message

com.telelogic.rhapsody.core
Interface IRPCollaborationDiagram

All Superinterfaces:
> IRPDiagram, IRPModelElement, IRPUnit

---

public interface **IRPCollaborationDiagram**
extends IRPDiagram

The IRPCollaborationDiagram interface represents collaboration diagrams in a Rational Rhapsody model.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPCollaboration	**getLogicalCollaboration**() Returns the IRPCollaboration object underlying the collaboration diagram.

Methods inherited from interface com.telelogic.rhapsody.core.IRPDiagram
addFreeShapeByType, addImage, addNewEdgeByType, addNewEdgeForElement, addNewNodeByType, addNewNodeForElement, addTextBox, closeDiagram, completeRelations, getCorrespondingGraphicElements, getElementsInDiagram, getGraphicalElements, getLastVisualizationModifiedTime, getPicture, getPictureAs, getPictureAsDividedMetafiles, getPictureEx, getPicturesWithImageMap, isOpen, isShowDiagramFrame, openDiagram, openDiagramView, populateDiagram, removeGraphElements, setShowDiagramFrame

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit
copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getLogicalCollaboration

IRPCollaboration **getLogicalCollaboration**()

Returns the IRPCollaboration object underlying the collaboration diagram.
Returns:
the IRPCollaboration object underlying the collaboration diagram

---

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | <u>METHOD</u>   DETAIL: FIELD | CONSTR | <u>METHOD</u>

com.telelogic.rhapsody.core
Interface IRPCollection

```
public interface IRPCollection
```

The IRPCollection interface contains methods used to store and manipulate collections of various types of elements that you may have in your Rational Rhapsody model. Collections of this type are used by methods that return multiple model elements and by certain methods that take a collection of model elements as an argument.

Method Summary

void	**addGraphicalItem**(<u>IRPGraphElement</u> newVal) Adds a graphical element to a collection.
void	**addItem**(<u>IRPModelElement</u> newVal) Adds a model element to a collection.
void	**empty**() Used to empty out a collection.
int	**getCount**() Returns the number of items in a collection.
java.lang.Object	**getItem**(int index) Retrieves an item from a collection, using the index specified.
void	**remove**(int index) Removes an element from a collection.
void	**setInteger**(int index, int val) Used to place an integer in a specific place in a collection.
void	**setModelElement**(int index, <u>IRPModelElement</u> val) Places an item in a specific place in a collection.
void	**setSize**(int size) Sets the size of a collection.
void	**setString**(int index, java.lang.String val) Used to place a String in a specific place in a collection.
java.util.List	**toList**() Returns a java.util.List populated with the elements in the collection.

Method Detail

getCount

int **getCount**()

>> Returns the number of items in a collection.
>> **Returns:**
>>> the number of items in the collection

---

getItem

java.lang.Object **getItem**(int index)

>> Retrieves an item from a collection, using the index specified. Note that when using the getItem
>> method, the index parameter is based on an index value of 1 for the first element (not 0).
>> **Parameters:**
>>> index - the index of the item to be retrieved (index of first element is 1, not 0)
>> **Returns:**
>>> the item with the index specified
>> **Throws:**
>>> RhapsodyRuntimeException

---

addItem

void **addItem**(IRPModelElement newVal)

>> Adds a model element to a collection. This method adds items, one at a time, to the end of a
>> collection. When adding multiple items, it may be more efficient to call setSize to set the new size of
>> the collection and then call setModelElement to place elements in specific locations in the collection.
>> **Parameters:**
>>> newVal - the model element to add to the collection
>> **Throws:**
>>> RhapsodyRuntimeException

---

addGraphicalItem

void **addGraphicalItem**(IRPGraphElement newVal)

>> Adds a graphical element to a collection.
>> **Parameters:**
>>> newVal - the graphical element to add to the collection
>> **Throws:**
>>> RhapsodyRuntimeException

---

toList

```
java.util.List toList()
```

>Returns a java.util.List populated with the elements in the collection.
>**Returns:**
>>java.util.List populated with the elements in the collection
>
>**Throws:**
>>RhapsodyRuntimeException

---

setSize

```
void setSize(int size)
```

>Sets the size of a collection.
>**Parameters:**
>>size - the new size that should be used for the collection
>
>**Throws:**
>>RhapsodyRuntimeException

---

remove

```
void remove(int index)
```

>Removes an element from a collection.
>**Parameters:**
>>index - the index of the element that should be removed from the collection
>
>**Throws:**
>>RhapsodyRuntimeException

---

setString

```
void setString(int index,
               java.lang.String val)
```

>Used to place a String in a specific place in a collection. Note that when using setString, the index parameter is based on an index value of 1 for the first element (not 0). The following code illustrates the use of this method with the populateDiagram method, which takes a number of arguments, one of which is a collection of Strings.

```
//The populateDiagram method takes 3 parameters, the first two being collections: a col
IRPDiagram classDiagramToCreate = vehiclePackage.addObjectModelDiagram("Classes in Vehi
IRPCollection classesToAddToDiagram = vehiclePackage.getClasses();
IRPCollection typesOfRelationsToShow = app.createNewCollection();
typesOfRelationsToShow.setSize(2);
typesOfRelationsToShow.setString(1, "Inheritance");
typesOfRelationsToShow.setString(2, "Dependency");
classDiagramToCreate.populateDiagram(classesToAddToDiagram, typesOfRelationsToShow, "fr
```

Parameters:

> `index` - the index representing the place in the collection where the String should be placed (index of first element is 1, not 0)
>
> `val` - the String to place in the collection

Throws:

> [RhapsodyRuntimeException](#)

---

setModelElement

```
void setModelElement(int index,
                     IRPModelElement val)
```

Places an item in a specific place in a collection. Note that when using setModelElement, the index parameter is based on an index value of 1 for the first element (not 0).

Parameters:

> `index` - the index representing the place in the collection where the item should be placed (index of first element is 1, not 0)
>
> `val` - the item to place in the collection

Throws:

> [RhapsodyRuntimeException](#)

---

empty

```
void empty()
```

Used to empty out a collection.

Throws:

> [RhapsodyRuntimeException](#)

---

setInteger

```
void setInteger(int index,
                int val)
```

Used to place an integer in a specific place in a collection. Note that when using setInteger, the index parameter is based on an index value of 1 for the first element (not 0).

Parameters:

> `index` - the index representing the place in the collection where the integer should be placed (index of first element is 1, not 0)
>
> `val` - the integer to place in the collection

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPComment

All Superinterfaces:
> IRPAnnotation, IRPModelElement, IRPUnit

---

public interface **IRPComment**
extends IRPAnnotation

The IRPComment interface represents comments in a Rational Rhapsody model.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

Methods inherited from interface com.telelogic.rhapsody.core.IRPAnnotation
addAnchor, getAnchoredByMe, getBody, getSpecification, getSpecificationRTF, isSpecificationRTF, removeAnchor, setBody, setSpecification, setSpecificationRTF

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit
copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies,

setInteger   154

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**

getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

PREV CLASS　**NEXT CLASS**　　　　　　　　　　　**FRAMES**　**NO FRAMES**　**All Classes**

SUMMARY: NESTED | FIELD | CONSTR | **METHOD**　　　　　　DETAIL: FIELD | CONSTR | **METHOD**

com.telelogic.rhapsody.core
Interface IRPComponent

All Superinterfaces:
　　　IRPModelElement, IRPUnit

```
public interface IRPComponent
```
extends IRPUnit

The IRPComponent interface represents a code generation component defined in a Rhapsody model.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPConfiguration	**addConfiguration**(java.lang.String name) 　　　Adds a new configuration to the component.
IRPFile	**addFile**(java.lang.String name) 　　　Adds a new File to the component.
IRPFile	**addFolder**(java.lang.String name) 　　　Adds a new Folder to the component.
IRPComponent	**addNestedComponent**(java.lang.String name) 　　　Adds a new nested component to the component.
void	**addScopeElement**(IRPModelElement pVal) 　　　Adds the specified model element to the scope of the component.
void	**addScopeElementWithoutAggregates**(IRPModelElement pVal) 　　　method addScopeElementWithoutAggregates
void	**addToScope**(IRPFile file, IRPCollection classes, IRPCollection packages) 　　　method addToScope
void	**allElementsInScope**() 　　　Adds all the elements in the model to the scope of the component.

Method Summary

void	**deleteConfiguration**(IRPConfiguration configuration) Deletes the specified configuration.
void	**deleteFile**(IRPFile file) Deletes the specified File.
IRPConfiguration	**findConfiguration**(java.lang.String name) Returns the configuration with the specified name.
java.lang.String	**getAdditionalSources**() Returns the additional sources defined for the component.
java.lang.String	**getBuildType**() Returns the build type of the component - Library, Executable, or Analysis.
IRPConfiguration	**getConfigByDependency**(IRPDependency o) method getConfigByDependency
IRPCollection	**getConfigurations**() Returns a collection of all the configurations in the component.
IRPFile	**getFile**(IRPClassifier c, int spec) method getFile
java.lang.String	**getFileName**(IRPClassifier c, int spec, int withExt) method getFileName
IRPCollection	**getFiles**() Returns a collection of all the Files in the component.
java.lang.String	**getIncludePath**() Returns the include path defined for the component.
java.lang.String	**getLibraries**() get property libraries
java.lang.String	**getModelElementFileName**(IRPModelElement c, int spec, int withExt) method getModelElementFileName
IRPCollection	**getNestedComponents**() Returns a collection of all the nested components in the component.
IRPFile	**getPackageFile**(IRPPackage c, int spec) method getPackageFile
IRPCollection	**getPanelDiagrams**() Returns a collection of all the panel diagrams in the component.
java.lang.String	**getPath**(int fullPath) get property path
IRPCollection	**getPossibleVariants**(IRPModelElement variationPoint) method getPossibleVariants
int	**getScopeBySelectedElements**() Checks whether the scope of the component has been set to include all elements or only specific elements.

Method Summary

IRPCollection	**getScopeElements**() Returns a collection of all the model elements in the scope of the component.
IRPCollection	**getScopeElementsByCategory**(java.lang.String category) method getScopeElementsByCategory
java.lang.String	**getStandardHeaders**() Returns the standard headers defined for the component.
IRPModelElement	**getVariant**(_IRPModelElement_ variationPoint) method getVariant
IRPCollection	**getVariationPoints**() Returns a collection of the variation points that are included in the scope of the component.
int	**isDirectoryPerModelComponent**(_IRPModelElement_ o) method isDirectoryPerModelComponent
void	**removeScopeElement**(_IRPModelElement_ pVal) Removes the specified model element from the scope of the component.
void	**setAdditionalSources**(java.lang.String additionalSources) Specifies the additional sources to use for the component.
void	**setBuildType**(java.lang.String buildType) Specifies the build type for the component.
void	**setIncludePath**(java.lang.String includePath) Specifies the include path to use for the component.
void	**setLibraries**(java.lang.String libraries) set property libraries
void	**setPath**(java.lang.String path) method setPath
void	**setScopeBySelectedElements**(int scopeBySelectedElements) set toggle the scope between selected and all-elements
void	**setStandardHeaders**(java.lang.String standardHeaders) Specifies the standard headers for the component.
void	**setVariant**(_IRPModelElement_ variationPoint, _IRPModelElement_ pVariant) method setVariant

Methods inherited from interface com.telelogic.rhapsody.core.**IRPUnit**

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addConfiguration

IRPConfiguration **addConfiguration**(java.lang.String name)

Adds a new configuration to the component.
Parameters:
name - the name to use for the new configuration
Returns:
the configuration that was created

addFile

IRPFile **addFile**(java.lang.String name)

Adds a new File to the component.
Parameters:
name - the name to use for the new File
Returns:

the File that was created

---

addFolder

<u>IRPFile</u> **addFolder**(java.lang.String name)

> Adds a new Folder to the component.
> **Parameters:**
>> name - the name to use for the new Folder
> **Returns:**
>> the Folder that was created

---

addNestedComponent

<u>IRPComponent</u> **addNestedComponent**(java.lang.String name)

> Adds a new nested component to the component.
> **Parameters:**
>> name - the name to use for the new component
> **Returns:**
>> the component that was created

---

addScopeElement

void **addScopeElement**(<u>IRPModelElement</u> pVal)

> Adds the specified model element to the scope of the component.
> **Parameters:**
>> pVal - the model element that should be added to the scope of the component

---

addScopeElementWithoutAggregates

void **addScopeElementWithoutAggregates**(<u>IRPModelElement</u> pVal)

> method addScopeElementWithoutAggregates
> **Throws:**
>> <u>RhapsodyRuntimeException</u>

---

addToScope

void **addToScope**(<u>IRPFile</u> file,
                <u>IRPCollection</u> classes,
                <u>IRPCollection</u> packages)

> method addToScope

Throws:

<u>RhapsodyRuntimeException</u>

---

allElementsInScope

void **allElementsInScope**()

Adds all the elements in the model to the scope of the component.

---

deleteConfiguration

void **deleteConfiguration**(<u>IRPConfiguration</u> configuration)

Deletes the specified configuration.
Parameters:
configuration - the configuration that should be deleted

---

deleteFile

void **deleteFile**(<u>IRPFile</u> file)

Deletes the specified File.
Parameters:
file - the File that should be deleted

---

findConfiguration

<u>IRPConfiguration</u> **findConfiguration**(java.lang.String name)

Returns the configuration with the specified name.
Parameters:
name - the name of the configuration to return
Returns:
the configuration with the specified name

---

getAdditionalSources

java.lang.String **getAdditionalSources**()

Returns the additional sources defined for the component.
Returns:
the additional sources defined for the component

---

getBuildType

java.lang.String **getBuildType**()

> Returns the build type of the component - Library, Executable, or Analysis.
> **Returns:**
> > the build type of the component - Library, Executable, or Analysis

getConfigByDependency

IRPConfiguration **getConfigByDependency**(IRPDependency o)

> method getConfigByDependency
> **Throws:**
> > RhapsodyRuntimeException

getConfigurations

IRPCollection **getConfigurations**()

> Returns a collection of all the configurations in the component.
> **Returns:**
> > all the configurations in the component

getFile

IRPFile **getFile**(IRPClassifier c,
> > int spec)

> method getFile
> **Throws:**
> > RhapsodyRuntimeException

getFileName

java.lang.String **getFileName**(IRPClassifier c,
> > int spec,
> > int withExt)

> method getFileName
> **Throws:**
> > RhapsodyRuntimeException

getFiles

[IRPCollection](#) **getFiles**()

> Returns a collection of all the Files in the component.
> **Returns:**
> > all the Files in the component

---

getIncludePath

java.lang.String **getIncludePath**()

> Returns the include path defined for the component.
> **Returns:**
> > the include path defined for the component

---

getLibraries

java.lang.String **getLibraries**()

> get property libraries
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

getModelElementFileName

java.lang.String **getModelElementFileName**([IRPModelElement](#) c,
                                             int spec,
                                             int withExt)

> method getModelElementFileName
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

getNestedComponents

[IRPCollection](#) **getNestedComponents**()

> Returns a collection of all the nested components in the component.
> **Returns:**
> > all the nested components in the component

---

getPackageFile

```
IRPFile getPackageFile(IRPPackage c,
                       int spec)
```

method getPackageFile
Throws:
    RhapsodyRuntimeException

---

getPanelDiagrams

```
IRPCollection getPanelDiagrams()
```

Returns a collection of all the panel diagrams in the component.
Returns:
    all the panel diagrams in the component

---

getPath

```
java.lang.String getPath(int fullPath)
```

get property path
Throws:
    RhapsodyRuntimeException

---

getPossibleVariants

```
IRPCollection getPossibleVariants(IRPModelElement variationPoint)
```

method getPossibleVariants
Throws:
    RhapsodyRuntimeException

---

getScopeBySelectedElements

```
int getScopeBySelectedElements()
```

Checks whether the scope of the component has been set to include all elements or only specific elements. This corresponds to the All Elements and Selected Elements radio buttons on the Scope tab of the Features dialog for components.
Returns:
    1 if the scope has been set to include only specific elements, 0 if the scope has been set to include all elements

---

getScopeElements

IRPCollection **getScopeElements**()

> Returns a collection of all the model elements in the scope of the component.
> **Returns:**
>> all the model elements in the scope of the component

---

getScopeElementsByCategory

IRPCollection **getScopeElementsByCategory**(java.lang.String category)

> method getScopeElementsByCategory
> **Throws:**
>> RhapsodyRuntimeException

---

getStandardHeaders

java.lang.String **getStandardHeaders**()

> Returns the standard headers defined for the component.
> **Returns:**
>> the standard headers defined for the component

---

getVariant

IRPModelElement **getVariant**(IRPModelElement variationPoint)

> method getVariant
> **Throws:**
>> RhapsodyRuntimeException

---

getVariationPoints

IRPCollection **getVariationPoints**()

> Returns a collection of the variation points that are included in the scope of the component. The collection consists of objects of type IRPClass.
> **Returns:**
>> a collection of the variation points that are included in the scope of the component

---

isDirectoryPerModelComponent

int **isDirectoryPerModelComponent**(IRPModelElement o)

method isDirectoryPerModelComponent
Throws:
> RhapsodyRuntimeException

removeScopeElement

void **removeScopeElement**(IRPModelElement pVal)

> Removes the specified model element from the scope of the component.
> **Parameters:**
>> `pVal` - the model element that should be removed from the scope of the component

setAdditionalSources

void **setAdditionalSources**(java.lang.String additionalSources)

> Specifies the additional sources to use for the component.
> **Parameters:**
>> `additionalSources` - the additional sources to use for the component

setBuildType

void **setBuildType**(java.lang.String buildType)

> Specifies the build type for the component.
> **Parameters:**
>> `buildType` - the build type that should be used for the component. The valid strings for this parameter are: Executable, Library, and Analysis.

setIncludePath

void **setIncludePath**(java.lang.String includePath)

> Specifies the include path to use for the component.
> **Parameters:**
>> `includePath` - the include path to use for the component

setLibraries

void **setLibraries**(java.lang.String libraries)

> set property libraries
> **Throws:**
>> RhapsodyRuntimeException

setPath

void **setPath**(java.lang.String path)

> method setPath
> **Throws:**
> > RhapsodyRuntimeException

---

setScopeBySelectedElements

void **setScopeBySelectedElements**(int scopeBySelectedElements)

> set toggle the scope between selected and all-elements
> **Throws:**
> > RhapsodyRuntimeException

---

setStandardHeaders

void **setStandardHeaders**(java.lang.String standardHeaders)

> Specifies the standard headers for the component.
> **Parameters:**
> > standardHeaders - a string consisting of a comma-separated list of the files that should be used as standard headers for the component

---

setVariant

void **setVariant**(IRPModelElement variationPoint,
                IRPModelElement pVariant)

> method setVariant
> **Throws:**
> > RhapsodyRuntimeException

Package Class [Use](Use) Tree Serialized Deprecated Index Help

PREV CLASS NEXT CLASS   **FRAMES NO FRAMES   All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPComponentDiagram

All Superinterfaces:
> IRPDiagram, IRPModelElement, IRPUnit

```
public interface IRPComponentDiagram
```
extends IRPDiagram

The IRPComponentDiagram interface represents component diagrams in Rational Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement

IRPModelElement.OSLCLink

Method Summary

Methods inherited from interface com.telelogic.rhapsody.core.IRPDiagram

addFreeShapeByType, addImage, addNewEdgeByType, addNewEdgeForElement, addNewNodeByType, addNewNodeForElement, addTextBox, closeDiagram, completeRelations, getCorrespondingGraphicElements, getElementsInDiagram, getGraphicalElements, getLastVisualizationModifiedTime, getPicture, getPictureAs, getPictureAsDividedMetafiles, getPictureEx, getPicturesWithImageMap, isOpen, isShowDiagramFrame, openDiagram, openDiagramView, populateDiagram, removeGraphElements, setShowDiagramFrame

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype,

---

setVariant   168

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

---

Package   **Class**   **Use**   **Tree**   **Serialized**   **Deprecated**   **Index**   **Help**

PREV CLASS   **NEXT CLASS**   **FRAMES**   **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPComponentInstance

All Superinterfaces:
> IRPModelElement

---

public interface **IRPComponentInstance**
extends IRPModelElement

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPComponent	**getComponentType**() get property componentType
IRPNode	**getNode**() get property node
void	**setComponentType**(IRPComponent componentType) set property componentType

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit,

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getComponentType

IRPComponent **getComponentType**()

> get property componentType
> **Throws:**
> > RhapsodyRuntimeException

---

getNode

IRPNode **getNode**()

> get property node
> **Throws:**
> > RhapsodyRuntimeException

---

setComponentType

void **setComponentType**(IRPComponent componentType)

> set property componentType
> **Throws:**
> > RhapsodyRuntimeException

---

com.telelogic.rhapsody.core
Interface IRPConditionMark

All Superinterfaces:
>   IRPMessage, IRPModelElement

---

public interface **IRPConditionMark**
extends IRPMessage

The IRPConditionMark interface represents condition marks in sequence diagrams.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

Methods inherited from interface com.telelogic.rhapsody.core.IRPMessage
addSourceExecutionOccurrence, addTargetExecutionOccurrence, getActualParameterList, getCommunicationConnection, getCondition, getDurationConstraint, getDurationObservation, getFlowPort, getFormalInterfaceItem, getFormalType, getInvariant, getMessageType, getPort, getReturnValue, getSequenceNumber, getSignature, getSource, getSourceExecutionOccurrence, getTarget, getTargetExecutionOccurrence, getTimeConstraint, getTimeObservation, getTimerValue, reroute, setActualParameterList, setDurationConstraint, setDurationObservation, setFlowPort, setFormalInterfaceItem, setFormalType, setInvariant, setPort, setReturnValue, setTimeConstraint, setTimeObservation, setTimerValue

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal,

setComponentType   172

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

com.telelogic.rhapsody.core
Interface IRPConfiguration

All Superinterfaces:
> IRPModelElement

public interface **IRPConfiguration**
extends IRPModelElement

The IRPConfiguration interface represents a code generation configuration within a specific component.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

void	**addInitialInstance**(IRPModelElement newVal) method addInitialInstance
void	**addPackageToInstrumentationScope**(IRPPackage pVal) method addPackageToInstrumentationScope
void	**addToInstrumentationScope**(IRPClassifier pVal) method addToInstrumentationScope
void	**deleteInitialInstance**(IRPModelElement newVal) method deleteInitialInstance
java.lang.String	**getAdditionalSources**() get property additionalSources
int	**getAllElementsInInstrumentationScope**() Checks whether the instrumentation mode selected for the configuration applies to all elements or just selected elements.
java.lang.String	**getBuildSet**() get property buildSet
java.lang.String	**getCompilerSwitches**() get property compilerSwitches

Method Summary

java.lang.String	**getDirectory**(int fullpath, java.lang.String nawName) method getDirectory
java.lang.String	**getExecutableName**() method getExecutableName
int	**getGenerateCodeForActors**() get property generateCodeForActors
java.lang.String	**getIncludePath**() get property includePath
[IRPCollection](#)	**getInitialInstances**() get method initialInstances
java.lang.String	**getInitializationCode**() get property initializationCode
[IRPCollection](#)	**getInstrumentationScope**() get property instrumentationScope
java.lang.String	**getInstrumentationType**() get property instrumentationType
[IRPComponent](#)	**getItsComponent**() method getItsComponent
java.lang.String	**getLibraries**() get property libraries
java.lang.String	**getLinkSwitches**() get property linkSwitches
java.lang.String	**getMainName**() method getMainName
java.lang.String	**getMakefileName**(int fullpath) method getMakefileName
java.lang.String	**getPath**(int fullPath) get property path
java.lang.String	**getScopeType**() get property scopeType
java.lang.String	**getStandardHeaders**() get property standardHeaders
java.lang.String	**getStatechartImplementation**() get property statechartImplementation
java.lang.String	**getTargetName**(int fullpath) method getTargetName
java.lang.String	**getTimeModel**() get property timeModel

Method Summary

int	**needsCodeGeneration**() method needsCodeGeneration checks is code generation is needed
void	**removeFromInstrumentationScope**([IRPClassifier](#) pVal) method removeFromInstrumentationScope
void	**removePackageFromInstrumentationScope**([IRPPackage](#) pVal) method removePackageFromInstrumentationScope
void	**setAdditionalSources**(java.lang.String additionalSources) set property additionalSources
void	**setAllElementsInInstrumentationScope**(int allElementsInInstrumentationScope) set property allElementsInInstrumentationScope
void	**setBuildSet**(java.lang.String buildSet) set property buildSet
void	**setCompilerSwitches**(java.lang.String compilerSwitches) set property compilerSwitches
void	**setDirectory**(int fullpath, java.lang.String newName) method setDirectory
void	**setGenerateCodeForActors**(int generateCodeForActors) set property generateCodeForActors
void	**setIncludePath**(java.lang.String includePath) set property includePath
void	**setInitializationCode**(java.lang.String initializationCode) set property initializationCode
void	**setInstrumentationType**(java.lang.String instrumentationType) set property instrumentationType
void	**setItsComponent**([IRPComponent](#) newVal) method setItsComponent
void	**setLibraries**(java.lang.String libraries) set property libraries
void	**setLinkSwitches**(java.lang.String linkSwitches) set property linkSwitches
void	**setScopeType**(java.lang.String scopeType) set property scopeType
void	**setStandardHeaders**(java.lang.String standardHeaders) set property standardHeaders
void	**setStatechartImplementation**(java.lang.String statechartImplementation) set property statechartImplementation
void	**setTimeModel**(java.lang.String timeModel) set property timeModel

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addInitialInstance

void **addInitialInstance**(IRPModelElement newVal)

> method addInitialInstance
> **Throws:**
> > RhapsodyRuntimeException

---

addPackageToInstrumentationScope

void **addPackageToInstrumentationScope**(IRPPackage pVal)

> method addPackageToInstrumentationScope
> **Throws:**
> > RhapsodyRuntimeException

---

addToInstrumentationScope

void **addToInstrumentationScope**([IRPClassifier](#) pVal)

> method addToInstrumentationScope
> **Throws:**
> > [RhapsodyRuntimeException](#)

deleteInitialInstance

void **deleteInitialInstance**([IRPModelElement](#) newVal)

> method deleteInitialInstance
> **Throws:**
> > [RhapsodyRuntimeException](#)

getAdditionalSources

java.lang.String **getAdditionalSources**()

> get property additionalSources
> **Throws:**
> > [RhapsodyRuntimeException](#)

getAllElementsInInstrumentationScope

int **getAllElementsInInstrumentationScope**()

> Checks whether the instrumentation mode selected for the configuration applies to all elements or just selected elements. This corresponds to the Instrumentation Scope options in the Advanced Instrumentation Settings dialog for configurations.
> **Returns:**
> > 1 if the instrumentation mode applies to all elements, 0 otherwise

getBuildSet

java.lang.String **getBuildSet**()

> get property buildSet
> **Throws:**
> > [RhapsodyRuntimeException](#)

getCompilerSwitches

```
java.lang.String getCompilerSwitches()
```

> get property compilerSwitches
> **Throws:**
> > RhapsodyRuntimeException

---

getDirectory

```
java.lang.String getDirectory(int fullpath,
                              java.lang.String nawName)
```

> method getDirectory
> **Throws:**
> > RhapsodyRuntimeException

---

getExecutableName

```
java.lang.String getExecutableName()
```

> method getExecutableName
> **Throws:**
> > RhapsodyRuntimeException

---

getGenerateCodeForActors

```
int getGenerateCodeForActors()
```

> get property generateCodeForActors
> **Throws:**
> > RhapsodyRuntimeException

---

getIncludePath

```
java.lang.String getIncludePath()
```

> get property includePath
> **Throws:**
> > RhapsodyRuntimeException

---

getInitialInstances

```
IRPCollection getInitialInstances()
```

> get method initialInstances

getCompilerSwitches

Throws:
>   RhapsodyRuntimeException

getInitializationCode

java.lang.String **getInitializationCode**()

>   get property initializationCode
>   **Throws:**
>   RhapsodyRuntimeException

getInstrumentationScope

IRPCollection **getInstrumentationScope**()

>   get property instrumentationScope
>   **Throws:**
>   RhapsodyRuntimeException

getInstrumentationType

java.lang.String **getInstrumentationType**()

>   get property instrumentationType
>   **Throws:**
>   RhapsodyRuntimeException

getItsComponent

IRPComponent **getItsComponent**()

>   method getItsComponent
>   **Throws:**
>   RhapsodyRuntimeException

getLibraries

java.lang.String **getLibraries**()

>   get property libraries
>   **Throws:**
>   RhapsodyRuntimeException

getLinkSwitches

`java.lang.String` **`getLinkSwitches`**`()`

> get property linkSwitches
> **Throws:**
> > RhapsodyRuntimeException

getMainName

`java.lang.String` **`getMainName`**`()`

> method getMainName
> **Throws:**
> > RhapsodyRuntimeException

getMakefileName

`java.lang.String` **`getMakefileName`**`(int fullpath)`

> method getMakefileName
> **Throws:**
> > RhapsodyRuntimeException

getPath

`java.lang.String` **`getPath`**`(int fullPath)`

> get property path
> **Throws:**
> > RhapsodyRuntimeException

getScopeType

`java.lang.String` **`getScopeType`**`()`

> get property scopeType
> **Throws:**
> > RhapsodyRuntimeException

getStandardHeaders

`java.lang.String` **`getStandardHeaders`**`()`

> get property standardHeaders
> **Throws:**

RhapsodyRuntimeException

getStatechartImplementation

java.lang.String **getStatechartImplementation**()

> get property statechartImplementation
> **Throws:**
> > RhapsodyRuntimeException

getTargetName

java.lang.String **getTargetName**(int fullpath)

> method getTargetName
> **Throws:**
> > RhapsodyRuntimeException

getTimeModel

java.lang.String **getTimeModel**()

> get property timeModel
> **Throws:**
> > RhapsodyRuntimeException

needsCodeGeneration

int **needsCodeGeneration**()

> method needsCodeGeneration checks is code generation is needed
> **Throws:**
> > RhapsodyRuntimeException

removeFromInstrumentationScope

void **removeFromInstrumentationScope**(IRPClassifier pVal)

> method removeFromInstrumentationScope
> **Throws:**
> > RhapsodyRuntimeException

removePackageFromInstrumentationScope

void **removePackageFromInstrumentationScope**(IRPPackage pVal)

> method removePackageFromInstrumentationScope
> **Throws:**
> > RhapsodyRuntimeException

setAdditionalSources

void **setAdditionalSources**(java.lang.String additionalSources)

> set property additionalSources
> **Throws:**
> > RhapsodyRuntimeException

setAllElementsInInstrumentationScope

void **setAllElementsInInstrumentationScope**(int allElementsInInstrumentationScope)

> set property allElementsInInstrumentationScope
> **Throws:**
> > RhapsodyRuntimeException

setBuildSet

void **setBuildSet**(java.lang.String buildSet)

> set property buildSet
> **Throws:**
> > RhapsodyRuntimeException

setCompilerSwitches

void **setCompilerSwitches**(java.lang.String compilerSwitches)

> set property compilerSwitches
> **Throws:**
> > RhapsodyRuntimeException

setDirectory

void **setDirectory**(int fullpath,
                java.lang.String newName)

> method setDirectory

setGenerateCodeForActors

void **setGenerateCodeForActors**(int generateCodeForActors)

> set property generateCodeForActors
> **Throws:**
>   RhapsodyRuntimeException

setIncludePath

void **setIncludePath**(java.lang.String includePath)

> set property includePath
> **Throws:**
>   RhapsodyRuntimeException

setInitializationCode

void **setInitializationCode**(java.lang.String initializationCode)

> set property initializationCode
> **Throws:**
>   RhapsodyRuntimeException

setInstrumentationType

void **setInstrumentationType**(java.lang.String instrumentationType)

> set property instrumentationType
> **Throws:**
>   RhapsodyRuntimeException

setItsComponent

void **setItsComponent**(IRPComponent newVal)

> method setItsComponent
> **Throws:**
>   RhapsodyRuntimeException

setLibraries

void **setLibraries**(java.lang.String libraries)

> set property libraries
> **Throws:**
> > RhapsodyRuntimeException

---

setLinkSwitches

void **setLinkSwitches**(java.lang.String linkSwitches)

> set property linkSwitches
> **Throws:**
> > RhapsodyRuntimeException

---

setScopeType

void **setScopeType**(java.lang.String scopeType)

> set property scopeType
> **Throws:**
> > RhapsodyRuntimeException

---

setStandardHeaders

void **setStandardHeaders**(java.lang.String standardHeaders)

> set property standardHeaders
> **Throws:**
> > RhapsodyRuntimeException

---

setStatechartImplementation

void **setStatechartImplementation**(java.lang.String statechartImplementation)

> set property statechartImplementation
> **Throws:**
> > RhapsodyRuntimeException

---

setTimeModel

void **setTimeModel**(java.lang.String timeModel)

> set property timeModel
> **Throws:**

com.telelogic.rhapsody.core

com.telelogic.rhapsody.core
Interface IRPConnector

All Superinterfaces:
>   IRPModelElement, IRPStateVertex

All Known Subinterfaces:
>   IRPPin

```
public interface IRPConnector
```
extends IRPStateVertex

The IRPConnector interface represents the characteristics shared by the various types of "connector" elements that can be included in a statechart, such as condition connectors, history connectors, join sync bar connectors, and fork sync bar connectors.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPTransition	**createDefaultTransition**(IRPState from) Creates a default transition leading to this connector, within the state specified.
java.lang.String	**getConnectorType**() Returns the type of the connector: Condition, Diagram, EnterExit, Fork, History, Join, Junction, Termination, InPin, OutPin, or InOutPin.
IRPCollection	**getDerivedInEdges**() Returns a collection of the transitions coming into the connector.
IRPTransition	**getDerivedOutEdge**() Returns the transition exiting the connector.
IRPSwimlane	**getItsSwimlane**() For connectors in a swimlane, returns the swimlane that contains the connector.
IRPState	**getOfState**() For history connectors, returns the state that the history connector belongs to.

Method Summary

int	**isConditionConnector**() Checks whether the connector is a condition connector.
int	**isDiagramConnector**() Checks whether the connector is a diagram connector.
int	**isForkConnector**() Checks whether the connector is a fork sync bar connector.
int	**isHistoryConnector**() Checks whether the connector is a history connector.
int	**isJoinConnector**() Checks whether the connector is a join sync bar connector.
int	**isJunctionConnector**() Checks whether the connector is a junction connector.
int	**isStubConnector**() Checks whether the connector is an EnterExit point.
int	**isTerminationConnector**() Checks whether the connector is a termination connector.
void	**setItsSwimlane**(IRPSwimlane pVal) Specifies the swimlane that should contain this connector.
void	**setOfState**(IRPState OfState) For history connectors, specifies the state for which the connector should maintain historical state information.

Methods inherited from interface com.telelogic.rhapsody.core.IRPStateVertex

addFlow, addTransition, deleteTransition, getInTransitions, getOutTransitions, getParent, setParent

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype,

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

createDefaultTransition

IRPTransition **createDefaultTransition**(IRPState from)

> Creates a default transition leading to this connector, within the state specified.
> **Parameters:**
> > from - the state for which the default transition should be created
> **Returns:**
> > the default transition that was created

---

getConnectorType

java.lang.String **getConnectorType**()

> Returns the type of the connector: Condition, Diagram, EnterExit, Fork, History, Join, Junction, Termination, InPin, OutPin, or InOutPin.
> **Returns:**
> > the type of the connector

---

getDerivedInEdges

IRPCollection **getDerivedInEdges**()

> Returns a collection of the transitions coming into the connector.
> **Returns:**
> > the transitions coming into the connector (a collection of IRPTransition elements)

---

getDerivedOutEdge

IRPTransition **getDerivedOutEdge**()

Returns the transition exiting the connector.
Returns:
> the transition exiting the connector

---

getItsSwimlane

IRPSwimlane **getItsSwimlane**()

For connectors in a swimlane, returns the swimlane that contains the connector.
Returns:
> the swimlane that contains the connector

---

getOfState

IRPState **getOfState**()

For history connectors, returns the state that the history connector belongs to. This is the state for which the history connector maintains historical state information.
Returns:
> the state that this history connector belongs to

---

isConditionConnector

int **isConditionConnector**()

Checks whether the connector is a condition connector.
Returns:
> 1 if the connector is a condition connector, 0 otherwise

---

isDiagramConnector

int **isDiagramConnector**()

Checks whether the connector is a diagram connector.
Returns:
> 1 if the connector is a diagram connector, 0 otherwise

---

isForkConnector

int **isForkConnector**()

Checks whether the connector is a fork sync bar connector.
Returns:
> 1 if the connector is a fork sync bar connector, 0 otherwise

---

isHistoryConnector

int **isHistoryConnector**()

Checks whether the connector is a history connector.
Returns:
1 if the connector is a history connector, 0 otherwise

isJoinConnector

int **isJoinConnector**()

Checks whether the connector is a join sync bar connector.
Returns:
1 if the connector is a join sync bar connector, 0 otherwise

isJunctionConnector

int **isJunctionConnector**()

Checks whether the connector is a junction connector.
Returns:
1 if the connector is a junction connector, 0 otherwise

isStubConnector

int **isStubConnector**()

Checks whether the connector is an EnterExit point. (Prior to version 6.0 of Rhapsody, EnterExit points were known as stub connectors.)
Returns:
1 if the connector is an EnterExit point, 0 otherwise

isTerminationConnector

int **isTerminationConnector**()

Checks whether the connector is a termination connector.
Returns:
1 if the connector is a termination connector, 0 otherwise

setItsSwimlane

void **setItsSwimlane**(<u>IRPSwimlane</u> pVal)

Specifies the swimlane that should contain this connector.
Parameters:
pVal - the swimlane that should contain this connector

---

setOfState

void **setOfState**(_IRPState_ OfState)

For history connectors, specifies the state for which the connector should maintain historical state information.
Parameters:
OfState - the state for which the connector should maintain historical state information

---

Package   **Class**   **Use**   **Tree**   **Serialized**   **Deprecated**   **Index**   **Help**

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

PREV CLASS　**NEXT CLASS**　　　　　　　　　**FRAMES**　**NO FRAMES**　　**All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD　　　　　DETAIL: FIELD | CONSTR | METHOD

---

com.telelogic.rhapsody.core
Interface IRPConstraint

All Superinterfaces:
　　　IRPAnnotation, IRPModelElement, IRPUnit

```
public interface IRPConstraint
```
extends IRPAnnotation

The IRPConstraint interface represents constraints in a Rational Rhapsody model.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPCollection	**getConstraintsByMe**() Returns all of the model elements affected by this constraint.

Methods inherited from interface com.telelogic.rhapsody.core.IRPAnnotation
addAnchor, getAnchoredByMe, getBody, getSpecification, getSpecificationRTF, isSpecificationRTF, removeAnchor, setBody, setSpecification, setSpecificationRTF

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit
copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage,

---

setOfState　　　　　　　　　　　　　　　　　　　　　　　　　　193

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getConstraintsByMe

IRPCollection **getConstraintsByMe**()

> Returns all of the model elements affected by this constraint.
> **Returns:**
> > a collection of all the model elements affected by this constraint.

---

com.telelogic.rhapsody.core
Interface IRPContextSpecification

All Superinterfaces:
> IRPModelElement, IRPValueSpecification

---

public interface **IRPContextSpecification**
extends IRPValueSpecification

The IRPContextSpecification interface represents the exact context of an object in a hierarchy. The context consists of two collections: 1) a collection of strings representing the model elements that constitute the full path to the element 2) a collection of the relevant indices for each of the model elements in the first collection. This makes it possible to point to a specific instance of the target model element when multiplicity is greater than one. The collection must consist of integers provided as strings.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPCollection	**getMultiplicities**() Returns the collection of the relevant indices for each of the model elements in the "value" collection.
IRPCollection	**getValue**() Returns the collection of strings that represents the model elements that constitute the full path to the element.
void	**setMultiplicities**(IRPCollection multiplicities) Specifies the collection of indices to use for the model elements in the "value" collection.
void	**setValue**(IRPCollection value) Specifies the collection of strings that represents the model elements that constitute the full path to the element.

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

getConstraintsByMe   195

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getMultiplicities

IRPCollection **getMultiplicities**()

> Returns the collection of the relevant indices for each of the model elements in the "value" collection. The collection consists of integers provided as strings.
> **Returns:**
> > the collection of the relevant indices for each of the model elements in the "value" collection

getValue

IRPCollection **getValue**()

> Returns the collection of strings that represents the model elements that constitute the full path to the element.
> **Returns:**
> > the collection of strings that represents the model elements that constitute the full path to the element

setMultiplicities

void **setMultiplicities**(IRPCollection multiplicities)

> Specifies the collection of indices to use for the model elements in the "value" collection. . The collection must consist of integers provided as strings.
> **Parameters:**
>> multiplicities - the collection of indices to use for the model elements in the "value" collection

setValue

void **setValue**(IRPCollection value)

> Specifies the collection of strings that represents the model elements that constitute the full path to the element.
> **Parameters:**
>> value - the collection of strings to use to represent the model elements that constitute the full path to the element

com.telelogic.rhapsody.core
Interface IRPControlledFile

All Superinterfaces:
> IRPModelElement, IRPUnit

---

public interface **IRPControlledFile**
extends **IRPUnit**

The IRPControlledFile interface represents a controlled file in a Rhapsody model. To access an element's controlled files, use the method IRPModelElement.getControlledFiles().

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

java.lang.String	**getFullPathFileName**() Returns the full path of the controlled file.
void	**open**() Opens the controlled file, using the associated program.
void	**setTarget**(java.lang.String filename) Specifies a different file to associate with the Controlled File element.

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit
copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype,

setValue   198

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getFullPathFileName

```
java.lang.String getFullPathFileName()
```

> Returns the full path of the controlled file.
> **Returns:**
> > the full path of the controlled file

open

```
void open()
```

> Opens the controlled file, using the associated program.

setTarget

```
void setTarget(java.lang.String filename)
```

Specifies a different file to associate with the Controlled File element. Note that this must be a file that already exists in the project directory.

Parameters:

`filename` - the file to associate with the Controlled File element - must be a file that already exists in the project directory

---

com.telelogic.rhapsody.core
Interface IRPDependency

All Superinterfaces:
> IRPModelElement

All Known Subinterfaces:
> IRPHyperLink

---

public interface **IRPDependency**
extends IRPModelElement

The IRPDependency interface represents dependencies in a Rational Rhapsody model.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPModelElement	**getDependent**() Returns the source element in the dependency relation, meaning the element that depends on the other element.
IRPModelElement	**getDependsOn**() Returns the target element in the dependency relation, meaning the element on which the first element depends.
void	**setDependent**(IRPModelElement dependent) Sets the source element in the dependency relation, meaning the element that depends on the other element.
void	**setDependsOn**(IRPModelElement dependsOn) Sets the target element in the dependency relation, meaning the element on which the first element depends
void	**setOwnerWithoutChangingDependent**(IRPModelElement newOwner) Specifies a new owner for the dependency, without changing the dependent model element.

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getDependent

IRPModelElement **getDependent**()

> Returns the source element in the dependency relation, meaning the element that depends on the other element.
> **Returns:**
> > the source element in the dependency relation

getDependsOn

IRPModelElement **getDependsOn**()

> Returns the target element in the dependency relation, meaning the element on which the first element depends.
> **Returns:**
> > the target element in the dependency relation

setDependent

void **setDependent**([IRPModelElement](#) dependent)

> Sets the source element in the dependency relation, meaning the element that depends on the other element.
>
> **Parameters:**
>> dependent - the model element that should be used as the source element in the dependency relation

---

setDependsOn

void **setDependsOn**([IRPModelElement](#) dependsOn)

> Sets the target element in the dependency relation, meaning the element on which the first element depends
>
> **Parameters:**
>> dependsOn - the model element that should be used as the target element in the dependency relation

---

setOwnerWithoutChangingDependent

void **setOwnerWithoutChangingDependent**([IRPModelElement](#) newOwner)

> Specifies a new owner for the dependency, without changing the dependent model element. Note that if you call the method IRPModelElement.setOwner on a dependency, it will change both the owner and the dependent element.
>
> **Parameters:**
>> newOwner - the model element that should be the owner of the dependency
>
> **Throws:**
>> [RhapsodyRuntimeException](#)

[Package](#)   **Class**   [**Use**](#)   [**Tree**](#)   [**Serialized**](#)   [**Deprecated**](#)   [**Index**](#)   [**Help**](#)

[**PREV CLASS**](#) [**NEXT CLASS**](#)   [**FRAMES**](#)   [**NO FRAMES**](#)   [**All Classes**](#)

SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#)   DETAIL: FIELD | CONSTR | [METHOD](#)

com.telelogic.rhapsody.core
Interface IRPDeploymentDiagram

All Superinterfaces:
　　　IRPDiagram, IRPModelElement, IRPUnit

---

public interface **IRPDeploymentDiagram**
extends IRPDiagram

The IRPDeploymentDiagram interface represents deployment diagrams in Rational Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement

IRPModelElement.OSLCLink

Method Summary

Methods inherited from interface com.telelogic.rhapsody.core.IRPDiagram

addFreeShapeByType, addImage, addNewEdgeByType, addNewEdgeForElement, addNewNodeByType, addNewNodeForElement, addTextBox, closeDiagram, completeRelations, getCorrespondingGraphicElements, getElementsInDiagram, getGraphicalElements, getLastVisualizationModifiedTime, getPicture, getPictureAs, getPictureAsDividedMetafiles, getPictureEx, getPicturesWithImageMap, isOpen, isShowDiagramFrame, openDiagram, openDiagramView, populateDiagram, removeGraphElements, setShowDiagramFrame

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype,

---

setOwnerWithoutChangingDependent　　　　　　　　　　　　　　　　　　　　204

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

---

com.telelogic.rhapsody.core
Interface IRPDestructionEvent

All Superinterfaces:
> IRPMessage, IRPModelElement

---

public interface **IRPDestructionEvent**
extends IRPMessage

The IRPDestructionEvent interface represents destruction events in sequence diagrams.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

Methods inherited from interface com.telelogic.rhapsody.core.IRPMessage
addSourceExecutionOccurrence, addTargetExecutionOccurrence, getActualParameterList, getCommunicationConnection, getCondition, getDurationConstraint, getDurationObservation, getFlowPort, getFormalInterfaceItem, getFormalType, getInvariant, getMessageType, getPort, getReturnValue, getSequenceNumber, getSignature, getSource, getSourceExecutionOccurrence, getTarget, getTargetExecutionOccurrence, getTimeConstraint, getTimeObservation, getTimerValue, reroute, setActualParameterList, setDurationConstraint, setDurationObservation, setFlowPort, setFormalInterfaceItem, setFormalType, setInvariant, setPort, setReturnValue, setTimeConstraint, setTimeObservation, setTimerValue

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal,

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

---

 com.telelogic.rhapsody.core
Interface IRPDiagram

All Superinterfaces:
> IRPModelElement, IRPUnit

All Known Subinterfaces:
> IRPActivityDiagram, IRPCollaborationDiagram, IRPComponentDiagram, IRPDeploymentDiagram, IRPObjectModelDiagram, IRPPanelDiagram, IRPSequenceDiagram, IRPStatechartDiagram, IRPStructureDiagram, IRPTimingDiagram, IRPUseCaseDiagram

---

public interface **IRPDiagram**
extends IRPUnit

The IRPDiagram interface contains the methods shared by all the interfaces that represent specific types of diagrams.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPGraphElement	**addFreeShapeByType**(java.lang.String metaType, IRPCollection xPoints, IRPCollection yPoints) Adds a free shape of the type specified, using the x coordinates and y coordinates provided.
IRPGraphElement	**addImage**(java.lang.String filename, int xPosition, int yPosition, int nWidth, int nHeight) Adds an image to the diagram, using the specified file, starting point, width, and height.
IRPGraphEdge	**addNewEdgeByType**(java.lang.String metaType, IRPGraphElement src, int xSrcPosition, int ySrcPosition, IRPGraphElement trg, int xTrgPosition, int yTrgPosition) Adds a connector element of the specified type to the diagram, using the source and target elements specified.
IRPGraphEdge	

Method Summary

	addNewEdgeForElement([IRPModelElement](#) element, [IRPGraphNode](#) src, int xSrcPosition, int ySrcPosition, [IRPGraphNode](#) trg, int xTrgPosition, int yTrgPosition) Adds a connector graphical element to the diagram to represent the specified model element.
[IRPGraphNode](#)	**addNewNodeByType**(java.lang.String metaType, int xPosition, int yPosition, int nWidth, int nHeight) Adds a diagram element of the specified type to the diagram, using the position and dimensions specified.
[IRPGraphNode](#)	**addNewNodeForElement**([IRPModelElement](#) element, int xPosition, int yPosition, int nWidth, int nHeight) Adds a graphical element to the diagram to represent the specified model element.
[IRPGraphElement](#)	**addTextBox**(java.lang.String text, int xPosition, int yPosition, int nWidth, int nHeight) Adds a text box using the specified text, starting point, width, and height.
void	**closeDiagram**() Closes the diagram.
void	**completeRelations**([IRPCollection](#) graphElements, int selectedToAll) Adds connectors to the diagram to reflect the existing relations between the specified elements.
[IRPCollection](#)	**getCorrespondingGraphicElements**([IRPModelElement](#) modelElement) Returns the graphical elements that represent the specified model element in the diagram.
[IRPCollection](#)	**getElementsInDiagram**() Returns a collection of all the model elements in the diagram.
[IRPCollection](#)	**getGraphicalElements**() Returns a collection of all the graphical elements in the diagram.
java.lang.String	**getLastVisualizationModifiedTime**() Returns the time at which the visual representation of the diagram was last changed.
void	**getPicture**(java.lang.String filename) Saves the diagram as an emf format file, using the path and filename provided as a parameter.
[IRPCollection](#)	**getPictureAs**(java.lang.String firstFileName, java.lang.String imageFormat, int getImageMaps, [IRPCollection](#) diagrammap) Saves the diagram in the specified graphic format, breaking the diagram into a number of files if necessary.
[IRPCollection](#)	**getPictureAsDividedMetafiles**(java.lang.String firstFileName) Saves the diagram as an emf format file, breaking the diagram into a number of such files if necessary.
void	**getPictureEx**(java.lang.String filename, java.lang.String exportScale, int smartZoom)

Method Summary

	method getPictureEx
[IRPCollection](#)	**getPicturesWithImageMap**(java.lang.String firstFileName, [IRPCollection](#) diagrammap) Saves the diagram as an emf format file, breaking the diagram into a number of files if necessary.
int	**isOpen**() method isOpen
int	**isShowDiagramFrame**() Checks whether the diagram frame is currently visible.
void	**openDiagram**() Opens the diagram.
[IRPAXViewCtrl](#)	**openDiagramView**() Used internally by Rational Rhapsody to display diagrams within Eclipse (when using the Rhapsody-Eclipse platform integration).
void	**populateDiagram**([IRPCollection](#) elementsToPopulate, [IRPCollection](#) relationsTypes, java.lang.String createContent) Populates the diagram with the elements and types of relations specified.
void	**removeGraphElements**([IRPCollection](#) elementsToRemove) Removes the specified graphic elements from the diagram.
void	**setShowDiagramFrame**(int bShow) Shows/hides the diagram frame.

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue,

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addFreeShapeByType

IRPGraphElement **addFreeShapeByType**(java.lang.String metaType,
                                    IRPCollection xPoints,
                                    IRPCollection yPoints)

Adds a free shape of the type specified, using the x coordinates and y coordinates provided.
Parameters:
    metaType - the type of shape to add. The possible values for this parameter are: "Polyline", "Polygon", "Rectangle", "Polycurve", "Closed Curve", "Ellipse".
    xPoints - collection of integers representing the x coordinates for the shape
    yPoints - collection of integers representing the y coordinates for the shape
Returns:
    the new shape that was created

addImage

IRPGraphElement **addImage**(java.lang.String filename,
                        int xPosition,
                        int yPosition,
                        int nWidth,
                        int nHeight)

Adds an image to the diagram, using the specified file, starting point, width, and height.
Parameters:
    filename - the full path to the image
    xPosition - the x coordinate for the top left corner of the image, in pixels
    yPosition - the y coordinate for the top left corner of the image, in pixels
    nWidth - the width of the image, in pixels
    nHeight - the height of the image, in pixels
Returns:
    the new image element that was created

addNewEdgeByType

IRPGraphEdge **addNewEdgeByType**(java.lang.String metaType,
                              IRPGraphElement src,
                              int xSrcPosition,
                              int ySrcPosition,
                              IRPGraphElement trg,
                              int xTrgPosition,
                              int yTrgPosition)

Adds a connector element of the specified type to the diagram, using the source and target elements specified. Note that this method can only be used for connector elements that only have graphical representations and are not actual elements in the model. "Ordinary" connector elements are added to a diagram by carrying out two steps: 1) adding the new element to your model 2) adding a graphical representation of the element to the diagram using the method IRPDiagram.addNewEdgeForElement.

Parameters:

metaType - the type of connector element to add to the diagram. The strings that can be used for this parameter are: "anchor", "compRealization", "Containment Arrow", and "communication path".

src - the graphical element that is the source for the connector

xSrcPosition - the distance, in pixels, from the the left edge of the diagram to a point within the source graphical element

ySrcPosition - the distance, in pixels, from the the top edge of the diagram to a point within the source graphical element

trg - the graphical element that is the target for the connector

xTrgPosition - the distance, in pixels, from the the left edge of the diagram to a point within the target graphical element

yTrgPosition - the distance, in pixels, from the the top edge of the diagram to a point within the target graphical element

Returns:

the graphical element that was added to the diagram

See Also:

addNewEdgeForElement(com.telelogic.rhapsody.core.IRPModelElement,
com.telelogic.rhapsody.core.IRPGraphNode, int, int,
com.telelogic.rhapsody.core.IRPGraphNode, int, int)

addNewEdgeForElement

IRPGraphEdge **addNewEdgeForElement**(IRPModelElement element,
                              IRPGraphNode src,
                              int xSrcPosition,
                              int ySrcPosition,
                              IRPGraphNode trg,
                              int xTrgPosition,
                              int yTrgPosition)

Adds a connector graphical element to the diagram to represent the specified model element.

Parameters:

element - the model element to add to the diagram.

src - the graphical element that is the source for the connector

`xSrcPosition` - the distance, in pixels, from the the left edge of the diagram to a point within the source graphical element

`ySrcPosition` - the distance, in pixels, from the the top edge of the diagram to a point within the source graphical element

`trg` - the graphical element that is the target for the connector

`xTrgPosition` - the distance, in pixels, from the the left edge of the diagram to a point within the target graphical element

`yTrgPosition` - the distance, in pixels, from the the top edge of the diagram to a point within the target graphical element

Returns:

the connector graphical element that was added to the diagram

addNewNodeByType

[IRPGraphNode](#) **addNewNodeByType**(java.lang.String metaType,
                              int xPosition,
                              int yPosition,
                              int nWidth,
                              int nHeight)

Adds a diagram element of the specified type to the diagram, using the position and dimensions specified. Note that this method can only be used for diagram elements that only have graphical representations and are not actual elements in the model. "Ordinary" model elements are added to a diagram by carrying out two steps: 1) adding the new element to your model 2) adding a graphical representation of the element to the diagram using the method IRPDiagram.addNewNodeForElement.

Parameters:

`metaType` - the type of element to add to the diagram. The strings that can be used for this parameter are: "OrState"(for And Line), "Interaction Operand", "Swimlane" (for swimlane divider), "System Border", "PartitionLine", "SDActionBlock" (for action block in sequence diagram), "Note"; panel diagram elements: "Knob", "Gauge", "Meter", "LevelIndicator", "MatrixDisplay", "DigitalDisplay", "Led", "OnOffSwitch", "PushButton", "ButtonArray", "TextBox", "Slider"; free shapes: "Polyline", "Ploygon", "Rectangle", "Polycurve", "Closed Curve","Ellipse","Image".

`xPosition` - the position of the left edge of the graphical object, in pixels, relative to the left edge of the diagram

`yPosition` - the position of the top edge of the graphical object, in pixels, relative to the top edge of the diagram

`nWidth` - the width of the graphical object

`nHeight` - the height of the graphical object

Returns:

the graphical element that was added to the diagram

See Also:

[addNewNodeForElement(com.telelogic.rhapsody.core.IRPModelElement, int, int, int, int)](#)

addNewNodeForElement

[IRPGraphNode](#) **addNewNodeForElement**([IRPModelElement](#) element,
                                int xPosition,
                                int yPosition,

```
                    int nWidth,
                    int nHeight)
```

Adds a graphical element to the diagram to represent the specified model element. For connector elements, use the method addNewEdgeForElement.

Parameters:

element - the model element to add to the diagram.

xPosition - the position of the left edge of the graphical object, in pixels, relative to the left edge of the diagram

yPosition - the position of the top edge of the graphical object, in pixels, relative to the top edge of the diagram

nWidth - the width of the graphical object

nHeight - the height of the graphical object

Returns:

the graphical element that was added to the diagram

addTextBox

IRPGraphElement **addTextBox**(java.lang.String text,
                    int xPosition,
                    int yPosition,
                    int nWidth,
                    int nHeight)

Adds a text box using the specified text, starting point, width, and height.

Parameters:

text - the text that should be displayed

xPosition - the x coordinate for the top left corner of the box, in pixels

yPosition - the y coordinate for the top left corner of the box, in pixels

nWidth - the width of the text box, in pixels

nHeight - the height of the text box, in pixels

Returns:

the new text box that was created

openDiagramView

IRPAXViewCtrl **openDiagramView**()

Used internally by Rational Rhapsody to display diagrams within Eclipse (when using the Rhapsody-Eclipse platform integration).

closeDiagram

void **closeDiagram**()

Closes the diagram.

completeRelations

```
void completeRelations(IRPCollection graphElements,
                       int selectedToAll)
```

Adds connectors to the diagram to reflect the existing relations between the specified elements.
Parameters:
graphElements - the elements whose relations should be reflected on the diagram
selectedToAll - Use 0 if you just want to display the relations between the specified elements. Use 1 if you would also like to display any existing relations between the specified elements and other elements on the diagram.

getCorrespondingGraphicElements

```
IRPCollection getCorrespondingGraphicElements(IRPModelElement modelElement)
```

Returns the graphical elements that represent the specified model element in the diagram. In cases where the same model element appears multiple times in a single diagram, the collection returned will contain more than one graphical element.
Parameters:
modelElement - the model element in the diagram whose graphical elements should be returned
Returns:
the graphical elements that represent the specified model element in the diagram

getElementsInDiagram

```
IRPCollection getElementsInDiagram()
```

Returns a collection of all the model elements in the diagram.
Returns:
collection of all the model elements in the diagram

getGraphicalElements

```
IRPCollection getGraphicalElements()
```

Returns a collection of all the graphical elements in the diagram.
Returns:
collection of IRPGraphElement objects, representing all the graphical elements in the diagram.

getLastVisualizationModifiedTime

```
java.lang.String getLastVisualizationModifiedTime()
```

Returns the time at which the visual representation of the diagram was last changed. This takes into account not only the information stored in the diagram element itself, but also information from other elements that is reflected on the diagram, for example, changes to an attribute of a class that is included in the diagram.

Returns:
> the time at which the visual representation of the diagram was last changed

getPicture

void **getPicture**(java.lang.String filename)

> Saves the diagram as an emf format file, using the path and filename provided as a parameter.
> **Parameters:**
> > filename - the full path to use for saving the file

getPictureAs

IRPCollection **getPictureAs**(java.lang.String firstFileName,
                              java.lang.String imageFormat,
                              int getImageMaps,
                              IRPCollection diagrammap)

> Saves the diagram in the specified graphic format, breaking the diagram into a number of files if necessary. The need to break the diagram into a number of files is based on the value of the property General:Graphics:ExportedDiagramScale. If the property is set to a value other than FitToOnePage, more than one file will be created. In addition, this method can be used to retrieve diagram element information that can be used to create an HTML image map.
> **Parameters:**
> > firstFileName - the name to use for the file created. If more than one file is created, the filenames used will be based on the following convention: firstFileNameZ_X_Y, where Z is the number of the created file, X is the number of the page along the X vector, and Y is the number of the page along the Y vector.
> > imageFormat - the graphic format in which the diagram should be saved. This can be one of the following: EMF, BMP, JPEG, JPG, TIFF.
> > getImageMaps - use this argument to indicate whether the method should also provide a collection of IRPImageMap objects that can be used to construct an HTML image map for the diagram. (Use 1 if you want this information, else use 0.)
> > diagrammap - The collection to use to store the IRPImageMap objects containing the required information for constructing an HTML image map
> **Returns:**
> > collection that contains the names of the files that were created

getPictureAsDividedMetafiles

IRPCollection **getPictureAsDividedMetafiles**(java.lang.String firstFileName)

> Saves the diagram as an emf format file, breaking the diagram into a number of such files if necessary. The need to break the diagram into a number of files is based on the value of the property

General:Graphics:ExportedDiagramScale. If the property is set to a value other than FitToOnePage, more than one file will be created.

Parameters:

firstFileName - the name to use for the first file created. If more than one file is created, the filenames used will be based on the following convention: firstFileNameZ_X_Y, where Z is the number of the created file, X is the number of the page along the X vector, and Y is the number of the page along the Y vector.

Returns:

collection that contains the names of the files that were created

getPictureEx

```
void getPictureEx(java.lang.String filename,
                  java.lang.String exportScale,
                  int smartZoom)
```

method getPictureEx
Throws:
RhapsodyRuntimeException

getPicturesWithImageMap

```
IRPCollection getPicturesWithImageMap(java.lang.String firstFileName,
                                      IRPCollection diagrammap)
```

Saves the diagram as an emf format file, breaking the diagram into a number of files if necessary. The need to break the diagram into a number of files is based on the value of the property General:Graphics:ExportedDiagramScale. If the property is set to a value other than FitToOnePage, more than one file will be created. In addition, this method retrieves diagram element information that can be used to create an HTML image map.

Parameters:

firstFileName - the name to use for the file created. If more than one file is created, the filenames used will be based on the following convention: firstFileNameZ_X_Y, where Z is the number of the created file, X is the number of the page along the X vector, and Y is the number of the page along the Y vector.

diagrammap - The collection to use to store the IRPImageMap objects containing the required information for constructing an HTML image map

Returns:

collection that contains the names of the files that were created

isOpen

```
int isOpen()
```

method isOpen
Throws:
RhapsodyRuntimeException

isShowDiagramFrame

```
int isShowDiagramFrame()
```

Checks whether the diagram frame is currently visible.
Returns:
1 if the diagram frame is currently visible, 0 if it is not visible

openDiagram

```
void openDiagram()
```

Opens the diagram.

populateDiagram

```
void populateDiagram(IRPCollection elementsToPopulate,
                     IRPCollection relationsTypes,
                     java.lang.String createContent)
```

Populates the diagram with the elements and types of relations specified.
Parameters:
elementsToPopulate - the elements (nodes) to add to the diagram
relationsTypes - the types of relations that should be drawn on the diagram. You can use the string AllRelations to display all types, or use any combination of the following strings: Composition, Association, Link, Dependency, Inheritance, Anchor, InformationFlow
createContent - the elements that should be included in addition to those specified. This argument can take any of the following strings: among, from, to, fromto. If you use "among", only the elements you specified will be included. If you use one of the other strings, the diagram will also include elements that the selected elements are related to

```
 IRPApplication app = RhapsodyAppServer.getActiveRhapsodyApplication();
 IRPProject project = app.activeProject();
 IRPCollection packages = project.getPackages();
 IRPCollection relTypes = app.createNewCollection();
 relTypes.setSize(3);
 relTypes.setString(1, "Composition");
 relTypes.setString(2, "Association");
 relTypes.setString(3, "Dependency");
 IRPObjectModelDiagram diagram2 = project.addObjectModelDiagram("PopulateTest1");
 diagram2.populateDiagram(packages, relTypes, "fromto");
```

removeGraphElements

```
void removeGraphElements(IRPCollection elementsToRemove)
```

Removes the specified graphic elements from the diagram.
Parameters:

isShowDiagramFrame   218

> `elementsToRemove` - a collection of IRPGraphElement objects, representing the graphic elements that should be removed from the diagram

setShowDiagramFrame

void **setShowDiagramFrame**(int bShow)

> Shows/hides the diagram frame.
> **Parameters:**
>> `bShow` - use 1 to show the diagram frame, 0 to hide the frame.

com.telelogic.rhapsody.core
Interface IRPDiagSynthAPI

public interface **IRPDiagSynthAPI**

Method Summary

long	**addInstance**(long addedToSD, java.lang.String instanceNavExp) DiagSynthAPI : add instance to sequence diagram
int	**addSynthSDToModel2**(IRPSequenceDiagram pMscOrig, long synthSD, int openSD) DiagSynthAPI : add synth sequence diagarm to model
long	**createSD2**(IRPSequenceDiagram pMscOrig, java.lang.String testedmscname) DiagSynthAPI : create sequence diagram
java.lang.String	**getInterfaceName**() get property interfaceName
void	**receiveMessage**(long pTestedSD, long pEventSent) DiagSynthAPI : recieve sequence diagram message
int	**removeSynthSDToModel2**(IRPSequenceDiagram pMscOrig) DiagSynthAPI : remove synth sequence diagarm to model
long	**sDAddConditionMark**(long pTestedSD, java.lang.String instance, java.lang.String text, java.lang.String type) DiagSynthAPI : send condition mark to instance
long	**sendMessage**(long pTestedSD, java.lang.String source, java.lang.String target, java.lang.String event, java.lang.String operation, java.lang.String type) DiagSynthAPI : send sequence diagram message

Method Detail

addInstance

```
long addInstance(long addedToSD,
                 java.lang.String instanceNavExp)
```

setShowDiagramFrame   220

DiagSynthAPI : add instance to sequence diagram
Throws:
> RhapsodyRuntimeException

addSynthSDToModel2

```
int addSynthSDToModel2(IRPSequenceDiagram pMscOrig,
                       long synthSD,
                       int openSD)
```

> DiagSynthAPI : add synth sequence diagarm to model
> **Throws:**
> > RhapsodyRuntimeException

createSD2

```
long createSD2(IRPSequenceDiagram pMscOrig,
               java.lang.String testedmscname)
```

> DiagSynthAPI : create sequence diagram
> **Throws:**
> > RhapsodyRuntimeException

receiveMessage

```
void receiveMessage(long pTestedSD,
                    long pEventSent)
```

> DiagSynthAPI : recieve sequence diagram message
> **Throws:**
> > RhapsodyRuntimeException

removeSynthSDToModel2

```
int removeSynthSDToModel2(IRPSequenceDiagram pMscOrig)
```

> DiagSynthAPI : remove synth sequence diagarm to model
> **Throws:**
> > RhapsodyRuntimeException

sDAddConditionMark

```
long sDAddConditionMark(long pTestedSD,
                        java.lang.String instance,
                        java.lang.String text,
                        java.lang.String type)
```

DiagSynthAPI : send condition mark to instance
Throws:
   [RhapsodyRuntimeException](#)

---

sendMessage

```
long sendMessage(long pTestedSD,
                 java.lang.String source,
                 java.lang.String target,
                 java.lang.String event,
                 java.lang.String operation,
                 java.lang.String type)
```

  DiagSynthAPI : send sequence diagram message
  Throws:
   [RhapsodyRuntimeException](#)

---

getInterfaceName

```
java.lang.String getInterfaceName()
```

  get property interfaceName
  Throws:
   [RhapsodyRuntimeException](#)

---

com.telelogic.rhapsody.core
Interface IRPEnumerationLiteral

All Superinterfaces:
> IRPModelElement

---

public interface **IRPEnumerationLiteral**
extends IRPModelElement

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

java.lang.String	**getValue**() get property value
void	**setValue**(java.lang.String value) set property value

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML,

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getValue

java.lang.String **getValue**()

>   get property value
>   **Throws:**
>>   RhapsodyRuntimeException

---

setValue

void **setValue**(java.lang.String value)

>   set property value
>   **Throws:**
>>   RhapsodyRuntimeException

---

com.telelogic.rhapsody.core
Interface IRPEvent

All Superinterfaces:
> IRPClassifier, IRPInterfaceItem, IRPModelElement, IRPUnit

---

public interface **IRPEvent**
extends IRPInterfaceItem

The IRPEvent interface represents events in Rational Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPEvent	**getBaseEvent**() get property baseEvent
IRPEvent	**getSuperEvent**() get property baseEvent
void	**setBaseEvent**(IRPEvent baseEvent) set property baseEvent
void	**setSuperEvent**(IRPEvent superEvent) set property baseEvent

Methods inherited from interface com.telelogic.rhapsody.core.IRPInterfaceItem
addArgument, addArgumentBeforePosition, getArguments, getSignature, getSignatureNoArgNames, getSignatureNoArgTypes, matchOnSignature

Methods inherited from interface com.telelogic.rhapsody.core.IRPClassifier
addActivityDiagram, addAttribute, addFlowItems, addFlows, addGeneralization, addOperation, addRelation, addRelationTo, addStatechart, addUnidirectionalRelation, addUnidirectionalRelationTo, deleteAttribute, deleteFlowItems, deleteFlows, deleteGeneralization, deleteOperation, deleteRelation, findAttribute,

Methods inherited from interface com.telelogic.rhapsody.core.IRPClassifier

findBaseClassifier, findDerivedClassifier, findGeneralization, findInterfaceItem, findNestedClassifier, findNestedClassifierRecursive, findRelation, findTrigger, getActivityDiagram, getAttributes, getAttributesIncludingBases, getBaseClassifiers, getBehavioralDiagrams, getDerivedClassifiers, getFlowItems, getFlows, getGeneralizations, getInterfaceItems, getInterfaceItemsIncludingBases, getLinks, getNestedClassifiers, getOperations, getPorts, getRelations, getRelationsIncludingBases, getSequenceDiagrams, getSourceArtifacts, getStatechart

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getBaseEvent

<u>IRPEvent</u> **getBaseEvent**()

> get property baseEvent
> **Throws:**
>> <u>RhapsodyRuntimeException</u>

getSuperEvent

<u>IRPEvent</u> **getSuperEvent**()

> get property baseEvent
> **Throws:**
>> <u>RhapsodyRuntimeException</u>

setBaseEvent

void **setBaseEvent**(<u>IRPEvent</u> baseEvent)

> set property baseEvent
> **Throws:**
>> <u>RhapsodyRuntimeException</u>

setSuperEvent

void **setSuperEvent**(<u>IRPEvent</u> superEvent)

> set property baseEvent
> **Throws:**
>> <u>RhapsodyRuntimeException</u>

com.telelogic.rhapsody.core
Interface IRPEventReception

All Superinterfaces:
> IRPClassifier, IRPInterfaceItem, IRPModelElement, IRPUnit

```
public interface IRPEventReception
```
extends IRPInterfaceItem

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPEvent	**getEvent**() method getEvent
void	**setEvent**(IRPEvent pVal) method setEvent

Methods inherited from interface com.telelogic.rhapsody.core.IRPInterfaceItem
addArgument, addArgumentBeforePosition, getArguments, getSignature, getSignatureNoArgNames, getSignatureNoArgTypes, matchOnSignature

Methods inherited from interface com.telelogic.rhapsody.core.IRPClassifier
addActivityDiagram, addAttribute, addFlowItems, addFlows, addGeneralization, addOperation, addRelation, addRelationTo, addStatechart, addUnidirectionalRelation, addUnidirectionalRelationTo, deleteAttribute, deleteFlowItems, deleteFlows, deleteGeneralization, deleteOperation, deleteRelation, findAttribute, findBaseClassifier, findDerivedClassifier, findGeneralization, findInterfaceItem, findNestedClassifier, findNestedClassifierRecursive, findRelation, findTrigger, getActivityDiagram, getAttributes, getAttributesIncludingBases, getBaseClassifiers, getBehavioralDiagrams, getDerivedClassifiers, getFlowItems, getFlows, getGeneralizations, getInterfaceItems, getInterfaceItemsIncludingBases, getLinks, getNestedClassifiers, getOperations, getPorts, getRelations, getRelationsIncludingBases, getSequenceDiagrams, getSourceArtifacts, getStatechart

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getEvent

IRPEvent **getEvent**()

> method getEvent
> **Throws:**
> > RhapsodyRuntimeException

setEvent

```
void setEvent(IRPEvent pVal)
```

> method setEvent
> **Throws:**
> > RhapsodyRuntimeException

---

com.telelogic.rhapsody.core
Interface IRPExecutionOccurrence

All Superinterfaces:
    IRPModelElement

---

public interface **IRPExecutionOccurrence**
extends IRPModelElement

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPMessage	**getMessage**() get property message

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines,

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getMessage

IRPMessage **getMessage**()

get property message
Throws:
  RhapsodyRuntimeException

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPExternalCheckRegistry

public interface **IRPExternalCheckRegistry**

Method Summary

void	**appendFailedElementsComments**(java.lang.String strVal) method appendFailedElementsComments
java.lang.String	**getInterfaceName**() get property interfaceName
void	**setFailedElementsComments**(java.lang.String strVal) method setFailedElementsComments

Method Detail

appendFailedElementsComments

void **appendFailedElementsComments**(java.lang.String strVal)

method appendFailedElementsComments
Throws:
    RhapsodyRuntimeException

getInterfaceName

java.lang.String **getInterfaceName**()

get property interfaceName
Throws:
    RhapsodyRuntimeException

setFailedElementsComments

void **setFailedElementsComments**(java.lang.String strVal)

method setFailedElementsComments
Throws:
RhapsodyRuntimeException

---

com.telelogic.rhapsody.core
Interface IRPExternalCodeGeneratorInvoker

All Superinterfaces:
> IRPBaseExternalCodeGeneratorTool

---

public interface **IRPExternalCodeGeneratorInvoker**
extends IRPBaseExternalCodeGeneratorTool

Method Summary

java.lang.String	**getInterfaceName**() get property interfaceName
void	**notifyGenerationDone**() method notifyGenerationDone

Methods inherited from interface com.telelogic.rhapsody.core.IRPBaseExternalCodeGeneratorTool
advanceCodeGenProgressBar, shouldAbortCodeGeneration, writeCodeGenMessage

Method Detail

getInterfaceName

java.lang.String **getInterfaceName**()

> get property interfaceName
> **Throws:**
> > RhapsodyRuntimeException

---

notifyGenerationDone

void **notifyGenerationDone**()

> method notifyGenerationDone
> **Throws:**
> > RhapsodyRuntimeException

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPExternalIDERegistry

```
public interface IRPExternalIDERegistry
```

Method Summary

java.lang.String	**getInterfaceName**() get property interfaceName
void	**progressTaskAsynchCallback**(int nGroupNumber, int nTaskNumber) Initiate Progress Task execution
void	**progressTaskAsynchEliminate**(int nGroupNumber, int nTaskNumber) Initiate Progress Task execution
void	**sendIDETextMessage**(java.lang.String message) method SendIDETextMessage

Method Detail

progressTaskAsynchCallback

```
void progressTaskAsynchCallback(int nGroupNumber,
                                int nTaskNumber)
```

> Initiate Progress Task execution
> **Throws:**
>   RhapsodyRuntimeException

---

progressTaskAsynchEliminate

```
void progressTaskAsynchEliminate(int nGroupNumber,
                                 int nTaskNumber)
```

> Initiate Progress Task execution
> **Throws:**
>   RhapsodyRuntimeException

---

sendIDETextMessage

void **sendIDETextMessage**(java.lang.String message)

> method SendIDETextMessage
> **Throws:**
> > [RhapsodyRuntimeException](#)

getInterfaceName

java.lang.String **getInterfaceName**()

> get property interfaceName
> **Throws:**
> > [RhapsodyRuntimeException](#)

com.telelogic.rhapsody.core
Interface IRPExternalRoundtripInvoker

```
public interface IRPExternalRoundtripInvoker
```

Method Summary

java.lang.String	**getInterfaceName**() get property interfaceName

Method Detail

getInterfaceName

```
java.lang.String getInterfaceName()
```

get property interfaceName

Throws:

RhapsodyRuntimeException

 com.telelogic.rhapsody.core
Interface IRPFile

All Superinterfaces:
> IRPModelElement, IRPUnit

```
public interface IRPFile
```
extends IRPUnit

The IRPFile interface represents a file or folder to be generated during code generation.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

void	**addElement**(IRPClassifier element, java.lang.String fileFragmentType) method addElement Choose from = undefFragment, textFragment, implFragment, specFragment, moduleFragment
void	**addModelElement**(IRPModelElement element, java.lang.String fileFragmentType) method addModelElement Choose from = undefFragment, textFragment, implFragment, specFragment, moduleFragment
void	**addPackageToScope**(IRPPackage p) method addPackageToScope
void	**addTextElement**(java.lang.String text) method addTextElement
void	**addToScope**(IRPClassifier element) method addToScope
IRPCollection	**getElements**() get property elements
IRPCollection	**getFileFragments**() get property fileFragments

Method Summary

[IRPCollection](#)	**getFiles**() get property files
java.lang.String	**getFileType**() get property fileType
java.lang.String	**getImpName**(int includingPath) method getImpName
java.lang.String	**getPath**(int fullPath) get property path
java.lang.String	**getSpecName**(int includingPath) method getSpecName
int	**isEmpty**() method isEmpty
void	**setFileType**(java.lang.String fileType) set property fileType
void	**setPath**(java.lang.String path) property setPath

Methods inherited from interface com.telelogic.rhapsody.core.**IRPUnit**

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

---

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser,

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addElement

void **addElement**(IRPClassifier element,
                    java.lang.String fileFragmentType)

method addElement Choose from = undefFragment, textFragment, implFragment, specFragment, moduleFragment
Throws:
        RhapsodyRuntimeException

addModelElement

void **addModelElement**(IRPModelElement element,
                         java.lang.String fileFragmentType)

method addModelElement Choose from = undefFragment, textFragment, implFragment, specFragment, moduleFragment
Throws:
        RhapsodyRuntimeException

addPackageToScope

void **addPackageToScope**(IRPPackage p)

method addPackageToScope
Throws:
        RhapsodyRuntimeException

addTextElement

void **addTextElement**(java.lang.String text)

method addTextElement
Throws:

addToScope

void **addToScope**(IRPClassifier element)

> method addToScope
> **Throws:**
> > RhapsodyRuntimeException

getElements

IRPCollection **getElements**()

> get property elements
> **Throws:**
> > RhapsodyRuntimeException

getFileFragments

IRPCollection **getFileFragments**()

> get property fileFragments
> **Throws:**
> > RhapsodyRuntimeException

getFileType

java.lang.String **getFileType**()

> get property fileType
> **Throws:**
> > RhapsodyRuntimeException

getFiles

IRPCollection **getFiles**()

> get property files
> **Throws:**
> > RhapsodyRuntimeException

getImpName

java.lang.String **getImpName**(int includingPath)

> method getImpName
> **Throws:**
> > [RhapsodyRuntimeException](#)

getPath

java.lang.String **getPath**(int fullPath)

> get property path
> **Throws:**
> > [RhapsodyRuntimeException](#)

getSpecName

java.lang.String **getSpecName**(int includingPath)

> method getSpecName
> **Throws:**
> > [RhapsodyRuntimeException](#)

isEmpty

int **isEmpty**()

> method isEmpty
> **Throws:**
> > [RhapsodyRuntimeException](#)

setFileType

void **setFileType**(java.lang.String fileType)

> set property fileType
> **Throws:**
> > [RhapsodyRuntimeException](#)

setPath

void **setPath**(java.lang.String path)

> property setPath
> **Throws:**

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPFileFragment

All Superinterfaces:
> IRPModelElement

---

public interface **IRPFileFragment**
extends IRPModelElement

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPModelElement	**getFragmentElement**() get property fragmentElement
java.lang.String	**getFragmentText**() get property fragmentText
java.lang.String	**getFragmentType**() get property fragmentType
void	**moveFragmentInOwner**(int up) method moveFragmentInOwner
void	**setFragmentText**(java.lang.String fragmentText) set property fragmentText

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn,

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getFragmentElement

IRPModelElement **getFragmentElement**()

> get property fragmentElement
> **Throws:**
> > RhapsodyRuntimeException

getFragmentText

java.lang.String **getFragmentText**()

> get property fragmentText
> **Throws:**
> > RhapsodyRuntimeException

getFragmentType

java.lang.String **getFragmentType**()

> get property fragmentType
> **Throws:**
> > RhapsodyRuntimeException

moveFragmentInOwner

void **moveFragmentInOwner**(int up)

> method moveFragmentInOwner
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

setFragmentText

void **setFragmentText**(java.lang.String fragmentText)

> set property fragmentText
> **Throws:**
> > [RhapsodyRuntimeException](#)

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

PREV CLASS   **NEXT CLASS**   **FRAMES**   **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

PREV CLASS   **NEXT CLASS**   **FRAMES**   **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPFlow

All Superinterfaces:
> IRPModelElement

---

public interface **IRPFlow**
extends IRPModelElement

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

void	**addConveyed**(IRPModelElement pElement) method addConveyed
IRPCollection	**getConveyed**() get property conveyed
java.lang.String	**getDirection**() get property direction
IRPModelElement	**getEnd1**() get property end1
IRPPort	**getEnd1Port**() get property end1Port
IRPSysMLPort	**getEnd1SysMLPort**() get property end1SysMLPort
IRPModelElement	**getEnd2**() get property end2
IRPPort	**getEnd2Port**() get property end2Port
IRPSysMLPort	**getEnd2SysMLPort**() get property end2SysMLPort

Method Summary

void	**removeConveyed**([IRPModelElement](#) pElement) method removeConveyed
void	**setDirection**(java.lang.String direction) Specifies the direction to use for the flow.
void	**setEnd1**([IRPModelElement](#) end1) set property end1
void	**setEnd1ViaPort**([IRPInstance](#) pInstance, [IRPPort](#) pPort) method setEnd1ViaPort
void	**setEnd1ViaSysMLPort**([IRPInstance](#) pInstance, [IRPSysMLPort](#) pSysMLPort) method setEnd1ViaSysMLPort
void	**setEnd2**([IRPModelElement](#) end2) set property end2
void	**setEnd2ViaPort**([IRPInstance](#) pInstance, [IRPPort](#) pPort) method setEnd2ViaPort
void	**setEnd2ViaSysMLPort**([IRPInstance](#) pInstance, [IRPSysMLPort](#) pSysMLPort) method setEnd2ViaSysMLPort

Methods inherited from interface com.telelogic.rhapsody.core.**[IRPModelElement](#)**

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addConveyed

void **addConveyed**(IRPModelElement pElement)

> method addConveyed
> **Throws:**
> > RhapsodyRuntimeException

---

getConveyed

IRPCollection **getConveyed**()

> get property conveyed
> **Throws:**
> > RhapsodyRuntimeException

---

getDirection

java.lang.String **getDirection**()

> get property direction
> **Throws:**
> > RhapsodyRuntimeException

---

getEnd1

IRPModelElement **getEnd1**()

> get property end1
> **Throws:**
> > RhapsodyRuntimeException

---

getEnd1Port

IRPPort **getEnd1Port**()

> get property end1Port
> **Throws:**
> > RhapsodyRuntimeException

---

getEnd1SysMLPort

IRPSysMLPort **getEnd1SysMLPort**()

> get property end1SysMLPort
> **Throws:**
> > RhapsodyRuntimeException

getEnd2

IRPModelElement **getEnd2**()

> get property end2
> **Throws:**
> > RhapsodyRuntimeException

getEnd2Port

IRPPort **getEnd2Port**()

> get property end2Port
> **Throws:**
> > RhapsodyRuntimeException

getEnd2SysMLPort

IRPSysMLPort **getEnd2SysMLPort**()

> get property end2SysMLPort
> **Throws:**
> > RhapsodyRuntimeException

removeConveyed

void **removeConveyed**(IRPModelElement pElement)

> method removeConveyed
> **Throws:**
> > RhapsodyRuntimeException

setDirection

void **setDirection**(java.lang.String direction)

> Specifies the direction to use for the flow.
> **Parameters:**

`direction` - can be one of the following values: "toEnd1", "toEnd2", "bidirectional"
Throws:
> RhapsodyRuntimeException

setEnd1

void **setEnd1**(IRPModelElement end1)

> set property end1
> **Throws:**
> > RhapsodyRuntimeException

setEnd1ViaPort

void **setEnd1ViaPort**(IRPInstance pInstance,
                        IRPPort pPort)

> method setEnd1ViaPort
> **Throws:**
> > RhapsodyRuntimeException

setEnd1ViaSysMLPort

void **setEnd1ViaSysMLPort**(IRPInstance pInstance,
                             IRPSysMLPort pSysMLPort)

> method setEnd1ViaSysMLPort
> **Throws:**
> > RhapsodyRuntimeException

setEnd2

void **setEnd2**(IRPModelElement end2)

> set property end2
> **Throws:**
> > RhapsodyRuntimeException

setEnd2ViaPort

void **setEnd2ViaPort**(IRPInstance pInstance,
                        IRPPort pPort)

> method setEnd2ViaPort
> **Throws:**
> > RhapsodyRuntimeException

setEnd2ViaSysMLPort

```
void setEnd2ViaSysMLPort(IRPInstance pInstance,
                         IRPSysMLPort pSysMLPort)
```

> method setEnd2ViaSysMLPort
> **Throws:**
> > RhapsodyRuntimeException

com.telelogic.rhapsody.core
Interface IRPFlowchart

All Superinterfaces:
>   IRPClass, IRPClassifier, IRPModelElement, IRPStatechart, IRPUnit

---

public interface **IRPFlowchart**
extends IRPStatechart

The IRPFlowchart interface represents activities in Rational Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPAcceptEventAction	**addAcceptEventAction**(java.lang.String name, IRPState parent) Adds a new Accept Event Action element to the activity.
IRPAcceptTimeEvent	**addAcceptTimeEvent**(java.lang.String name, IRPState parent) Adds a new Accept Time Event element to the activity.
IRPPin	**addActivityParameter**(java.lang.String name) Adds an activity parameter to the frame of the activity
IRPState	**addCallBehavior**(IRPModelElement referenced) Adds a new Call Behavior element to the activity.
IRPCallOperation	**addCallOperation**(java.lang.String name, IRPState parent) Adds a new Call Operation element to the activity.
IRPObjectNode	**addObjectNode**(java.lang.String name, IRPState parent) Adds a new Object Node element to the activity.
IRPState	**addReferenceActivity**(IRPModelElement referenced) Adds a new Call Behavior element to the activity.
IRPSwimlane	**addSwimlane**(java.lang.String name) Adds a new swimlane to the activity.
IRPActivityDiagram	

Method Summary

	getFlowchartDiagram() Returns the IRPActivityDiagram object associated with the activity.
int	**getIsAnalysisOnly**() Checks whether the activity is defined as analysis-only, meaning that it is used only for modeling purposes and code is not generated for the activity.
IRPOperation	**getItsOwner**() **Deprecated.** *Use IRPModelElement.getOwner instead.*
IRPCollection	**getSwimlanes**() Returns a collection of all the swimlanes in the activity.
void	**setIsAnalysisOnly**(int isAnalysisOnly) Specifies whether the activity should be defined as analysis-only.
void	**setItsOwner**(IRPOperation itsOwner) **Deprecated.** *Use IRPModelElement.setOwner instead.*

Methods inherited from interface com.telelogic.rhapsody.core.**IRPStatechart**

addFreeShapeByType, addImage, addNewEdgeByType, addNewEdgeForElement, addNewNodeByType, addNewNodeForElement, addTextBox, closeDiagram, createGraphics, deleteState, findTrigger, getAllTriggers, getElementsInDiagram, getGraphicalElements, getInheritsFrom, getIsMainBehavior, getIsOverridden, getItsClass, getPicture, getPictureAs, getPictureAsDividedMetafiles, getPicturesWithImageMap, getRootState, getStatechartDiagram, openDiagramView, overrideInheritance, populateDiagram, setAsMainBehavior, setShowDiagramFrame, unoverrideInheritance

Methods inherited from interface com.telelogic.rhapsody.core.**IRPClass**

addClass, addConstructor, addDestructor, addEventReception, addLink, addLinkToPartViaPort, addReception, addSuperclass, addTriggeredOperation, addType, deleteClass, deleteConstructor, deleteDestructor, deleteEventReception, deleteReception, deleteSuperclass, deleteType, getIsAbstract, getIsActive, getIsBehaviorOverriden, getIsComposite, getIsFinal, getIsReactive, setIsAbstract, setIsActive, setIsBehaviorOverriden, setIsFinal

Methods inherited from interface com.telelogic.rhapsody.core.**IRPClassifier**

addActivityDiagram, addAttribute, addFlowItems, addFlows, addGeneralization, addOperation, addRelation, addRelationTo, addStatechart, addUnidirectionalRelation, addUnidirectionalRelationTo, deleteAttribute, deleteFlowItems, deleteFlows, deleteGeneralization, deleteOperation, deleteRelation, findAttribute, findBaseClassifier, findDerivedClassifier, findGeneralization, findInterfaceItem, findNestedClassifier, findNestedClassifierRecursive, findRelation, findTrigger, getActivityDiagram, getAttributes, getAttributesIncludingBases, getBaseClassifiers, getBehavioralDiagrams, getDerivedClassifiers, getFlowItems, getFlows, getGeneralizations, getInterfaceItems, getInterfaceItemsIncludingBases, getLinks, getNestedClassifiers, getOperations, getPorts, getRelations, getRelationsIncludingBases, getSequenceDiagrams, getSourceArtifacts, getStatechart

Methods inherited from interface com.telelogic.rhapsody.core.**IRPUnit**

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addAcceptEventAction

IRPAcceptEventAction **addAcceptEventAction**(java.lang.String name,
                                   IRPState parent)

Adds a new Accept Event Action element to the activity.
Parameters:
   name - the name to use for the new Accept Event Action element
   parent - the diagram element to which the new Accept Event Action element should be added. If the Accept Event Action element is being added to an Action Block, this parameter should be the Action Block. Otherwise, it should be the root state of the diagram (which is

obtained by calling IRPStatechart.getRootState()).
Returns:
the Accept Event Action element that was created

---

addAcceptTimeEvent

IRPAcceptTimeEvent **addAcceptTimeEvent**(java.lang.String name,
IRPState parent)

Adds a new Accept Time Event element to the activity.
Parameters:
name - the name to use for the new Accept Time Event element
parent - the diagram element to which the new Accept Time Event element should be added.
If the Accept Time Event element is being added to an Action Block, this parameter should
be the Action Block. Otherwise, it should be the root state of the diagram (which is obtained
by calling IRPStatechart.getRootState()).
Returns:
the Accept Time Event element that was created

---

addActivityParameter

IRPPin **addActivityParameter**(java.lang.String name)

Adds an activity parameter to the frame of the activity
Parameters:
name - the name to use for the new activity parameter
Returns:
the activity parameter element that was created

---

addCallBehavior

IRPState **addCallBehavior**(IRPModelElement referenced)

Adds a new Call Behavior element to the activity.
Parameters:
referenced - the activity that the new Call Behavior element should invoke
Returns:
the Call Behavior element that was created

---

addCallOperation

IRPCallOperation **addCallOperation**(java.lang.String name,
IRPState parent)

Adds a new Call Operation element to the activity.
Parameters:

---

name - the name to use for the new Call Operation element

parent - the diagram element to which the new Call Operation element should be added. If the Call Operation element is being added to an Action Block, this parameter should be the Action Block. Otherwise, it should be the root state of the diagram (which is obtained by calling IRPStatechart.getRootState()).

Returns:

the Call Operation element that was created

addObjectNode

IRPObjectNode **addObjectNode**(java.lang.String name,
                               IRPState parent)

Adds a new Object Node element to the activity.

Parameters:

name - the name to use for the new Object Node element

parent - the diagram element to which the new Object Node element should be added. If the Object Node element is being added to an Action Block, this parameter should be the Action Block. Otherwise, it should be the root state of the diagram (which is obtained by calling IRPStatechart.getRootState()).

Returns:

the Object Node element that was created

addReferenceActivity

IRPState **addReferenceActivity**(IRPModelElement referenced)

Adds a new Call Behavior element to the activity. Performs same action as the addCallBehavior method.

Parameters:

referenced - the activity that the new Call Behavior element should invoke

Returns:

the Call Behavior element that was created

addSwimlane

IRPSwimlane **addSwimlane**(java.lang.String name)

Adds a new swimlane to the activity.

Parameters:

name - the name to use for the new swimlane

Returns:

the swimlane that was created

getFlowchartDiagram

IRPActivityDiagram **getFlowchartDiagram**()

> Returns the IRPActivityDiagram object associated with the activity.
> **Returns:**
> > the IRPActivityDiagram object associated with the activity

getIsAnalysisOnly

int **getIsAnalysisOnly**()

> Checks whether the activity is defined as analysis-only, meaning that it is used only for modeling
> purposes and code is not generated for the activity.
> **Returns:**
> > 1 if the activity is defined as analysis-only, 0 otherwise

getItsOwner

IRPOperation **getItsOwner**()

> **Deprecated.** *Use IRPModelElement.getOwner instead.*

getSwimlanes

IRPCollection **getSwimlanes**()

> Returns a collection of all the swimlanes in the activity.
> **Returns:**
> > collection of IRPSwimlane objects

setIsAnalysisOnly

void **setIsAnalysisOnly**(int isAnalysisOnly)

> Specifies whether the activity should be defined as analysis-only.
> **Parameters:**
> > isAnalysisOnly - Use 1 to specify that the activity should be defined as analysis-only. Use 0
> > to specify that the activity should not be defined as analysis-only.

setItsOwner

void **setItsOwner**(IRPOperation itsOwner)

> **Deprecated.** *Use IRPModelElement.setOwner instead.*

com.telelogic.rhapsody.core
Interface IRPFlowItem

All Superinterfaces:
> IRPClassifier, IRPModelElement, IRPUnit

---

public interface **IRPFlowItem**
extends IRPClassifier

The IRPFlowItem interface represents item flows in Rational Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

void	**addRepresented**(IRPModelElement pElement) Adds an element to the collection of information elements that are represented by the item flow.
IRPCollection	**getRepresented**() Returns a collection of all the information elements that are represented by the item flow.
void	**removeRepresented**(IRPModelElement pElement) Removes the specified element from the collection of information elements that are represented by the item flow.

Methods inherited from interface com.telelogic.rhapsody.core.IRPClassifier
addActivityDiagram, addAttribute, addFlowItems, addFlows, addGeneralization, addOperation, addRelation, addRelationTo, addStatechart, addUnidirectionalRelation, addUnidirectionalRelationTo, deleteAttribute, deleteFlowItems, deleteFlows, deleteGeneralization, deleteOperation, deleteRelation, findAttribute, findBaseClassifier, findDerivedClassifier, findGeneralization, findInterfaceItem, findNestedClassifier, findNestedClassifierRecursive, findRelation, findTrigger, getActivityDiagram, getAttributes, getAttributesIncludingBases, getBaseClassifiers, getBehavioralDiagrams, getDerivedClassifiers, getFlowItems, getFlows, getGeneralizations, getInterfaceItems, getInterfaceItemsIncludingBases, getLinks, getNestedClassifiers, getOperations, getPorts, getRelations,

Methods inherited from interface com.telelogic.rhapsody.core.IRPClassifier
getRelationsIncludingBases, getSequenceDiagrams, getSourceArtifacts, getStatechart

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit
copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addRepresented

```
void addRepresented(IRPModelElement pElement)
```

> Adds an element to the collection of information elements that are represented by the item flow.
> **Parameters:**
> > pElement - the element to add to the collection

getRepresented

<u>IRPCollection</u> **getRepresented**()

Returns a collection of all the information elements that are represented by the item flow.
Returns:
all the information elements that are represented by the item flow

removeRepresented

void **removeRepresented**(<u>IRPModelElement</u> pElement)

Removes the specified element from the collection of information elements that are represented by the item flow.
Parameters:
pElement - the element that should be removed from the collection

---

---

com.telelogic.rhapsody.core
Interface IRPGeneralization

All Superinterfaces:
>   IRPModelElement

---

public interface **IRPGeneralization**
extends IRPModelElement

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPClassifier	**getBaseClass**() get method baseClass
IRPClassifier	**getDerivedClass**() get method derivedClass
java.lang.String	**getExtensionPoint**() get property extensionPoint
int	**getIsVirtual**() get property is virtual
java.lang.String	**getVisibility**() get property visibility
void	**setBaseClass**(IRPClassifier baseClass) set method baseClass
void	**setDerivedClass**(IRPClassifier derivedClass) set method derivedClass
void	**setExtensionPoint**(java.lang.String extensionPoint) set property extensionPoint
void	**setIsVirtual**(int isVirtual) set property is virtual

Method Summary

void	**setVisibility**(java.lang.String visibility) set property visibility

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getBaseClass

IRPClassifier **getBaseClass**()

> get method baseClass
> **Throws:**
> > RhapsodyRuntimeException

getDerivedClass

IRPClassifier **getDerivedClass**()

> get method derivedClass

com.telelogic.rhapsody.coreInterface IRPGeneralization   266

Throws:
[RhapsodyRuntimeException](#)

---

getExtensionPoint

`java.lang.String` **`getExtensionPoint`**`()`

get property extensionPoint
Throws:
[RhapsodyRuntimeException](#)

---

getIsVirtual

`int` **`getIsVirtual`**`()`

get property is virtual
Throws:
[RhapsodyRuntimeException](#)

---

getVisibility

`java.lang.String` **`getVisibility`**`()`

get property visibility
Throws:
[RhapsodyRuntimeException](#)

---

setBaseClass

`void` **`setBaseClass`**`(`[IRPClassifier](#) `baseClass)`

set method baseClass
Throws:
[RhapsodyRuntimeException](#)

---

setDerivedClass

`void` **`setDerivedClass`**`(`[IRPClassifier](#) `derivedClass)`

set method derivedClass
Throws:
[RhapsodyRuntimeException](#)

---

setExtensionPoint

void **setExtensionPoint**(java.lang.String extensionPoint)

> set property extensionPoint
> **Throws:**
> > RhapsodyRuntimeException

setIsVirtual

void **setIsVirtual**(int isVirtual)

> set property is virtual
> **Throws:**
> > RhapsodyRuntimeException

setVisibility

void **setVisibility**(java.lang.String visibility)

> set property visibility
> **Throws:**
> > RhapsodyRuntimeException

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

com.telelogic.rhapsody.core
Interface IRPGraphEdge

All Superinterfaces:
> IRPGraphElement

---

public interface **IRPGraphEdge**
extends IRPGraphElement

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPGraphElement
IRPGraphElement.ImageLayout

Method Summary

IRPGraphEdge	**embedFlow**(IRPFlow flow) method embedFlow
IRPGraphEdge	**embedNewFlow**() method embedNewFlow
IRPGraphEdge	**getContainingArrow**() method getContainingArrow
IRPGraphElement	**getSource**() get property source
IRPGraphElement	**getTarget**() get property target

Methods inherited from interface com.telelogic.rhapsody.core.IRPGraphElement
addProperty, applyDefaultFormat, getAllGraphicalProperties, getAllProperties, getAssociatedImage, getDiagram, getGraphicalParent, getGraphicalProperty, getGraphicalPropertyOfText, getImageLayout, getInterfaceName, getLocalProperties, getModelObject, getPropertyValue, getSelectedImage, removeProperty, setAssociatedImage, setGraphicalProperty, setGraphicalPropertyOfText, setImageLayout, setPropertyValue, setSelectedImage

Method Detail

embedFlow

IRPGraphEdge **embedFlow**(IRPFlow flow)

> method embedFlow
> **Throws:**
> > RhapsodyRuntimeException

---

embedNewFlow

IRPGraphEdge **embedNewFlow**()

> method embedNewFlow
> **Throws:**
> > RhapsodyRuntimeException

---

getContainingArrow

IRPGraphEdge **getContainingArrow**()

> method getContainingArrow
> **Throws:**
> > RhapsodyRuntimeException

---

getSource

IRPGraphElement **getSource**()

> get property source
> **Throws:**
> > RhapsodyRuntimeException

---

getTarget

IRPGraphElement **getTarget**()

> get property target
> **Throws:**
> > RhapsodyRuntimeException

Package   Class   [Use](Use)   Tree   Serialized   Deprecated   Index   Help
PREV CLASS   NEXT CLASS   FRAMES   NO FRAMES   All Classes

com.telelogic.rhapsody.coreInterface IRPGraphEdge   270

com.telelogic.rhapsody.core
Interface IRPGraphElement

All Known Subinterfaces:
> IRPGraphEdge, IRPGraphNode

```
public interface IRPGraphElement
```

Nested Class Summary

static class	**IRPGraphElement.ImageLayout** This class contains constant values for use with the method setImageLayout

Method Summary

void	**addProperty**(java.lang.String propertyKey, java.lang.String propertyType, java.lang.String propertyValue) method addProperty
void	**applyDefaultFormat**() method applyDefaultFormat
IRPCollection	**getAllGraphicalProperties**() method getAllGraphicalProperties
IRPCollection	**getAllProperties**() method getAllProperties
java.lang.String	**getAssociatedImage**() get associatedImage
IRPDiagram	**getDiagram**() method getDiagram
IRPGraphElement	**getGraphicalParent**() get property graphicalParent
IRPGraphicalProperty	**getGraphicalProperty**(java.lang.String name) method getGraphicalProperty
IRPGraphicalProperty	**getGraphicalPropertyOfText**(java.lang.String textName, java.lang.String name) Returns the specified graphical property for a textual element associated with the graphic element.

Method Summary

java.lang.String	**getImageLayout**() Returns the image layout specified for the image linked to the graphic element.
java.lang.String	**getInterfaceName**() get property interfaceName
IRPCollection	**getLocalProperties**() method getLocalProperties
IRPModelElement	**getModelObject**() get property modelObject
java.lang.String	**getPropertyValue**(java.lang.String propertyKey) method getPropertyValue
java.lang.String	**getSelectedImage**() Returns the full path of the image that was linked to the graphic element.
void	**removeProperty**(java.lang.String propertyKey) method removeProperty
void	**setAssociatedImage**(java.lang.String associatedImage) set associatedImage
void	**setGraphicalProperty**(java.lang.String name, java.lang.String value) Sets a new value for a graphical property.
void	**setGraphicalPropertyOfText**(java.lang.String textName, java.lang.String name, java.lang.String value) Sets a new value for a graphical property for the specified textual element associated with the graphic element.
void	**setImageLayout**(java.lang.String imageLayout) Used to specify the image layout that should be used for the image linked to the graphic element.
void	**setPropertyValue**(java.lang.String propertyKey, java.lang.String propertyValue) method setPropertyValue
void	**setSelectedImage**(java.lang.String selectedImage) Links the graphic element to the image represented by the path specified.

Method Detail

addProperty

```
void addProperty(java.lang.String propertyKey,
                 java.lang.String propertyType,
                 java.lang.String propertyValue)
```

method addProperty
Throws:
> RhapsodyRuntimeException

---

applyDefaultFormat

void **applyDefaultFormat**()

> method applyDefaultFormat
> **Throws:**
> > RhapsodyRuntimeException

---

getAllGraphicalProperties

IRPCollection **getAllGraphicalProperties**()

> method getAllGraphicalProperties
> **Throws:**
> > RhapsodyRuntimeException

---

getAllProperties

IRPCollection **getAllProperties**()

> method getAllProperties
> **Throws:**
> > RhapsodyRuntimeException

---

getAssociatedImage

java.lang.String **getAssociatedImage**()

> get associatedImage
> **Throws:**
> > RhapsodyRuntimeException

---

getDiagram

IRPDiagram **getDiagram**()

> method getDiagram
> **Throws:**
> > RhapsodyRuntimeException

---

getGraphicalParent

IRPGraphElement **getGraphicalParent**()

get property graphicalParent
Throws:
     RhapsodyRuntimeException

---

getGraphicalProperty

IRPGraphicalProperty **getGraphicalProperty**(java.lang.String name)

method getGraphicalProperty
Throws:
     RhapsodyRuntimeException

---

getGraphicalPropertyOfText

IRPGraphicalProperty **getGraphicalPropertyOfText**(java.lang.String textName,
                                                  java.lang.String name)

Returns the specified graphical property for a textual element associated with the graphic element. This method is intended for use with graphic elements that have more than one textual element associated with them. The textName parameter is used to indicate which of the textual elements you want the property for. The values that can be used for the textName parameter depend upon the type of graphic element, as follows:

◊ For all graphic elements - "Name", "Stereotype"
◊ For lines only - "Label"
◊ For flows only - "Keyword", "Conveyed"
◊ For parts and objects only - "Multiplicity"
◊ For association ends and links only - "SourceRole", "TargetRole", "SourceMultiplicity", "TargetMultiplicity"
◊ For association ends only - "SourceQualifier", "TargetQualifier"
◊ For ports only - "ProvidedInterfaceLabel", "RequiredInterfaceLabel"

For graphic elements associated with no more than one textual element, use the method getGraphicalProperty(java.lang.String).

Parameters:
     textName - the specific textual element that you want the property for
     name - the name of the graphical property, for example, "TextFontName", "TextColor", "TextFontItalic", "TextFontSize", "TextFontBold"
Returns:
     the graphical property that was requested

---

getImageLayout

java.lang.String **getImageLayout**()

Returns the image layout specified for the image linked to the graphic element. When using the Java version of the API, the value returned will be one of the constants defined in the class <u>IRPGraphElement.ImageLayout</u>. When using the COM version of the API, the value returned will be one of the following strings: "Image Only Show Name", "Image Only Without Name", "Structured", "Compartment".

Returns:
> the image layout specified for the image linked to the graphic element

---

getInterfaceName

java.lang.String **getInterfaceName**()

> get property interfaceName
> **Throws:**
> > <u>RhapsodyRuntimeException</u>

---

getLocalProperties

<u>IRPCollection</u> **getLocalProperties**()

> method getLocalProperties
> **Throws:**
> > <u>RhapsodyRuntimeException</u>

---

getModelObject

<u>IRPModelElement</u> **getModelObject**()

> get property modelObject
> **Throws:**
> > <u>RhapsodyRuntimeException</u>

---

getPropertyValue

java.lang.String **getPropertyValue**(java.lang.String propertyKey)

> method getPropertyValue
> **Throws:**
> > <u>RhapsodyRuntimeException</u>

---

getSelectedImage

java.lang.String **getSelectedImage**()

> Returns the full path of the image that was linked to the graphic element.

getImageLayout

Returns:
> the full path of the image linked to the graphic element

---

removeProperty

void **removeProperty**(java.lang.String propertyKey)

> method removeProperty
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

setAssociatedImage

void **setAssociatedImage**(java.lang.String associatedImage)

> set associatedImage
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

setGraphicalProperty

void **setGraphicalProperty**(java.lang.String name,
> > > > > java.lang.String value)

> Sets a new value for a graphical property. Certain graphical properties are available only for specific types of elements. Therefore, before including calls to this method in your code, you should call IRPGraphElement.getAllGraphicalProperties, which returns a collection of IRPGraphicalProperty objects representing the graphical properties available for the element in question. Note that for the name parameter, you only have to provide the property name, not the entire hierarchy as is the case with the method setPropertyValue.
> **Parameters:**
> > name - the name of the graphical property to set
> > value - the value to use for the specified property

---

setGraphicalPropertyOfText

void **setGraphicalPropertyOfText**(java.lang.String textName,
> > > > > java.lang.String name,
> > > > > java.lang.String value)

> Sets a new value for a graphical property for the specified textual element associated with the graphic element. This method is intended for use with graphic elements that have more than one textual element associated with them. The textName parameter is used to indicate which of the textual elements you want to set the property for. The values that can be used for the textName parameter depend upon the type of graphic element, as follows:

> > ◊ For all graphic elements - "Name", "Stereotype"

◊ For lines only - "Label"
◊ For flows only - "Keyword", "Conveyed"
◊ For parts and objects only - "Multiplicity"
◊ For association ends and links only - "SourceRole", "TargetRole", "SourceMultiplicity", "TargetMultiplicity"
◊ For association ends only - "SourceQualifier", "TargetQualifier"
◊ For ports only - "ProvidedInterfaceLabel", "RequiredInterfaceLabel"

Certain graphical properties are available only for specific types of elements. Therefore, before including calls to this method in your code, you should call IRPGraphElement.getAllGraphicalProperties, which returns a collection of IRPGraphicalProperty objects representing the graphical properties available for the element in question. Note that for the name parameter, you only have to provide the property name, not the entire hierarchy as is the case with the method setPropertyValue. For graphic elements associated with no more than one textual element, use the method setGraphicalProperty(java.lang.String, java.lang.String).

Parameters:
> textName - the specific textual element that you want to set the property for
> name - the name of the graphical property to set
> value - the value to use for the specified property

setImageLayout

void **setImageLayout**(java.lang.String imageLayout)

Used to specify the image layout that should be used for the image linked to the graphic element. When using the Java version of the API, the value of the parameter should be one of the constants defined in the class IRPGraphElement.ImageLayout. When using the COM version of the API, the value of the parameter should be one of the following strings: "Image Only Show Name", "Image Only Without Name", "Structured", "Compartment".

Parameters:
> imageLayout - the image layout that should be used for the image linked to the graphic element

setPropertyValue

void **setPropertyValue**(java.lang.String propertyKey,
                        java.lang.String propertyValue)

method setPropertyValue
Throws:
> RhapsodyRuntimeException

setSelectedImage

void **setSelectedImage**(java.lang.String selectedImage)

Links the graphic element to the image represented by the path specified. To remove an existing link to an image without providing a new image, use an empty string for the parameter.

Parameters:

      `selectedImage` - the full path to the image that should be linked to the graphic element

---

---

com.telelogic.rhapsody.core
Class IRPGraphElement.ImageLayout

```
java.lang.Object
   └ com.telelogic.rhapsody.core.IRPGraphElement.ImageLayout
```

Enclosing interface:
>   IRPGraphElement

---

public static final class **IRPGraphElement.ImageLayout**
extends java.lang.Object

This class contains constant values for use with the method setImageLayout

---

Field Summary

static java.lang.String	**COMPARTMENT** Set image layout in a compartment
static java.lang.String	**IMAGE_ONLY_SHOW_NAME** Set image layout as show image only with name
static java.lang.String	**IMAGE_ONLY_WITHOUT_NAME** Set image layout as show image only without name
static java.lang.String	**STRUCTURED** Show image in structured layout

Constructor Summary

IRPGraphElement.ImageLayout()	

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

STRUCTURED

public static final java.lang.String **STRUCTURED**

> Show image in structured layout
>
> **See Also:**
> > [Constant Field Values](#)

IMAGE_ONLY_WITHOUT_NAME

public static final java.lang.String **IMAGE_ONLY_WITHOUT_NAME**

> Set image layout as show image only without name
>
> **See Also:**
> > [Constant Field Values](#)

IMAGE_ONLY_SHOW_NAME

public static final java.lang.String **IMAGE_ONLY_SHOW_NAME**

> Set image layout as show image only with name
>
> **See Also:**
> > [Constant Field Values](#)

COMPARTMENT

public static final java.lang.String **COMPARTMENT**

> Set image layout in a compartment
>
> **See Also:**
> > [Constant Field Values](#)

Constructor Detail

IRPGraphElement.ImageLayout

public **IRPGraphElement.ImageLayout**()

---

com.telelogic.rhapsody.core
Interface IRPGraphicalProperty

public interface **IRPGraphicalProperty**

Method Summary

java.lang.String	**getInterfaceName**() get property interfaceName
java.lang.String	**getKey**() get property key
java.lang.String	**getValue**() get property value

Method Detail

getInterfaceName

java.lang.String **getInterfaceName**()

>   get property interfaceName
>   **Throws:**
>   RhapsodyRuntimeException

---

getKey

java.lang.String **getKey**()

>   get property key
>   **Throws:**
>   RhapsodyRuntimeException

---

getValue

`java.lang.String` **`getValue`**`()`

> get property value
> **Throws:**
>> RhapsodyRuntimeException

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

---

com.telelogic.rhapsody.core
Interface IRPGraphNode

All Superinterfaces:
> IRPGraphElement

---

public interface **IRPGraphNode**
extends IRPGraphElement

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPGraphElement
IRPGraphElement.ImageLayout

Method Summary

void	**bringToFront**() method bringToFront
int	**getIsPanelWidget**() get property isPanelWidget
IRPCollection	**getPanelWidgetInstancePath**() get property panelWidgetInstancePath
void	**hideAllPorts**() method hideAllPorts
void	**sendToBack**() method sendToBack
void	**setPanelWidgetInstancePath**(IRPCollection panelWidgetInstancePath) set property panelWidgetInstancePath
void	**showAllPorts**() method showAllPorts

Methods inherited from interface com.telelogic.rhapsody.core.IRPGraphElement
addProperty, applyDefaultFormat, getAllGraphicalProperties, getAllProperties, getAssociatedImage, getDiagram, getGraphicalParent, getGraphicalProperty, getGraphicalPropertyOfText, getImageLayout, getInterfaceName, getLocalProperties,

Methods inherited from interface com.telelogic.rhapsody.core.IRPGraphElement

getModelObject, getPropertyValue, getSelectedImage, removeProperty, setAssociatedImage, setGraphicalProperty, setGraphicalPropertyOfText, setImageLayout, setPropertyValue, setSelectedImage

Method Detail

bringToFront

void **bringToFront**()

> method bringToFront
> **Throws:**
> > RhapsodyRuntimeException

---

getIsPanelWidget

int **getIsPanelWidget**()

> get property isPanelWidget
> **Throws:**
> > RhapsodyRuntimeException

---

getPanelWidgetInstancePath

IRPCollection **getPanelWidgetInstancePath**()

> get property panelWidgetInstancePath
> **Throws:**
> > RhapsodyRuntimeException

---

hideAllPorts

void **hideAllPorts**()

> method hideAllPorts
> **Throws:**
> > RhapsodyRuntimeException

---

sendToBack

void **sendToBack**()

method sendToBack
Throws:
>
> RhapsodyRuntimeException

setPanelWidgetInstancePath

void **setPanelWidgetInstancePath**(IRPCollection panelWidgetInstancePath)

set property panelWidgetInstancePath
Throws:
>
> RhapsodyRuntimeException

showAllPorts

void **showAllPorts**()

method showAllPorts
Throws:
>
> RhapsodyRuntimeException

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | **METHOD**   DETAIL: FIELD | CONSTR | **METHOD**

com.telelogic.rhapsody.core
Interface IRPGuard

All Superinterfaces:
> IRPModelElement

---

public interface **IRPGuard**
extends IRPModelElement

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

java.lang.String	**getBody**() get property body
void	**setBody**(java.lang.String body) set property body

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML,

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement,
isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser,
lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines,
removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks,
setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID,
setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner,
setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue,
setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation,
unlockOnDesignManager

Method Detail

getBody

java.lang.String **getBody**()

get property body
Throws:
RhapsodyRuntimeException

setBody

void **setBody**(java.lang.String body)

set property body
Throws:
RhapsodyRuntimeException

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**
PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.coreInterface IRPGuard   289

com.telelogic.rhapsody.core
Interface IRPHyperLink

All Superinterfaces:
> IRPDependency, IRPModelElement

---

public interface **IRPHyperLink**
extends IRPDependency

The IRPHyperLink interface represents hyperlinks in Rational Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

void	**getDisplayOption**(char pVal, java.lang.String pDisplayName) getDisplayOption
IRPModelElement	**getTarget**() Returns the target model element if the hyperlink points to a model element.
java.lang.String	**getURL**() Returns the target URL if the hyperlink points to a URL.
void	**setDisplayOption**(char newVal, java.lang.String newDisplayName) setDisplayOption
void	**setTarget**(IRPModelElement target) Sets the specified model element to be the target of the hyperlink.
void	**setURL**(java.lang.String uRL) Sets the specified URL to be the target of the hyperlink.

Methods inherited from interface com.telelogic.rhapsody.core.IRPDependency
getDependent, getDependsOn, setDependent, setDependsOn, setOwnerWithoutChangingDependent

com.telelogic.rhapsody.core

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getDisplayOption

```
void getDisplayOption(char pVal,
                      java.lang.String pDisplayName)
```

getDisplayOption
Throws:
RhapsodyRuntimeException

getTarget

IRPModelElement **getTarget**()

Returns the target model element if the hyperlink points to a model element.
Returns:
the model element that the hyperlink points to

getURL

java.lang.String **getURL**()

> Returns the target URL if the hyperlink points to a URL.
> **Returns:**
> > the URL that the hyperlink points to

---

setDisplayOption

void **setDisplayOption**(char newVal,
                        java.lang.String newDisplayName)

> setDisplayOption
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

setTarget

void **setTarget**([IRPModelElement](#) target)

> Sets the specified model element to be the target of the hyperlink.
> **Parameters:**
> > target - the model element that should be used as the target of the hyperlink

---

setURL

void **setURL**(java.lang.String uRL)

> Sets the specified URL to be the target of the hyperlink.
> **Parameters:**
> > uRL - the URL that should be used as the target of the hyperlink

---

com.telelogic.rhapsody.core
Interface IRPImageMap

```
public interface IRPImageMap
```

Method Summary

java.lang.String	**getInterfaceName**() get property interfaceName
int	**getIsGUID**() get property isGUID
java.lang.String	**getName**() get property name
java.lang.String	**getPictureFileName**() get property pictureFileName
java.lang.String	**getPoints**() get property points
java.lang.String	**getShape**() get property shape
java.lang.String	**getTarget**() get property target

Method Detail

getInterfaceName

```
java.lang.String getInterfaceName()
```

> get property interfaceName
> **Throws:**
> > RhapsodyRuntimeException

getIsGUID

int **getIsGUID**()

> get property isGUID
> **Throws:**
> > RhapsodyRuntimeException

getName

java.lang.String **getName**()

> get property name
> **Throws:**
> > RhapsodyRuntimeException

getPictureFileName

java.lang.String **getPictureFileName**()

> get property pictureFileName
> **Throws:**
> > RhapsodyRuntimeException

getPoints

java.lang.String **getPoints**()

> get property points
> **Throws:**
> > RhapsodyRuntimeException

getShape

java.lang.String **getShape**()

> get property shape
> **Throws:**
> > RhapsodyRuntimeException

getTarget

java.lang.String **getTarget**()

> get property target
> **Throws:**

com.telelogic.rhapsody.core
Interface IRPInstance

All Superinterfaces:
        IRPModelElement, IRPRelation, IRPUnit

All Known Subinterfaces:
        IRPModule, IRPPort, IRPSysMLPort

```
public interface IRPInstance
```
extends IRPRelation

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPRelation	**addRelationToTheWhole**(java.lang.String relName) method addRelationToTheWhole
java.lang.String	**getAttributeValue**(java.lang.String attName) method getAttributeValue
IRPCollection	**getInLinks**() method getInLinks
IRPOperation	**getInstantiatedBy**() get property instantiatedBy
IRPCollection	**getListOfInitializerArguments**() method getListOfInitializerArguments
IRPCollection	**getOutLinks**() method getOutLinks
void	**setAttributeValue**(java.lang.String attName, java.lang.String attValue) method setAttributeValue
void	**setExplicit**() method setExplicit

Method Summary

void	**setImplicit**() method setImplicit
void	**setInitializerArgumentValue**(java.lang.String argName, java.lang.String argValue) method setInitializerArgumentValue
void	**setInstantiatedBy**(IRPOperation instantiatedBy) set property instantiatedBy

Methods inherited from interface com.telelogic.rhapsody.core.**IRPRelation**

addQualifier, getAssociationClass, getInverse, getIsNavigable, getIsSymmetric, getMultiplicity, getObjectAsObjectType, getOfClass, getOtherClass, getQualifier, getQualifiers, getQualifierType, getRelationLabel, getRelationLinkName, getRelationRoleName, getRelationType, getVisibility, isTypelessObject, makeUnidirect, removeQualifier, setInverse, setIsNavigable, setMultiplicity, setOfClass, setOtherClass, setQualifier, setQualifierType, setRelationLabel, setRelationLinkName, setRelationRoleName, setRelationType

Methods inherited from interface com.telelogic.rhapsody.core.**IRPUnit**

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner,

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addRelationToTheWhole

IRPRelation **addRelationToTheWhole**(java.lang.String relName)

> method addRelationToTheWhole
> **Throws:**
> > RhapsodyRuntimeException

---

getAttributeValue

java.lang.String **getAttributeValue**(java.lang.String attName)

> method getAttributeValue
> **Throws:**
> > RhapsodyRuntimeException

---

getInLinks

IRPCollection **getInLinks**()

> method getInLinks
> **Throws:**
> > RhapsodyRuntimeException

---

getInstantiatedBy

IRPOperation **getInstantiatedBy**()

> get property instantiatedBy
> **Throws:**
> > RhapsodyRuntimeException

---

getListOfInitializerArguments

IRPCollection **getListOfInitializerArguments**()

method getListOfInitializerArguments
Throws:
>   [RhapsodyRuntimeException](#)

getOutLinks

[IRPCollection](#) **getOutLinks**()

method getOutLinks
Throws:
>   [RhapsodyRuntimeException](#)

setAttributeValue

void **setAttributeValue**(java.lang.String attName,
                      java.lang.String attValue)

method setAttributeValue
Throws:
>   [RhapsodyRuntimeException](#)

setExplicit

void **setExplicit**()

method setExplicit
Throws:
>   [RhapsodyRuntimeException](#)

setImplicit

void **setImplicit**()

method setImplicit
Throws:
>   [RhapsodyRuntimeException](#)

setInitializerArgumentValue

void **setInitializerArgumentValue**(java.lang.String argName,
                                java.lang.String argValue)

method setInitializerArgumentValue
Throws:
>   [RhapsodyRuntimeException](#)

setInstantiatedBy

void **setInstantiatedBy**(IRPOperation instantiatedBy)

set property instantiatedBy
Throws:
RhapsodyRuntimeException

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | **METHOD**   DETAIL: FIELD | CONSTR | **METHOD**

com.telelogic.rhapsody.core
Interface IRPInstanceSlot

All Superinterfaces:
> IRPModelElement

```
public interface IRPInstanceSlot
extends IRPModelElement
```

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPInstanceValue	**addElementValue**(IRPModelElement val) method addElementValue
IRPLiteralSpecification	**addStringValue**(java.lang.String val) method addStringValue
IRPModelElement	**getSlotProperty**() get property slotProperty
IRPCollection	**getValues**() get property values
void	**setSlotProperty**(IRPModelElement slotProperty) set property slotProperty

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn,

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addElementValue

IRPInstanceValue **addElementValue**(IRPModelElement val)

method addElementValue
Throws:
RhapsodyRuntimeException

---

addStringValue

IRPLiteralSpecification **addStringValue**(java.lang.String val)

method addStringValue
Throws:
RhapsodyRuntimeException

---

getSlotProperty

IRPModelElement **getSlotProperty**()

get property slotProperty
Throws:
RhapsodyRuntimeException

---

getValues

<u>IRPCollection</u> **getValues**()

> get property values
> **Throws:**
> > <u>RhapsodyRuntimeException</u>

setSlotProperty

void **setSlotProperty**(<u>IRPModelElement</u> slotProperty)

> set property slotProperty
> **Throws:**
> > <u>RhapsodyRuntimeException</u>

Package | **Class** | **Use** | **Tree** | **Serialized** | **Deprecated** | **Index** | **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPInstanceSpecification

All Superinterfaces:
> IRPModelElement

---

public interface **IRPInstanceSpecification**
extends IRPModelElement

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPClassifier	**getClassifier**() get property classifier
IRPCollection	**getInstanceSlots**() get property instanceSlots
void	**populateSlots**() method populateSlots
void	**setClassifier**(IRPClassifier classifier) set property classifier

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks,

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getClassifier

IRPClassifier **getClassifier**()

> get property classifier
> **Throws:**
> > RhapsodyRuntimeException

---

getInstanceSlots

IRPCollection **getInstanceSlots**()

> get property instanceSlots
> **Throws:**
> > RhapsodyRuntimeException

---

populateSlots

void **populateSlots**()

> method populateSlots
> **Throws:**
> > RhapsodyRuntimeException

---

setClassifier

```
void setClassifier(IRPClassifier classifier)
```

> set property classifier
> **Throws:**
> > RhapsodyRuntimeException

---

com.telelogic.rhapsody.core
Interface IRPInstanceValue

All Superinterfaces:
> IRPModelElement, IRPValueSpecification

---

public interface **IRPInstanceValue**
extends IRPValueSpecification

The IRPInstanceValue interface is used in contexts where a single model element must be stored.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPModelElement	**getValue**() Returns the stored value.
void	**setValue**(IRPModelElement value) Sets the value to store.

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit,

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getValue

IRPModelElement **getValue**()

> Returns the stored value.
> **Returns:**
> > the stored value

setValue

void **setValue**(IRPModelElement value)

> Sets the value to store.
> **Parameters:**
> > value - the model element to store as the value

com.telelogic.rhapsody.core
Interface IRPIntegrator

```
public interface IRPIntegrator
```

Method Summary

java.lang.String	**getInterfaceName**() get property interfaceName

Method Detail

getInterfaceName

```
java.lang.String getInterfaceName()
```

> get property interfaceName
> **Throws:**
> > RhapsodyRuntimeException

com.telelogic.rhapsody.core
Interface IRPInteractionOccurrence

All Superinterfaces:
   IRPModelElement

---

public interface **IRPInteractionOccurrence**
extends IRPModelElement

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPCollection	**getMessagePoints**() get property messagePoints
IRPSequenceDiagram	**getReferenceSequenceDiagram**() get property referenceSequenceDiagram
void	**setReferenceSequenceDiagram**(IRPSequenceDiagram referenceSequenceDiagram) set property referenceSequenceDiagram

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit,

getInterfaceName   310

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getMessagePoints

IRPCollection **getMessagePoints**()

> get property messagePoints
> **Throws:**
> > RhapsodyRuntimeException

getReferenceSequenceDiagram

IRPSequenceDiagram **getReferenceSequenceDiagram**()

> get property referenceSequenceDiagram
> **Throws:**
> > RhapsodyRuntimeException

setReferenceSequenceDiagram

void **setReferenceSequenceDiagram**(IRPSequenceDiagram referenceSequenceDiagram)

> set property referenceSequenceDiagram
> **Throws:**
> > RhapsodyRuntimeException

Package   **Class**   **Use**   **Tree**   **Serialized**   **Deprecated**   **Index**   **Help**

PREV CLASS   **NEXT CLASS**   **FRAMES**   **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPInteractionOperand

All Superinterfaces:
>   IRPCollaboration, IRPModelElement

---

public interface **IRPInteractionOperand**
extends IRPCollaboration

The IRPInteractionOperand interface represents interaction operands in Rational Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPCollection	**getContainedMessages**() Returns a collection of all the messages contained in the interaction operand.
java.lang.String	**getInteractionConstraint**() Returns the constraint (guard condition) that was defined for the interaction operand.
void	**setInteractionConstraint**(java.lang.String interactionConstraint) Sets the constraint (guard condition) for the interaction operand.

Methods inherited from interface com.telelogic.rhapsody.core.IRPCollaboration
addActionBlock, addCancelledTimeout, addClassifierRole, addClassifierRoleByName, addClassifierRoleForInstance, addConditionMark, addCtor, addDataFlow, addDestructionEvent, addDtor, addDurationConstraint, addDurationObservation, addFoundMessage, addInteractionOccurrence, addInteractionOperator, addLostMessage, addMessage, addReplyMessage, addStateInvariant, addSystemBorder, addTimeConstraint, addTimeInterval, addTimeObservation, addTimeout, generateSequence, getActivationCondition, getActivationMode, getActivator, getAssociations, getClassifier, getConcurrentGroup, getExecutionOccurrences, getInteractionOccurrences, getInteractionOperators, getMessagePoints, getMessagePoints, getMessages, getMode, getPredecessor, getSuccessor

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getContainedMessages

IRPCollection **getContainedMessages**()

> Returns a collection of all the messages contained in the interaction operand.
> **Returns:**
> > all the messages contained in the interaction operand

getInteractionConstraint

java.lang.String **getInteractionConstraint**()

> Returns the constraint (guard condition) that was defined for the interaction operand.
> **Returns:**
> > the constraint (guard condition) that was defined for the interaction operand

setInteractionConstraint

void **setInteractionConstraint**(java.lang.String interactionConstraint)

Sets the constraint (guard condition) for the interaction operand.

Parameters:

interactionConstraint - the constraint (guard condition) to use for the interaction operand, for example, "x = 5"

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPInteractionOperator

All Superinterfaces:
> IRPModelElement

---

public interface **IRPInteractionOperator**
extends IRPModelElement

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
`IRPModelElement.OSLCLink`

Method Summary

`IRPCollection`	**`getInteractionOperands`**`()` get property interactionOperands
`java.lang.String`	**`getInteractionType`**`()` get property interactionType
`void`	**`setInteractionType`**`(java.lang.String interactionType)` set property interactionType

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
`addAssociation`, `addDependency`, `addDependencyBetween`, `addDependencyTo`, `addLinkToElement`, `addNewAggr`, `addProperty`, `addRedefines`, `addRemoteDependencyTo`, `addSpecificStereotype`, `addStereotype`, `becomeTemplateInstantiationOf`, `changeTo`, `clone`, `createOSLCLink`, `deleteDependency`, `deleteFromProject`, `deleteOSLCLink`, `errorMessage`, `findElementsByFullName`, `findNestedElement`, `findNestedElementRecursive`, `getAllTags`, `getAnnotations`, `getAssociationClasses`, `getBinaryID`, `getConstraints`, `getConstraintsByHim`, `getControlledFiles`, `getDecorationStyle`, `getDependencies`, `getDescription`, `getDescriptionHTML`, `getDescriptionPlainText`, `getDescriptionRTF`, `getDisplayName`, `getDisplayNameRTF`, `getErrorMessage`, `getFullPathName`, `getFullPathNameIn`, `getGUID`, `getHyperLinks`, `getIconFileName`, `getInterfaceName`, `getIsExternal`, `getIsOfMetaClass`, `getIsShowDisplayName`, `getIsUnresolved`, `getLocalTags`, `getMainDiagram`, `getMetaClass`, `getName`, `getNestedElements`, `getNestedElementsByMetaClass`, `getNestedElementsRecursive`, `getNewTermStereotype`, `getOfTemplate`, `getOSLCLinks`, `getOverlayIconFileName`, `getOverriddenProperties`, `getOverriddenPropertiesByPattern`, `getOwnedDependencies`, `getOwner`, `getProject`, `getPropertyValue`, `getPropertyValueConditional`, `getPropertyValueConditionalExplicit`,

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getInteractionOperands

IRPCollection **getInteractionOperands**()

> get property interactionOperands
> **Throws:**
> > RhapsodyRuntimeException

---

getInteractionType

java.lang.String **getInteractionType**()

> get property interactionType
> **Throws:**
> > RhapsodyRuntimeException

---

setInteractionType

void **setInteractionType**(java.lang.String interactionType)

> set property interactionType
> **Throws:**
> > RhapsodyRuntimeException

Package　**Class**　Use　Tree　Serialized　Deprecated　Index　Help

PREV CLASS　NEXT CLASS　　　　　　　　　　　FRAMES　NO FRAMES　All Classes
SUMMARY: NESTED | FIELD | CONSTR | METHOD　　　　DETAIL: FIELD | CONSTR | METHOD

getInteractionOperands　　　　　　　　　　　　　　　　　　　　　316

com.telelogic.rhapsody.core
Interface IRPInterfaceItem

All Superinterfaces:
>   IRPClassifier, IRPModelElement, IRPUnit

All Known Subinterfaces:
>   IRPEvent, IRPEventReception, IRPOperation

```
public interface IRPInterfaceItem
```
extends IRPClassifier

The IRPInterfaceItem interface represents the features shared by operations, events, and event receptions in Rational Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPArgument	**addArgument**(java.lang.String newVal) Adds a new argument to the end of the argument list.
IRPArgument	**addArgumentBeforePosition**(java.lang.String newVal, int pos) Adds a new argument at the specified position in the argument list.
IRPCollection	**getArguments**() Returns a collection of all the arguments for the operation (collection of IRPArgument objects).
java.lang.String	**getSignature**() Returns the signature of the operation.
java.lang.String	**getSignatureNoArgNames**() Returns the signature of the operation without the argument names.
java.lang.String	**getSignatureNoArgTypes**() Returns the signature of the operation without the argument types.
int	

Method Summary

	matchOnSignature(IRPInterfaceItem Item) Compares the signature of the operation with the signature of the operation that was provided as an argument.

Methods inherited from interface com.telelogic.rhapsody.core.IRPClassifier

addActivityDiagram, addAttribute, addFlowItems, addFlows, addGeneralization, addOperation, addRelation, addRelationTo, addStatechart, addUnidirectionalRelation, addUnidirectionalRelationTo, deleteAttribute, deleteFlowItems, deleteFlows, deleteGeneralization, deleteOperation, deleteRelation, findAttribute, findBaseClassifier, findDerivedClassifier, findGeneralization, findInterfaceItem, findNestedClassifier, findNestedClassifierRecursive, findRelation, findTrigger, getActivityDiagram, getAttributes, getAttributesIncludingBases, getBaseClassifiers, getBehavioralDiagrams, getDerivedClassifiers, getFlowItems, getFlows, getGeneralizations, getInterfaceItems, getInterfaceItemsIncludingBases, getLinks, getNestedClassifiers, getOperations, getPorts, getRelations, getRelationsIncludingBases, getSequenceDiagrams, getSourceArtifacts, getStatechart

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue,

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addArgument

IRPArgument **addArgument**(java.lang.String newVal)

> Adds a new argument to the end of the argument list. The method takes only a single argument - the name to use for the argument. The type of the argument is set by default to "int". To change the type of the argument, use the method IRPArgument.setType.
> **Parameters:**
> > newVal - the name to use for the new argument
> **Returns:**
> > the argument that was created

addArgumentBeforePosition

IRPArgument **addArgumentBeforePosition**(java.lang.String newVal,
                                        int pos)

> Adds a new argument at the specified position in the argument list. Like the addArgument method, the type of the argument is set by default to "int". To change the type of the argument, use the method IRPArgument.setType.
> **Parameters:**
> > newVal - the name to use for the new argument
> > pos - the position in the argument list where the new argument should be placed (1 signifies the first argument in the list)
> **Returns:**
> > the argument that was created

getArguments

IRPCollection **getArguments**()

> Returns a collection of all the arguments for the operation (collection of IRPArgument objects).
> **Returns:**
> > all the arguments for the operation

getSignature

java.lang.String **getSignature**()

> Returns the signature of the operation.
> **Returns:**
> > the signature of the operation

getSignatureNoArgNames

java.lang.String **getSignatureNoArgNames**()

> Returns the signature of the operation without the argument names.
> **Returns:**
> > the signature of the operation without the argument names

getSignatureNoArgTypes

java.lang.String **getSignatureNoArgTypes**()

> Returns the signature of the operation without the argument types.
> **Returns:**
> > the signature of the operation without the argument types

matchOnSignature

int **matchOnSignature**([IRPInterfaceItem](#) Item)

> Compares the signature of the operation with the signature of the operation that was provided as an argument. This method is useful if you are moving an operation from one class to another because Rhapsody will throw an exception if an operation with the identical signature already exists in the class.
> **Parameters:**
> > Item - the operation whose signature should be compared to the signature of the current operation
> **Returns:**
> > 1 if the two signatures are identical, 0 otherwise

com.telelogic.rhapsody.core
Interface IRPInternalOEMPlugin

public interface **IRPInternalOEMPlugin**

Method Summary

int	**activeProjectAboutToChange**() Notify the Plugin upon ActiveProjectAboutToChange
int	**activeProjectHasChanged**() Notify the Plugin upon ActiveProjectHasChanged
java.lang.String	**onMenuItemSelect**(java.lang.String menuItem) Selects a given menu item
java.lang.String	**onMenuItemSelectWithParameters**(java.lang.String menuItem, java.lang.String parameters) For internal use only.
int	**rhapPluginAnimationStopped**() Notify the Plugin upon RhapPluginAnimationStopped
int	**rhpPluginAnimationStarted**() Notify the Plugin upon RhpPluginAnimationStarted
int	**rhpPluginCleanup**() Performs cleanup of the Plugin
void	**rhpPluginDoCommand**(java.lang.String theCommand) Notify the Plugin to executes a command
int	**rhpPluginFinalCleanup**() Performs final cleanup of the plugin
int	**rhpPluginInit**() Initializes the plugin
int	**rhpPluginInvokeItem**() Invoke an item of the Plugin
void	**rhpPluginOnIDEBuildDone**(java.lang.String buildStatus) Notify the Plugin upon build done
int	**rhpPluginSetApplication**(<u>IRPApplication</u> pRPApp) Sets the IRPApplication of the plugin

Method Summary

int	**rhpSavingProject**() Notify the Plugin upon Rhapsody save

Method Detail

activeProjectAboutToChange

int **activeProjectAboutToChange**()

> Notify the Plugin upon ActiveProjectAboutToChange
> **Throws:**
>> RhapsodyRuntimeException

---

activeProjectHasChanged

int **activeProjectHasChanged**()

> Notify the Plugin upon ActiveProjectHasChanged
> **Throws:**
>> RhapsodyRuntimeException

---

onMenuItemSelect

java.lang.String **onMenuItemSelect**(java.lang.String menuItem)

> Selects a given menu item
> **Throws:**
>> RhapsodyRuntimeException

---

onMenuItemSelectWithParameters

java.lang.String **onMenuItemSelectWithParameters**(java.lang.String menuItem,
                                                   java.lang.String parameters)

> For internal use only.

---

rhapPluginAnimationStopped

int **rhapPluginAnimationStopped**()

> Notify the Plugin upon RhapPluginAnimationStopped

Throws:
    RhapsodyRuntimeException

rhpPluginAnimationStarted

int **rhpPluginAnimationStarted**()

Notify the Plugin upon RhpPluginAnimationStarted
Throws:
    RhapsodyRuntimeException

rhpPluginCleanup

int **rhpPluginCleanup**()

Performs cleanup of the Plugin
Throws:
    RhapsodyRuntimeException

rhpPluginDoCommand

void **rhpPluginDoCommand**(java.lang.String theCommand)

Notify the Plugin to executes a command
Throws:
    RhapsodyRuntimeException

rhpPluginFinalCleanup

int **rhpPluginFinalCleanup**()

Performs final cleanup of the plugin
Throws:
    RhapsodyRuntimeException

rhpPluginInit

int **rhpPluginInit**()

Initializes the plugin
Throws:
    RhapsodyRuntimeException

rhpPluginInvokeItem

int **rhpPluginInvokeItem**()

> Invoke an item of the Plugin
> **Throws:**
> > RhapsodyRuntimeException

---

rhpPluginOnIDEBuildDone

void **rhpPluginOnIDEBuildDone**(java.lang.String buildStatus)

> Notify the Plugin upon build done
> **Throws:**
> > RhapsodyRuntimeException

---

rhpPluginSetApplication

int **rhpPluginSetApplication**(IRPApplication pRPApp)

> Sets the IRPApplication of the plugin
> **Throws:**
> > RhapsodyRuntimeException

---

rhpSavingProject

int **rhpSavingProject**()

> Notify the Plugin upon Rhapsody save
> **Throws:**
> > RhapsodyRuntimeException

---

com.telelogic.rhapsody.core

Package **Class** <u>Use</u> <u>Tree</u> <u>Serialized</u> <u>Deprecated</u> <u>Index</u> <u>Help</u>

<u>PREV CLASS</u> <u>NEXT CLASS</u>   <u>FRAMES</u> <u>NO FRAMES</u> <u>All Classes</u>
SUMMARY: NESTED | FIELD | CONSTR | <u>METHOD</u>   DETAIL: FIELD | CONSTR | <u>METHOD</u>

com.telelogic.rhapsody.core
Interface IRPJavaPlugins

```
public interface IRPJavaPlugins
```

Method Summary

java.lang.String	**getInterfaceName**() get property interfaceName

Method Detail

getInterfaceName

```
java.lang.String getInterfaceName()
```

get property interfaceName
Throws:
<u>RhapsodyRuntimeException</u>

Package **Class** <u>Use</u> <u>Tree</u> <u>Serialized</u> <u>Deprecated</u> <u>Index</u> <u>Help</u>

<u>PREV CLASS</u> <u>NEXT CLASS</u>   <u>FRAMES</u> <u>NO FRAMES</u> <u>All Classes</u>
SUMMARY: NESTED | FIELD | CONSTR | <u>METHOD</u>   DETAIL: FIELD | CONSTR | <u>METHOD</u>

com.telelogic.rhapsody.core
Interface IRPLink

All Superinterfaces:
IRPModelElement, IRPUnit

---

public interface **IRPLink**
extends IRPUnit

The IRPLink interface represents links in Rational Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

java.lang.String	**getEnd1Multiplicity**() get property end1Multiplicity
java.lang.String	**getEnd1Name**() get property end1Name
java.lang.String	**getEnd2Multiplicity**() get property end2Multiplicity
java.lang.String	**getEnd2Name**() get property end2Name
IRPInstance	**getFrom**() get property from
IRPModelElement	**getFromElement**() get property fromElement
IRPPort	**getFromPort**() get property fromPort
IRPSysMLPort	**getFromSysMLPort**() get property fromSysMLPort
IRPRelation	

getInterfaceName 326

Method Summary

	getInstantiates() get property instantiates
IRPLink	**getOther**() get property other
IRPInstance	**getTo**() Returns the target of a link.
IRPModelElement	**getToElement**() get property toElement
IRPPort	**getToPort**() Returns the port through which a link reaches a target object.
IRPSysMLPort	**getToSysMLPort**() get property toSysMLPort
void	**setEnd1Multiplicity**(java.lang.String end1Multiplicity) set property end1Multiplicity
void	**setEnd1Name**(java.lang.String end1Name) set property end1Name
void	**setEnd2Multiplicity**(java.lang.String end2Multiplicity) set property end2Multiplicity
void	**setEnd2Name**(java.lang.String end2Name) set property end2Name
void	**setInstantiates**(_IRPRelation_ pVal) method setInstantiates

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass,

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getEnd1Multiplicity

java.lang.String **getEnd1Multiplicity**()

> get property end1Multiplicity
> **Throws:**
> > RhapsodyRuntimeException

getEnd1Name

java.lang.String **getEnd1Name**()

> get property end1Name
> **Throws:**
> > RhapsodyRuntimeException

getEnd2Multiplicity

java.lang.String **getEnd2Multiplicity**()

> get property end2Multiplicity
> **Throws:**
> > RhapsodyRuntimeException

getEnd2Name

`java.lang.String` **`getEnd2Name`**`()`

get property end2Name
Throws:
[RhapsodyRuntimeException](#)

---

getFrom

[IRPInstance](#) **`getFrom`**`()`

get property from
Throws:
[RhapsodyRuntimeException](#)

---

getFromElement

[IRPModelElement](#) **`getFromElement`**`()`

get property fromElement
Throws:
[RhapsodyRuntimeException](#)

---

getFromPort

[IRPPort](#) **`getFromPort`**`()`

get property fromPort
Throws:
[RhapsodyRuntimeException](#)

---

getFromSysMLPort

[IRPSysMLPort](#) **`getFromSysMLPort`**`()`

get property fromSysMLPort
Throws:
[RhapsodyRuntimeException](#)

---

getInstantiates

[IRPRelation](#) **`getInstantiates`**`()`

get property instantiates
Throws:

RhapsodyRuntimeException

getOther

IRPLink **getOther**()

> get property other
> **Throws:**
> > RhapsodyRuntimeException

getTo

IRPInstance **getTo**()

> Returns the target of a link. When a link is connected to an object directly or via a port on the object, the method returns the "to" object. When a link is connected to a port on a class, the method returns the "to" port.
> **Returns:**
> > the target of the link

getToElement

IRPModelElement **getToElement**()

> get property toElement
> **Throws:**
> > RhapsodyRuntimeException

getToPort

IRPPort **getToPort**()

> Returns the port through which a link reaches a target object. When a link is connected to a port on an object, the method returns the port on the "to" object. When a link is connected to a port on a class, or is connected to an object directly, the method returns null.
> **Returns:**
> > the port through which the link reaches its target object

getToSysMLPort

IRPSysMLPort **getToSysMLPort**()

> get property toSysMLPort
> **Throws:**
> > RhapsodyRuntimeException

setEnd1Multiplicity

void **setEnd1Multiplicity**(java.lang.String end1Multiplicity)

set property end1Multiplicity
Throws:
        RhapsodyRuntimeException

setEnd1Name

void **setEnd1Name**(java.lang.String end1Name)

set property end1Name
Throws:
        RhapsodyRuntimeException

setEnd2Multiplicity

void **setEnd2Multiplicity**(java.lang.String end2Multiplicity)

set property end2Multiplicity
Throws:
        RhapsodyRuntimeException

setEnd2Name

void **setEnd2Name**(java.lang.String end2Name)

set property end2Name
Throws:
        RhapsodyRuntimeException

setInstantiates

void **setInstantiates**(IRPRelation pVal)

method setInstantiates
Throws:
        RhapsodyRuntimeException

Package   **Class**   **Use**   **Tree**   **Serialized**   **Deprecated**   **Index**   **Help**

PREV CLASS   **NEXT CLASS**   **FRAMES**   **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

getToSysMLPort   331

com.telelogic.rhapsody.core
Interface IRPLiteralSpecification

All Superinterfaces:
> IRPModelElement, IRPValueSpecification

---

public interface **IRPLiteralSpecification**
extends IRPValueSpecification

The IRPLiteralSpecification interface is used in contexts where a single value must be stored.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

java.lang.String	**getValue**() Returns the stored value.
void	**setValue**(java.lang.String value) Sets the value to store.

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit,

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getValue

java.lang.String **getValue**()

Returns the stored value.
Returns:
the stored value

setValue

void **setValue**(java.lang.String value)

Sets the value to store.
Parameters:
value - the value to store

com.telelogic.rhapsody.core
Interface IRPMatrixLayout

All Superinterfaces:
 IRPModelElement, IRPUnit

---

public interface **IRPMatrixLayout**
extends IRPUnit

Nested Class Summary

static class	**IRPMatrixLayout.QueryOrElementsList** This class contains constant values for use with the methods setFromElementTypesUseQueryOrElementsList and setToElementTypesUseQueryOrElementsList.

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement

IRPModelElement.OSLCLink

Method Summary

IRPCollection	**getCellElementTypes**() Returns a collection of the element types that were specified to be displayed in the cells of the matrix.
IRPCollection	**getFromElementTypes**() Returns a collection of the "from" element types specified to be displayed in the matrix.
IRPTableLayout	**getFromElementTypesQueryToUse**() Returns the query that was specified to determine the "from" element types.
int	**getFromElementTypesUseQueryOrElementsList**() Checks whether a query or collection of element types was used to specify the "from" element types.
IRPCollection	**getToElementTypes**() Returns a collection of the "to" element types specified to be displayed in the matrix.
IRPTableLayout	**getToElementTypesQueryToUse**() Returns the query that was specified to determine the "to" element types.

Method Summary

int	**getToElementTypesUseQueryOrElementsList**() Checks whether a query or collection of element types was used to specify the "to" element types.
void	**setCellElementTypes**(IRPCollection pCollection) Specifies the element types to display in the cells of the matrix.
void	**setFromElementTypes**(IRPCollection pCollection) Specifies the "from" element types that should be displayed in the matrix.
void	**setFromElementTypesQueryToUse**(IRPTableLayout query) Specifies the query to use to determine the "from" element types for the matrix layout.
void	**setFromElementTypesUseQueryOrElementsList**(int queryOrElementsList) Specifies whether a query or collection of element types should be used to determine the "from" element types for the matrix layout.
void	**setToElementTypes**(IRPCollection pCollection) Specifies the "to" element types that should be displayed in the matrix.
void	**setToElementTypesQueryToUse**(IRPTableLayout query) Specifies the query to use to determine the "to" element types for the matrix layout.
void	**setToElementTypesUseQueryOrElementsList**(int queryOrElementsList) Specifies whether a query or collection of element types should be used to determine the "to" element types for the matrix layout.

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit,

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getCellElementTypes

IRPCollection **getCellElementTypes**()

> Returns a collection of the element types that were specified to be displayed in the cells of the matrix. The collection consists of strings (from the list of types displayed on the Cell Element Types tab of the Features window for matrix layouts).
> **Returns:**
> > the element types that were specified to be displayed in the cells of the matrix

getFromElementTypes

IRPCollection **getFromElementTypes**()

> Returns a collection of the "from" element types specified to be displayed in the matrix. The collection consists of strings (from the list of types displayed on the From Element Types tab of the Features window for matrix layouts).
> **Returns:**
> > the "from" element types specified to be displayed in the matrix

getFromElementTypesQueryToUse

IRPTableLayout **getFromElementTypesQueryToUse**()

> Returns the query that was specified to determine the "from" element types.
> **Returns:**
> > the query that was specified to determine the "from" element types for the matrix layout

getFromElementTypesUseQueryOrElementsList

`int` **`getFromElementTypesUseQueryOrElementsList`**`()`

> Checks whether a query or collection of element types was used to specify the "from" element types.
> **Returns:**
>> one of the constants contained in the class IRPMatrixLayout.QueryOrElementsList: QUERY if a query was used, ELEMENTS_LIST if a collection of element types was used.

getToElementTypes

[IRPCollection](#) **`getToElementTypes`**`()`

> Returns a collection of the "to" element types specified to be displayed in the matrix. The collection consists of strings (from the list of types displayed on the To Element Types tab of the Features window for matrix layouts).
> **Returns:**
>> the "to" element types specified to be displayed in the matrix

getToElementTypesQueryToUse

[IRPTableLayout](#) **`getToElementTypesQueryToUse`**`()`

> Returns the query that was specified to determine the "to" element types.
> **Returns:**
>> the query that was specified to determine the "to" element types for the matrix layout

getToElementTypesUseQueryOrElementsList

`int` **`getToElementTypesUseQueryOrElementsList`**`()`

> Checks whether a query or collection of element types was used to specify the "to" element types.
> **Returns:**
>> one of the constants contained in the class IRPMatrixLayout.QueryOrElementsList: QUERY if a query was used, ELEMENTS_LIST if a collection of element types was used.

setCellElementTypes

`void` **`setCellElementTypes`**`(`[IRPCollection](#) `pCollection)`

> Specifies the element types to display in the cells of the matrix. The parameter must be a collection of strings (from the list of types displayed on the Cell Element Types tab of the Features window for matrix layouts).
> **Parameters:**
>> `pCollection` - the element types to display in the cells of the matrix

setFromElementTypes

void **setFromElementTypes**([IRPCollection](#) pCollection)

> Specifies the "from" element types that should be displayed in the matrix. The parameter must be a collection of strings (from the list of element types displayed on the From Element Types tab of the Features window for matrix layouts).
> **Parameters:**
> > pCollection - the "from" element types that should be displayed in the matrix

setFromElementTypesQueryToUse

void **setFromElementTypesQueryToUse**([IRPTableLayout](#) query)

> Specifies the query to use to determine the "from" element types for the matrix layout.
> **Parameters:**
> > query - the query to use to determine the "from" element types for the matrix layout. To clear a previous query, use null for the parameter.

setFromElementTypesUseQueryOrElementsList

void **setFromElementTypesUseQueryOrElementsList**(int queryOrElementsList)

> Specifies whether a query or collection of element types should be used to determine the "from" element types for the matrix layout.
> **Parameters:**
> > queryOrElementsList - one of the constants contained in the class IRPMatrixLayout.QueryOrElementsList: QUERY if a query should be used, ELEMENTS_LIST if a collection of element types should be used.

setToElementTypes

void **setToElementTypes**([IRPCollection](#) pCollection)

> Specifies the "to" element types that should be displayed in the matrix. The parameter must be a collection of strings (from the list of types displayed on the To Element Types tab of the Features window for matrix layouts).
> **Parameters:**
> > pCollection - the "to" element types that should be displayed in the matrix

setToElementTypesQueryToUse

void **setToElementTypesQueryToUse**([IRPTableLayout](#) query)

> Specifies the query to use to determine the "to" element types for the matrix layout.
> **Parameters:**

query - the query to use to determine the "to" element types for the matrix layout. To clear a previous query, use null for the parameter.

setToElementTypesUseQueryOrElementsList

void **setToElementTypesUseQueryOrElementsList**(int queryOrElementsList)

Specifies whether a query or collection of element types should be used to determine the "to" element types for the matrix layout.

Parameters:

queryOrElementsList - one of the constants contained in the class IRPMatrixLayout.QueryOrElementsList: QUERY if a query should be used, ELEMENTS_LIST if a collection of element types should be used.

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: <u>NESTED</u> | FIELD | CONSTR | <u>METHOD</u>   DETAIL: FIELD | CONSTR | <u>METHOD</u>

---

com.telelogic.rhapsody.core
Class IRPMatrixLayout.QueryOrElementsList

```
java.lang.Object
  └─com.telelogic.rhapsody.core.IRPMatrixLayout.QueryOrElementsList
```

Enclosing interface:
> IRPMatrixLayout

---

public static final class **IRPMatrixLayout.QueryOrElementsList**
extends java.lang.Object

This class contains constant values for use with the methods setFromElementTypesUseQueryOrElementsList and setToElementTypesUseQueryOrElementsList.

Field Summary

static int	**ELEMENTS_LIST** When ELEMENTS_LIST is used as the parameter for the methods setFromElementTypesUseQueryOrElementsList and setToElementTypesUseQueryOrElementsList, it indicates that elements selected in the element types list will be used to specify the "from" element types or "to" element types for the matrix.
static int	**QUERY** When QUERY is used as the parameter for the methods setFromElementTypesUseQueryOrElementsList and setToElementTypesUseQueryOrElementsList, it indicates that a query will be used to determine the "from" element types or "to" element types for the matrix.

Constructor Summary

IRPMatrixLayout.QueryOrElementsList()

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

QUERY

`public static final int` **`QUERY`**

When QUERY is used as the parameter for the methods setFromElementTypesUseQueryOrElementsList and setToElementTypesUseQueryOrElementsList, it indicates that a query will be used to determine the "from" element types or "to" element types for the matrix.

See Also:
> [Constant Field Values](#)

ELEMENTS_LIST

`public static final int` **`ELEMENTS_LIST`**

When ELEMENTS_LIST is used as the parameter for the methods setFromElementTypesUseQueryOrElementsList and setToElementTypesUseQueryOrElementsList, it indicates that elements selected in the element types list will be used to specify the "from" element types or "to" element types for the matrix.

See Also:
> [Constant Field Values](#)

Constructor Detail

IRPMatrixLayout.QueryOrElementsList

`public` **`IRPMatrixLayout.QueryOrElementsList`**`()`

[Package](#)	**Class**	[Use](#)	[Tree](#)	[Serialized](#)	[Deprecated](#)	[Index](#)	[Help](#)

com.telelogic.rhapsody.core
Class IRPMatrixView.ContentFormat

```
java.lang.Object
```
  └─**com.telelogic.rhapsody.core.IRPMatrixView.ContentFormat**

Enclosing interface:
    IRPMatrixView

---

public static final class **IRPMatrixView.ContentFormat**
extends java.lang.Object

This class contains values that specify export format

Field Summary

static java.lang.String	**CSV** Export in Comma Separated Value (CSV) format.
static java.lang.String	**HTML** Export in HTML format.
static java.lang.String	**XML** Export in XML format.

Constructor Summary

IRPMatrixView.ContentFormat()

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

HTML

`public static final java.lang.String `**`HTML`**

Export in HTML format. Exported only string representations.

See Also:
[Constant Field Values](#)

XML

`public static final java.lang.String `**`XML`**

Export in XML format. For each model element, its GUID is exported as well.

See Also:
[Constant Field Values](#)

CSV

`public static final java.lang.String `**`CSV`**

Export in Comma Separated Value (CSV) format. Exported only string representations.

See Also:
[Constant Field Values](#)

Constructor Detail

IRPMatrixView.ContentFormat

`public `**`IRPMatrixView.ContentFormat`**`()`

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS   **NEXT CLASS**   **FRAMES**   **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPMatrixView

All Superinterfaces:
> IRPModelElement, IRPUnit

---

public interface **IRPMatrixView**
extends IRPUnit

The IRPMatrixView interface represents Matrix View elements in Rhapsody models.

Nested Class Summary

static class	**IRPMatrixView.ContentFormat** This class contains values that specify export format

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

java.lang.String	**getContent**(java.lang.String format) Retrieves the content of the matrix in the specified format.
IRPCollection	**getFromScope**() method GetFromScope
java.lang.String	**getHTMLContent**() Returns the content of the matrix as HTML.
IRPCollection	**getImageCollection**(java.lang.String sFolder, java.lang.String sFilename, java.lang.String sExtension) method GetImageCollection
int	**getIncludeDescendantsFromScope**() get property includeDescendantsFromScope
int	**getIncludeDescendantsToScope**() get property includeDescendantsToScope
IRPMatrixLayout	**getItsMatrixLayout**() method GetItsMatrixLayout

---

Method Summary

[IRPCollection](#)	**getToScope**() method GetToScope
void	**open**() method open
void	**setFromScope**([IRPCollection](#) pCollection) Specifies the "from" scope to use for this matrix view.
void	**setIncludeDescendantsFromScope**(int includeDescendantsFromScope) set property includeDescendantsFromScope
void	**setIncludeDescendantsToScope**(int includeDescendantsToScope) set property includeDescendantsToScope
void	**setItsMatrixLayout**([IRPMatrixLayout](#) pVal) Specifies the matrix layout to use for this matrix view.
void	**setToScope**([IRPCollection](#) pCollection) Specifies the "to" scope to use for this matrix view.

Methods inherited from interface com.telelogic.rhapsody.core.**[IRPUnit](#)**

[copyToAnotherProject](#), [getAddToModelMode](#), [getCMHeader](#), [getCMState](#), [getCurrentDirectory](#), [getFilename](#), [getIncludeInNextLoad](#), [getIsStub](#), [getLanguage](#), [getLastModifiedTime](#), [getNestedSaveUnits](#), [getNestedSaveUnitsCount](#), [getStructureDiagrams](#), [isReadOnly](#), [isReferenceUnit](#), [isSeparateSaveUnit](#), [load](#), [moveToAnotherProjectLeaveAReference](#), [referenceToAnotherProject](#), [save](#), [setCMHeader](#), [setFilename](#), [setIncludeInNextLoad](#), [setLanguage](#), [setReadOnly](#), [setSeparateSaveUnit](#), [setUnitPath](#), [unload](#)

Methods inherited from interface com.telelogic.rhapsody.core.**[IRPModelElement](#)**

[addAssociation](#), [addDependency](#), [addDependencyBetween](#), [addDependencyTo](#), [addLinkToElement](#), [addNewAggr](#), [addProperty](#), [addRedefines](#), [addRemoteDependencyTo](#), [addSpecificStereotype](#), [addStereotype](#), [becomeTemplateInstantiationOf](#), [changeTo](#), [clone](#), [createOSLCLink](#), [deleteDependency](#), [deleteFromProject](#), [deleteOSLCLink](#), [errorMessage](#), [findElementsByFullName](#), [findNestedElement](#), [findNestedElementRecursive](#), [getAllTags](#), [getAnnotations](#), [getAssociationClasses](#), [getBinaryID](#), [getConstraints](#), [getConstraintsByHim](#), [getControlledFiles](#), [getDecorationStyle](#), [getDependencies](#), [getDescription](#), [getDescriptionHTML](#), [getDescriptionPlainText](#), [getDescriptionRTF](#), [getDisplayName](#), [getDisplayNameRTF](#), [getErrorMessage](#), [getFullPathName](#), [getFullPathNameIn](#), [getGUID](#), [getHyperLinks](#), [getIconFileName](#), [getInterfaceName](#), [getIsExternal](#), [getIsOfMetaClass](#), [getIsShowDisplayName](#), [getIsUnresolved](#), [getLocalTags](#), [getMainDiagram](#), [getMetaClass](#), [getName](#), [getNestedElements](#), [getNestedElementsByMetaClass](#), [getNestedElementsRecursive](#), [getNewTermStereotype](#), [getOfTemplate](#), [getOSLCLinks](#), [getOverlayIconFileName](#), [getOverriddenProperties](#), [getOverriddenPropertiesByPattern](#), [getOwnedDependencies](#), [getOwner](#), [getProject](#), [getPropertyValue](#), [getPropertyValueConditional](#), [getPropertyValueConditionalExplicit](#), [getPropertyValueExplicit](#), [getRedefines](#), [getReferences](#), [getRemoteDependencies](#), [getRemoteURI](#), [getRequirementTraceabilityHandle](#), [getSaveUnit](#), [getStereotype](#), [getStereotypes](#), [getTag](#), [getTemplateParameters](#), [getTi](#), [getToolTipHTML](#), [getUserDefinedMetaClass](#), [hasNestedElements](#), [hasPanelWidget](#), [highLightElement](#), [isATemplate](#), [isDescriptionRTF](#), [isDisplayNameRTF](#), [isModified](#), [isRemote](#), [locateInBrowser](#), [lockOnDesignManager](#), [openFeaturesDialog](#), [removeProperty](#), [removeRedefines](#), [removeStereotype](#), [setDecorationStyle](#), [setDescription](#), [setDescriptionAndHyperlinks](#), [setDescriptionHTML](#), [setDescriptionRTF](#), [setDisplayName](#), [setDisplayNameRTF](#), [setGUID](#),

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getContent

```
java.lang.String getContent(java.lang.String format)
```

Retrieves the content of the matrix in the specified format. The value of the parameter should be one of the values defined in the class IRPMatrixView.ContentFormat. Note that when you call this method, the matrix is also displayed in Rhapsody.

Parameters:

format - one of the formats defined in the class IRPMatrixView.ContentFormat, for example, IRPMatrixView.ContentFormat.CSV

Returns:

the content of the matrix in the specified format

getFromScope

```
IRPCollection getFromScope()
```

method GetFromScope

Throws:

RhapsodyRuntimeException

getHTMLContent

```
java.lang.String getHTMLContent()
```

Returns the content of the matrix as HTML. The content returned begins and ends with the "table" tag. Note that when this method is called, the matrix is opened in Rational Rhapsody before the HTML is returned.

Returns:

the content of the matrix as HTML

getImageCollection

```
IRPCollection getImageCollection(java.lang.String sFolder,
                                 java.lang.String sFilename,
                                 java.lang.String sExtension)
```

method GetImageCollection
Throws:
  RhapsodyRuntimeException

---

getItsMatrixLayout

IRPMatrixLayout **getItsMatrixLayout**()

  method GetItsMatrixLayout
  Throws:
    RhapsodyRuntimeException

---

getToScope

IRPCollection **getToScope**()

  method GetToScope
  Throws:
    RhapsodyRuntimeException

---

setFromScope

void **setFromScope**(IRPCollection pCollection)

  Specifies the "from" scope to use for this matrix view.
  Parameters:
    pCollection - the "from" scope to use for this matrix view. Note that the parameter is a
    Rhapsody collection, but at the moment, only the first value in the collection is used for the
    "from" scope.

---

setItsMatrixLayout

void **setItsMatrixLayout**(IRPMatrixLayout pVal)

  Specifies the matrix layout to use for this matrix view.
  Parameters:
    pVal - the matrix layout to use for this matrix view

---

setToScope

void **setToScope**(IRPCollection pCollection)

  Specifies the "to" scope to use for this matrix view.
  Parameters:

pCollection - the "to" scope to use for this matrix view. Note that the parameter is a Rhapsody collection, but at the moment, only the first value in the collection is used for the "to" scope.

getIncludeDescendantsFromScope

int **getIncludeDescendantsFromScope**()

get property includeDescendantsFromScope
Throws:
RhapsodyRuntimeException

getIncludeDescendantsToScope

int **getIncludeDescendantsToScope**()

get property includeDescendantsToScope
Throws:
RhapsodyRuntimeException

open

void **open**()

method open
Throws:
RhapsodyRuntimeException

setIncludeDescendantsFromScope

void **setIncludeDescendantsFromScope**(int includeDescendantsFromScope)

set property includeDescendantsFromScope
Throws:
RhapsodyRuntimeException

setIncludeDescendantsToScope

void **setIncludeDescendantsToScope**(int includeDescendantsToScope)

set property includeDescendantsToScope
Throws:
RhapsodyRuntimeException

com.telelogic.rhapsody.core
Interface IRPMessage

All Superinterfaces:
> IRPModelElement

All Known Subinterfaces:
> IRPActionBlock, IRPConditionMark, IRPDestructionEvent

```
public interface IRPMessage
```
extends IRPModelElement

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPExecutionOccurrence	**addSourceExecutionOccurrence**() method addSourceExecutionOccurrence
IRPExecutionOccurrence	**addTargetExecutionOccurrence**() method addTargetExecutionOccurrence
IRPCollection	**getActualParameterList**() get property actualParameterList
IRPAssociationRole	**getCommunicationConnection**() get property communicationConnection
java.lang.String	**getCondition**() get property condition
java.lang.String	**getDurationConstraint**() Gets the text of the Duration Constraint.
java.lang.String	**getDurationObservation**() Gets the text of the Duration Observation.
IRPSysMLPort	**getFlowPort**() get property flowPort

Method Summary

[IRPInterfaceItem](#)	**getFormalInterfaceItem**() get property formalInterfaceItem
[IRPModelElement](#)	**getFormalType**() Returns the model element associated with an action block, condition mark, timeout, or canceled timeout, in a sequence diagram.
java.lang.String	**getInvariant**() Gets the text of the Invariant field for the state invariant.
java.lang.String	**getMessageType**() get property messageType
[IRPPort](#)	**getPort**() get property Port
java.lang.String	**getReturnValue**() get property returnValue
java.lang.String	**getSequenceNumber**() get property sequenceNumber
java.lang.String	**getSignature**() method getSignature
[IRPClassifierRole](#)	**getSource**() get property source
[IRPExecutionOccurrence](#)	**getSourceExecutionOccurrence**() get property sourceExecutionOccurrence
[IRPClassifierRole](#)	**getTarget**() get property target
[IRPExecutionOccurrence](#)	**getTargetExecutionOccurrence**() get property targetExecutionOccurrence
java.lang.String	**getTimeConstraint**() Gets the text for the Time Constraint that was applied to this state variant.
java.lang.String	**getTimeObservation**() Gets the text of the Time Observation.
java.lang.String	**getTimerValue**() get property timerValue
void	**reroute**() method reroute
void	**setActualParameterList**([IRPCollection](#) pVal) method setActualParameterList
void	**setDurationConstraint**(java.lang.String durationConstraint) Modifies the text of this Duration Constraint.
void	**setDurationObservation**(java.lang.String durationObservation) Modifies the text of this Duration Observation.

Method Summary

void	**setFlowPort**([IRPSysMLPort](#) flowPort) set property flowPort
void	**setFormalInterfaceItem**([IRPInterfaceItem](#) newVal) Sets the realization of a message.
void	**setFormalType**([IRPModelElement](#) formalType) Used to specify the model element that should be associated with an action block, condition mark, timeout, or canceled timeout, in a sequence diagram.
void	**setInvariant**(java.lang.String invariant) Modifies the text of the Invariant field for the state invariant.
void	**setPort**([IRPPort](#) port) set property Port
void	**setReturnValue**(java.lang.String returnValue) set property returnValue
void	**setTimeConstraint**(java.lang.String timeConstraint) Modifies the text of this Time Constraint.
void	**setTimeObservation**(java.lang.String timeObservation) Modifies the text of this Time Observation.
void	**setTimerValue**(java.lang.String timerValue) set property timerValue

Methods inherited from interface com.telelogic.rhapsody.core.[IRPModelElement](#)

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation,

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
unlockOnDesignManager

Method Detail

addSourceExecutionOccurrence

IRPExecutionOccurrence **addSourceExecutionOccurrence**()

> method addSourceExecutionOccurrence
> **Throws:**
> > RhapsodyRuntimeException

addTargetExecutionOccurrence

IRPExecutionOccurrence **addTargetExecutionOccurrence**()

> method addTargetExecutionOccurrence
> **Throws:**
> > RhapsodyRuntimeException

getActualParameterList

IRPCollection **getActualParameterList**()

> get property actualParameterList
> **Throws:**
> > RhapsodyRuntimeException

getCommunicationConnection

IRPAssociationRole **getCommunicationConnection**()

> get property communicationConnection
> **Throws:**
> > RhapsodyRuntimeException

getCondition

java.lang.String **getCondition**()

> get property condition
> **Throws:**

RhapsodyRuntimeException

---

getDurationConstraint

`java.lang.String` **getDurationConstraint**()

Gets the text of the Duration Constraint.
Returns:
the text of the Duration Constraint

---

getDurationObservation

`java.lang.String` **getDurationObservation**()

Gets the text of the Duration Observation.
Returns:
the text of the Duration Observation

---

getFlowPort

IRPSysMLPort **getFlowPort**()

get property flowPort
Throws:
RhapsodyRuntimeException

---

getFormalInterfaceItem

IRPInterfaceItem **getFormalInterfaceItem**()

get property formalInterfaceItem
Throws:
RhapsodyRuntimeException

---

getFormalType

IRPModelElement **getFormalType**()

Returns the model element associated with an action block, condition mark, timeout, or canceled timeout, in a sequence diagram.
Returns:
the model element associated with an action block, condition mark, timeout, or canceled timeout, in a sequence diagram

---

getInvariant

java.lang.String **getInvariant**()

>   Gets the text of the Invariant field for the state invariant.
>   **Returns:**
>   >   the text of the Invariant field

---

getMessageType

java.lang.String **getMessageType**()

>   get property messageType
>   **Throws:**
>   >   RhapsodyRuntimeException

---

getPort

IRPPort **getPort**()

>   get property Port
>   **Throws:**
>   >   RhapsodyRuntimeException

---

getReturnValue

java.lang.String **getReturnValue**()

>   get property returnValue
>   **Throws:**
>   >   RhapsodyRuntimeException

---

getSequenceNumber

java.lang.String **getSequenceNumber**()

>   get property sequenceNumber
>   **Throws:**
>   >   RhapsodyRuntimeException

---

getSignature

java.lang.String **getSignature**()

>   method getSignature

Throws:
        RhapsodyRuntimeException

getSource

IRPClassifierRole **getSource**()

    get property source
    Throws:
        RhapsodyRuntimeException

getSourceExecutionOccurrence

IRPExecutionOccurrence **getSourceExecutionOccurrence**()

    get property sourceExecutionOccurrence
    Throws:
        RhapsodyRuntimeException

getTarget

IRPClassifierRole **getTarget**()

    get property target
    Throws:
        RhapsodyRuntimeException

getTargetExecutionOccurrence

IRPExecutionOccurrence **getTargetExecutionOccurrence**()

    get property targetExecutionOccurrence
    Throws:
        RhapsodyRuntimeException

getTimeConstraint

java.lang.String **getTimeConstraint**()

    Gets the text for the Time Constraint that was applied to this state variant.
    Returns:
        the text for the Time Constraint that was applied to this state variant

getTimeObservation

java.lang.String **getTimeObservation**()

> Gets the text of the Time Observation.
> **Returns:**
> > the text of the Time Observation

getTimerValue

java.lang.String **getTimerValue**()

> get property timerValue
> **Throws:**
> > RhapsodyRuntimeException

reroute

void **reroute**()

> method reroute
> **Throws:**
> > RhapsodyRuntimeException

setActualParameterList

void **setActualParameterList**(IRPCollection pVal)

> method setActualParameterList
> **Throws:**
> > RhapsodyRuntimeException

setDurationConstraint

void **setDurationConstraint**(java.lang.String durationConstraint)

> Modifies the text of this Duration Constraint.
> **Parameters:**
> > durationConstraint - the text to use for the Duration Constraint

setDurationObservation

void **setDurationObservation**(java.lang.String durationObservation)

> Modifies the text of this Duration Observation.

getTimeObservation 358

Parameters:

        `durationObservation` - the text to use for the Duration Observation

---

setFlowPort

void **setFlowPort**(<u>IRPSysMLPort</u> flowPort)

    set property flowPort
    Throws:
        <u>RhapsodyRuntimeException</u>

---

setFormalInterfaceItem

void **setFormalInterfaceItem**(<u>IRPInterfaceItem</u> newVal)

    Sets the realization of a message.
    Parameters:
        `newVal` - the operation or other IRPInterfaceItem object to use for the realization of the message
    Throws:
        <u>RhapsodyRuntimeException</u>

---

setFormalType

void **setFormalType**(<u>IRPModelElement</u> formalType)

    Used to specify the model element that should be associated with an action block, condition mark, timeout, or canceled timeout, in a sequence diagram.
    Parameters:
        `formalType` - the model element that should be associated with this sequence diagram element

---

setInvariant

void **setInvariant**(java.lang.String invariant)

    Modifies the text of the Invariant field for the state invariant.
    Parameters:
        `invariant` - the text to use for the Invariant field

---

setPort

void **setPort**(<u>IRPPort</u> port)

    set property Port

setDurationObservation

Throws:
> RhapsodyRuntimeException

---

setReturnValue

void **setReturnValue**(java.lang.String returnValue)

> set property returnValue
> **Throws:**
> > RhapsodyRuntimeException

---

setTimeConstraint

void **setTimeConstraint**(java.lang.String timeConstraint)

> Modifies the text of this Time Constraint.
> **Parameters:**
> > timeConstraint - the text to use for this Time Constraint

---

setTimeObservation

void **setTimeObservation**(java.lang.String timeObservation)

> Modifies the text of this Time Observation.
> **Parameters:**
> > timeObservation - the text to use for the Time Observation

---

setTimerValue

void **setTimerValue**(java.lang.String timerValue)

> set property timerValue
> **Throws:**
> > RhapsodyRuntimeException

Package **Class** **[Use](Use)** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS NEXT CLASS   **FRAMES NO FRAMES All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPMessagePoint

All Superinterfaces:
> IRPModelElement

---

public interface **IRPMessagePoint**
extends IRPModelElement

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPClassifierRole	**getClassifierRole**() method getClassifierRole
IRPInteractionOccurrence	**getInteractionOccurrence**() get property interactionOccurrence
IRPInteractionOperator	**getInteractionOperator**() get property interactionOperator
IRPMessage	**getMessage**() get property message
java.lang.String	**getType**() get property type

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn,

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getClassifierRole

IRPClassifierRole **getClassifierRole**()

> method getClassifierRole
> **Throws:**
> > RhapsodyRuntimeException

---

getInteractionOccurrence

IRPInteractionOccurrence **getInteractionOccurrence**()

> get property interactionOccurrence
> **Throws:**
> > RhapsodyRuntimeException

---

getInteractionOperator

IRPInteractionOperator **getInteractionOperator**()

> get property interactionOperator
> **Throws:**
> > RhapsodyRuntimeException

---

getMessage

IRPMessage **getMessage**()

> get property message
> **Throws:**
>> RhapsodyRuntimeException

---

getType

java.lang.String **getType**()

> get property type
> **Throws:**
>> RhapsodyRuntimeException

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPModelElement

All Known Subinterfaces:
> IRPAcceptEventAction, IRPAcceptTimeEvent, IRPAction, IRPActionBlock, IRPActivityDiagram, IRPActor, IRPAnnotation, IRPArgument, IRPAssociationClass, IRPAssociationRole, IRPAttribute, IRPCallOperation, IRPClass, IRPClassifier, IRPClassifierRole, IRPCollaboration, IRPCollaborationDiagram, IRPComment, IRPComponent, IRPComponentDiagram, IRPComponentInstance, IRPConditionMark, IRPConfiguration, IRPConnector, IRPConstraint, IRPContextSpecification, IRPControlledFile, IRPDependency, IRPDeploymentDiagram, IRPDestructionEvent, IRPDiagram, IRPEnumerationLiteral, IRPEvent, IRPEventReception, IRPExecutionOccurrence, IRPFile, IRPFileFragment, IRPFlow, IRPFlowchart, IRPFlowItem, IRPGeneralization, IRPGuard, IRPHyperLink, IRPInstance, IRPInstanceSlot, IRPInstanceSpecification, IRPInstanceValue, IRPInteractionOccurrence, IRPInteractionOperand, IRPInteractionOperator, IRPInterfaceItem, IRPLink, IRPLiteralSpecification, IRPMatrixLayout, IRPMatrixView, IRPMessage, IRPMessagePoint, IRPModule, IRPNode, IRPObjectModelDiagram, IRPObjectNode, IRPOperation, IRPPackage, IRPPanelDiagram, IRPPin, IRPPort, IRPProfile, IRPProject, IRPRelation, IRPRequirement, IRPSendAction, IRPSequenceDiagram, IRPState, IRPStatechart, IRPStatechartDiagram, IRPStateVertex, IRPStereotype, IRPStructureDiagram, IRPSwimlane, IRPSysMLPort, IRPTableLayout, IRPTableView, IRPTag, IRPTemplateInstantiation, IRPTemplateInstantiationParameter, IRPTemplateParameter, IRPTimingDiagram, IRPTransition, IRPTrigger, IRPType, IRPUnit, IRPUseCase, IRPUseCaseDiagram, IRPValueSpecification, IRPVariable

---

public interface **IRPModelElement**

The IRPModelElement interface represents an element in a Rational Rhapsody model, and its methods reflect the behavior shared by the various types of model elements. The specific types of elements in a model are derived from this interface.

Nested Class Summary

static class	**IRPModelElement.OSLCLink** Constant values used with elements of this type

Method Summary

IRPAssociationClass	**addAssociation**(IRPRelation end1, IRPRelation end2, java.lang.String name) Creates an association class using the specified IRPRelation elements.

getType   364

Method Summary

[IRPDependency](#)	**addDependency**(java.lang.String dependsOnName, java.lang.String dependsOnType) 　　　Adds a dependency from the model element to the model element specified by the parameters.
[IRPDependency](#)	**addDependencyBetween**([IRPModelElement](#) dependent, [IRPModelElement](#) dependsOn) 　　　Creates a dependency between the two specified elements.
[IRPDependency](#)	**addDependencyTo**([IRPModelElement](#) element) 　　　Adds a dependency upon another model element.
[IRPLink](#)	**addLinkToElement**([IRPModelElement](#) toElement, [IRPRelation](#) assoc, [IRPModelElement](#) fromPort, [IRPModelElement](#) toPort) 　　　Creates a link between this model element and the model element specified as an argument.
[IRPModelElement](#)	**addNewAggr**(java.lang.String metaType, java.lang.String name) 　　　Adds a new model element to the current element, for example, adding a class to a package.
void	**addProperty**(java.lang.String propertyKey, java.lang.String propertyType, java.lang.String propertyValue) 　　　Adds a new property to the model element and assigns a value to it.
void	**addRedefines**([IRPModelElement](#) newRedefine) 　　　method addRedefines
[IRPDependency](#)	**addRemoteDependencyTo**([IRPModelElement](#) element, java.lang.String linkType) 　　　For Design Manager projects, used to create a dependency from a model element to a remote element.
void	**addSpecificStereotype**([IRPStereotype](#) stereotype) 　　　Applies the specified stereotype to the model element.
[IRPStereotype](#)	**addStereotype**(java.lang.String name, java.lang.String metaType) 　　　Applies the specified stereotype to the model element if the project contains a stereotype with the name specified and applicable to the metaclass specified.
void	**becomeTemplateInstantiationOf**([IRPModelElement](#) newVal) 　　　Makes the current model element a template instantiation of the specified template.
[IRPModelElement](#)	**changeTo**(java.lang.String metaClass) 　　　Changes the model element to the type of element specified by the parameter provided.
[IRPModelElement](#)	**clone**(java.lang.String name, [IRPModelElement](#) newOwner) 　　　Clones a model element.
void	**createOSLCLink**(java.lang.String type, java.lang.String purl) 　　　Creates an OSLC link between the element and the element represented by the specified URL.
void	

Method Summary

	deleteDependency([IRPDependency](#) dependency) Deletes the specified dependency from the model.
void	**deleteFromProject**() Deletes the current model element from the model.
void	**deleteOSLCLink**(java.lang.String type, java.lang.String purl) Deletes the specified OSLC link from the model.
java.lang.String	**errorMessage**() Returns error message for last method called.
[IRPModelElement](#)	**findElementsByFullName**(java.lang.String name, java.lang.String metaClass) Searches for the specified model element in the specified path under the current model element.
[IRPModelElement](#)	**findNestedElement**(java.lang.String name, java.lang.String metaClass) Searches for the specified model element.
[IRPModelElement](#)	**findNestedElementRecursive**(java.lang.String name, java.lang.String metaClass) Searches recursively for the specified model element.
[IRPCollection](#)	**getAllTags**() Returns a collection of all the element's tags.
[IRPCollection](#)	**getAnnotations**() Returns all of the element's annotations.
[IRPCollection](#)	**getAssociationClasses**() Returns a collection of all the association classes directly beneath this model element.
byte[]	**getBinaryID**() Returns the GUID of the model element as an array of bytes, as opposed to the method getGUID, which returns the GUID as a string.
[IRPCollection](#)	**getConstraints**() Returns all of the element's constraints.
[IRPCollection](#)	**getConstraintsByHim**() For internal use only.
[IRPCollection](#)	**getControlledFiles**() Returns a collection of all the element's controlled files.
java.lang.String	**getDecorationStyle**() Returns the name of the decoration style currently associated with the model element.
[IRPCollection](#)	**getDependencies**() Returns all of the element's dependencies.
java.lang.String	**getDescription**()

Method Summary

	Returns the description defined for the element.
java.lang.String	**getDescriptionHTML**() Returns HTML representation of the element description.
java.lang.String	**getDescriptionPlainText**() Returns the description defined for the element in plain text format.
java.lang.String	**getDescriptionRTF**() Returns the description defined for the element in RTF format.
java.lang.String	**getDisplayName**() Returns the label of the model element.
java.lang.String	**getDisplayNameRTF**() Returns the label of the model element as an RTF string.
java.lang.String	**getErrorMessage**() Returns error message for last method called.
java.lang.String	**getFullPathName**() Returns the full path name of the model element.
java.lang.String	**getFullPathNameIn**() Retrieves the full path name of the element as a string in the following format: (class) in (package).
java.lang.String	**getGUID**() Returns the GUID of the model element.
IRPCollection	**getHyperLinks**() Returns a collection of all the hyperlinks associated with the element.
java.lang.String	**getIconFileName**() Returns the full path of the graphic file used to represent elements of this type in the browser, for example, D:\programs\rhapsody80\Share\PredefinedPictures\Icons\RhapsodyIcons_72.gif.
java.lang.String	**getInterfaceName**() Returns the name of the API interface corresponding to the current element, for example, IRPClass for a class element, IRPOperation for an operation element.
int	**getIsExternal**() Checks whether the element is an "external" element - corresponds to the value of the property UseAsExternal.
int	**getIsOfMetaClass**(java.lang.String metaClass) Indicates whether the model element is based on the metaclass provided as a parameter.
int	**getIsShowDisplayName**() Checks whether the model element is configured to have its label displayed instead of its name whenever it is included in a diagram.
int	**getIsUnresolved**() Checks if the element is an element that can't be resolved by Rhapsody.

Method Summary

[IRPCollection](#)	**getLocalTags**() Returns a collection of the tags that were created locally for this model element.
[IRPDiagram](#)	**getMainDiagram**() Returns the "main" diagram for the element.
java.lang.String	**getMetaClass**() Gets the name of the metaclass on which the model element is based.
java.lang.String	**getName**() Returns the name of the element.
[IRPCollection](#)	**getNestedElements**() Gets a collection of all the model elements that are directly under the current element.
[IRPCollection](#)	**getNestedElementsByMetaClass**(java.lang.String metaClass, int recursive) Retrieves all of the model elements of the specified type below the current element.
[IRPCollection](#)	**getNestedElementsRecursive**() Returns a collection that consists of the current element and all of the model elements below it.
[IRPStereotype](#)	**getNewTermStereotype**() If a "new term" stereotype has been applied to the element, returns the stereotype.
[IRPModelElement](#)	**getOfTemplate**() If the element is an instantiation of a template, this method returns the template that it instantiates.
[IRPCollection](#)	**getOSLCLinks**() Returns a collection of all the element's OSLC links.
java.lang.String	**getOverlayIconFileName**() Returns the full path of the graphic file that is used as an overlay on this specific model element, on top of the regular icon that represent elements of this type in the browser.
[IRPCollection](#)	**getOverriddenProperties**(int recursive) Returns a collection of all the properties whose value was overridden for this model element.
[IRPCollection](#)	**getOverriddenPropertiesByPattern**(java.lang.String pattern, int localyOverridenOnly, int withDefaultValues) method getOverriddenPropertiesByPattern
[IRPCollection](#)	**getOwnedDependencies**() Returns all of the dependencies that are owned by the element.
[IRPModelElement](#)	**getOwner**() Returns the model element that owns this model element.

Method Summary

[IRPProject](#)	**getProject**() Returns the project that the current element belongs to.
java.lang.String	**getPropertyValue**(java.lang.String propertyKey) Returns the value of the specified property for the model element.
java.lang.String	**getPropertyValueConditional**(java.lang.String propertyKey, [IRPCollection](#) formalKey, [IRPCollection](#) actualValues) Returns the value of the specified property for the model element, taking into account the collection of tokens specified and the collection of token values specified.
java.lang.String	**getPropertyValueConditionalExplicit**(java.lang.String propertyKey, [IRPCollection](#) formalKey, [IRPCollection](#) actualValues) Returns the value of the specified property for the model element, if the default value was overridden, taking into account the collection of tokens specified and the collection of token values specified.
java.lang.String	**getPropertyValueExplicit**(java.lang.String propertyKey) Returns the value of the specified property for the model element if the default value was overridden.
[IRPCollection](#)	**getRedefines**() method getRedefines
[IRPCollection](#)	**getReferences**() Returns a collection of all the model elements that point to this model element.
[IRPCollection](#)	**getRemoteDependencies**() For Design Manager projects, returns a collection of all the dependencies that the model element has on remote elements.
java.lang.String	**getRemoteURI**() For elements that are remote resources, returns the URI of the resource.
int	**getRequirementTraceabilityHandle**() Returns the ID used by DOORS to refer to this requirement.
[IRPUnit](#)	**getSaveUnit**() Returns the unit that the model element is saved in.
[IRPStereotype](#)	**getStereotype**() **Deprecated.** *Since Rhapsody now allows multiple stereotypes to be applied to a model element, the getStereotypes() method should be used instead.*
[IRPCollection](#)	**getStereotypes**() Returns a collection of the stereotypes that have been applied to the element.
[IRPTag](#)	**getTag**(java.lang.String name) Returns the tag specified.
[IRPCollection](#)	**getTemplateParameters**() For model elements that are templates, returns the template parameters.
[IRPTemplateInstantiation](#)	

Method Summary

	getTi() 　　　For model elements that are template instantiations, returns an object that contains the template instantiation parameters.
java.lang.String	**getToolTipHTML**() 　　　Returns the HTML that would be used to display the tooltip for the element in the user interface.
java.lang.String	**getUserDefinedMetaClass**() 　　　Gets the name of the New Term on which the model element is based.
int	**hasNestedElements**() 　　　Checks whether the model element contains other elements.
int	**hasPanelWidget**() 　　　Checks whether the model element is bound to a panel diagram widget.
void	**highLightElement**() 　　　Locates the element in the Rhapsody browser, and highlights the element in the diagram where it appears.
int	**isATemplate**() 　　　Checks whether the model element is a template.
int	**isDescriptionRTF**() 　　　Checks whether the description for the element is in RTF format.
int	**isDisplayNameRTF**() 　　　Checks whether the label of the element is in RTF format.
int	**isModified**() 　　　Checks if the element was modified since the model was last saved.
int	**isRemote**() 　　　Checks whether the model element is a remote resource such as a DOORS/DNG requirement.
int	**locateInBrowser**() 　　　Locates the model element in the Rhapsody browser.
void	**lockOnDesignManager**() 　　　Locks the model element on the Design Manager server.
void	**openFeaturesDialog**(int newDialog) 　　　Displays the information for the element in the Features window.
void	**removeProperty**(java.lang.String propertyKey) 　　　Removes the value that was set for the specified property.
void	**removeRedefines**(IRPModelElement removedRedefine) 　　　method removeRedefines
void	**removeStereotype**(IRPStereotype stereotype) 　　　Removes the specified stereotype from the element.
void	**setDecorationStyle**(java.lang.String newVal)

Method Summary

	Used to specify the decoration style that should now be associated with the model element.
void	**setDescription**(java.lang.String description) Sets the specified string as the description of the element.
void	**setDescriptionAndHyperlinks**(java.lang.String rtfText, IRPCollection targets) Specifies an RTF string to use as the description for the element, and a collection of elements to which hyperlinks should be created.
void	**setDescriptionHTML**(java.lang.String descriptionHTML) Specifies the HTML text to use for the description of the model element.
void	**setDescriptionRTF**(java.lang.String descriptionRTF) Specifies the RTF string to use for the description of the model element.
void	**setDisplayName**(java.lang.String displayName) Specifies the text to use for the label of the model element.
void	**setDisplayNameRTF**(java.lang.String newVal) Specifies the RTF string to use for the label of the model element.
void	**setGUID**(java.lang.String gUID) Sets a new GUID for the model element.
void	**setIsShowDisplayName**(int isShowDisplayName) Specifies whether the label of the element should be displayed instead of the element name whenever the element is used in a diagram.
void	**setMainDiagram**(IRPDiagram mainDiagram) Specifies the "main" diagram for the element.
void	**setName**(java.lang.String name) Sets the specified string as the name of the element.
void	**setOfTemplate**(IRPModelElement ofTemplate) Makes the current model element a template instantiation of the specified template.
void	**setOwner**(IRPModelElement owner) Specifies the model element that should be the owner of this element.
void	**setPropertyValue**(java.lang.String propertyKey, java.lang.String propertyValue) Sets the value of a property for the model element.
void	**setRequirementTraceabilityHandle**(int requirementTraceabilityHandle) Sets a new ID to be used to reference this requirement
void	**setStereotype**(IRPStereotype stereotype) Applies the specified stereotype to the element.
IRPTag	**setTagContextValue**(IRPTag tag, IRPCollection elements, IRPCollection multiplicities) Applies the specified tag to the model element, and sets the value of the tag

Method Summary

	to a specific instance of another model element.
IRPTag	**setTagElementValue**(_IRPTag_ tag, _IRPModelElement_ val) Applies a tag whose type is a model element to the current element with the value specified.
IRPTag	**setTagValue**(_IRPTag_ tag, java.lang.String val) Applies the specified tag to the model element with the value specified.
void	**setTi**(_IRPTemplateInstantiation_ ti) For internal use only.
void	**synchronizeTemplateInstantiation**() After changes are made to a template, this method can be called on each instantiation of the template in order to update the instantiation to match the changes that were made to the template.
void	**unlockOnDesignManager**() Unlocks the model element on the Design Manager server.

Method Detail

addAssociation

IRPAssociationClass **addAssociation**(_IRPRelation_ end1,
                                 IRPRelation end2,
                                 java.lang.String name)

Creates an association class using the specified IRPRelation elements. Can only be called on the elements that can contain association classes - packages and classes. To add an ordinary association, use the method IRPClassifier.addRelationTo.

Parameters:
        end1 - the IRPRelation element at one end of the association
        end2 - the IRPRelation element at the second end of the association
        name - the name to use for the new association class

Returns:
        the association class that was created

---

addDependency

IRPDependency **addDependency**(java.lang.String dependsOnName,
                       java.lang.String dependsOnType)

Adds a dependency from the model element to the model element specified by the parameters. The method searches the model recursively until it finds an element that matches the name and metaclass specified. Since your model may contain multiple elements with the same name and type in different packages, the preferred way to add a dependency is to use the method addDependencyTo, which takes

a specific model element as an argument.

Parameters:

dependsOnName - the name of the model element on which this model element depends

dependsOnType - the type (metaclass) of the model element on which this model element depends. The strings to use for this parameter should be taken from the file metaclasses.txt in the Doc directory of the Rhapsody installation.

Returns:

the new dependency that was created

addDependencyBetween

IRPDependency **addDependencyBetween**(IRPModelElement dependent, IRPModelElement dependsOn)

Creates a dependency between the two specified elements. In most cases, you can use the method IRPModelElement.addDependencyTo to add a new dependency. However, in cases where you want to create a dependency between two read-only elements, you can use addDependencyBetween to create the new dependency and assign ownership of the dependency to a third model element.

Parameters:

dependent - the model element that is dependent on the other model element

dependsOn - the model element that the first element depends upon

Returns:

the new dependency that was created

Throws:

RhapsodyRuntimeException

addDependencyTo

IRPDependency **addDependencyTo**(IRPModelElement element)

Adds a dependency upon another model element.

Parameters:

element - the model element that this element depends upon

Returns:

the dependency created

addLinkToElement

IRPLink **addLinkToElement**(IRPModelElement toElement, IRPRelation assoc, IRPModelElement fromPort, IRPModelElement toPort)

Creates a link between this model element and the model element specified as an argument. The types of elements that can be connected with a link by using this method are the same types of elements that can be joined by a link in the Rational Rhapsody diagram editors. In addition to specifying the other model element that should be connected by this link, you must specify the association that the link should represent, or, alternatively, the two ports that should be used for the link. If you provide the

two ports as arguments, you should use Null for the association argument. Similarly, if you specify an association, you should use Null for the two port arguments. Note that if you are not specifying the two ports, you must provide an association as an argument even if there is only one relevant association.

Parameters:

> `toElement` - the model element that the link should connect to
>
> `assoc` - the association that the link should represent
>
> `fromPort` - the "from" port for the link
>
> `toPort` - the "to" port for the link

Returns:

> the link created

---

addNewAggr

[IRPModelElement](#) **addNewAggr**(java.lang.String metaType,
                     java.lang.String name)

Adds a new model element to the current element, for example, adding a class to a package.

Parameters:

> `metaType` - the type of element to add. The string to use is the name of the appropriate metaclass. The list of metaclass names that can be used for this argument can be found in the file metaclasses.txt in the Doc directory of your Rhapsody installation.
>
> `name` - the name to use for the new element

Returns:

> the new element that was created

```
static IRPApplication app = RhapsodyAppServer.getActiveRhapsodyApplication();
IRPProject prj = app.openProject("d:\\temp\\_sample_code\\Class_Tricks.rpy");
IRPPackage cameraPackage = prj.addPackage("Cameras");
cameraPackage.addNewAggr("Stereotype", "s1");
```

---

addProperty

void **addProperty**(java.lang.String propertyKey,
                java.lang.String propertyType,
                java.lang.String propertyValue)

Adds a new property to the model element and assigns a value to it. Note that this method does not have a user interface equivalent in the Features window.

Parameters:

> `propertyKey` - the name of the property to add. The syntax to use for this parameter is Subject.Metaclass.Property, for example, CG.Class.NewProperty
>
> `propertyType` - the property type. The strings that can be used for this parameter are: "Enum","Bool","String","Color","Int","Double","Font","File","Path", and "MultiLine". If you want to add a property of type Enum, you can specify the possible values using the following syntax for this parameter: "Enum,wood,plastic,metal".
>
> `propertyValue` - the value to assign to the new property. For boolean properties, use "True" or "False".

addRedefines

void **addRedefines**(<u>IRPModelElement</u> newRedefine)

> method addRedefines
> **Throws:**
> > <u>RhapsodyRuntimeException</u>

addRemoteDependencyTo

<u>IRPDependency</u> **addRemoteDependencyTo**(<u>IRPModelElement</u> element,
                                java.lang.String linkType)

> For Design Manager projects, used to create a dependency from a model element to a remote element. This method corresponds to the "link to remote requirement" option in the user interface. In order to have the remote element available as a model element to use with this method, you must first call the IRPProject.getRemoteResourcePackages() method. (For remote requirements that are not yet linked to any elements in the model, you must also call the IRPPackage.populateRemoteRequirements() method.) Note that while the first parameter can be any object of type IRPModelElement, at the moment you can only add dependencies to remote requirements.
> **Parameters:**
> > element - the remote element to which a dependency should be created
> > linkType - - one of the link types available with the requirement tool that you are using. For example, for Doors Next Generation, the possible types are "Derives From", "Refines", "Satisfies", and "Trace".
> **Returns:**
> > the new dependency that was created

addSpecificStereotype

void **addSpecificStereotype**(<u>IRPStereotype</u> stereotype)

> Applies the specified stereotype to the model element.
> **Parameters:**
> > stereotype - the stereotype to apply to the model element

addStereotype

<u>IRPStereotype</u> **addStereotype**(java.lang.String name,
                        java.lang.String metaType)

> Applies the specified stereotype to the model element if the project contains a stereotype with the name specified and applicable to the metaclass specified. If the project does not yet contain such a stereotype, this method creates the stereotype in the package that owns the model element, and applies the new stereotype to the model element.

Parameters:
> `name` - the name of the stereotype to apply (or create and apply)
> `metaType` - the metaclass that the stereotype is applicable to

Returns:
> the stereotype applied (or created and applied) to the model element

---

becomeTemplateInstantiationOf

void **becomeTemplateInstantiationOf**(<u>IRPModelElement</u> newVal)

> Makes the current model element a template instantiation of the specified template.
> **Parameters:**
> > `newVal` - the template to use for the instantiation

---

changeTo

<u>IRPModelElement</u> **changeTo**(java.lang.String metaClass)

> Changes the model element to the type of element specified by the parameter provided. This corresponds to the "Change to" option that is included in the pop-up menu for model elements in the browser. An element that is not a "new term" can be changed to any of the "new terms" that are based on it. An element that is a "new term" can be changed to the model element that it is based on or to any of the other "new terms" that are based on that base element. Note that when you use this method, you must always use a variable to store the model element that is returned. This is necessary because the original element is destroyed, so you will have problems if you try to access the original element after this method is called.
> **Parameters:**
> > `metaClass` - the metaclass of the element that this element should be changed to. The strings to use for this parameter should be taken from the file metaclasses.txt in the Doc directory of the Rhapsody installation.
> **Returns:**
> > the new model element that was created.

---

clone

<u>IRPModelElement</u> **clone**(java.lang.String name,
> > > > > > <u>IRPModelElement</u> newOwner)

> Clones a model element.
> **Parameters:**
> > `name` - the name to use for the new element
> > `newOwner` - the model element that should be the owner of the new element
> **Returns:**
> > the new model element that was created

---

createOSLCLink

void **createOSLCLink**(java.lang.String type,
                        java.lang.String purl)

> Creates an OSLC link between the element and the element represented by the specified URL.
> **Parameters:**
>> `type` - one of the OSLC link types that can be created. Must be one of the typed defined in IRPModelElement.OSLCLink.Types. Note that the "Derives From" link type can only be used when creating a link to a requirement from DNG.
>> `purl` - the URL for the target element
> **Throws:**
>> RhapsodyRuntimeException

---

deleteDependency

void **deleteDependency**(IRPDependency dependency)

> Deletes the specified dependency from the model.
> **Parameters:**
>> `dependency` - the dependency to be deleted

---

deleteFromProject

void **deleteFromProject**()

> Deletes the current model element from the model.

---

deleteOSLCLink

void **deleteOSLCLink**(java.lang.String type,
                        java.lang.String purl)

> Deletes the specified OSLC link from the model.
> **Parameters:**
>> `type` - the link type of the OSLC link that is to be deleted. Must be one of the typed defined in IRPModelElement.OSLCLink.Types. You can also use "*" to represent all of the types.
>> `purl` - the URL for the link's target element. You can use "*" as the value of the parameter in order to delete all links of the specified type.
> **Throws:**
>> RhapsodyRuntimeException

---

errorMessage

java.lang.String **errorMessage**()

Returns error message for last method called. If the last method completed successfully, then this method returns an empty string. To get the correct error message for a method, errorMessage() must be called immediately after the method is called.

Returns:
>   the error message for the last method called

---

findElementsByFullName

[IRPModelElement](#) **findElementsByFullName**(java.lang.String name,
                                    java.lang.String metaClass)

Searches for the specified model element in the specified path under the current model element.

```
// this code gets the class Webcam in the package SpecializedCameras which is a subpack
IRPProject currentProject = app.activeProject();
IRPClass classToFind = (IRPClass)currentProject.findElementsByFullName("Webcam in Camer
System.out.println(classToFind.getFullPathName());
```

Parameters:
>   name - the name of the element to search for and the relative path to the element starting at the current element. This argument can use the format "Class in Package::Subpackage" or the format "Package::Subpackage::Class", for example,
>   findElementsByFullName("Cameras::SpecializedCameras::Webcam", "Class")
>   metaClass - the metaclass of the element you are looking for. The strings to use for this parameter should be taken from the file metaclasses.txt in the Doc directory of the Rhapsody installation.

Returns:
>   the model element that was specified

---

findNestedElement

[IRPModelElement](#) **findNestedElement**(java.lang.String name,
                                java.lang.String metaClass)

Searches for the specified model element. This method only searches the first level of elements below the current element. To search all of the levels below the current element, use the method findNestedElementRecursive.

Parameters:
>   name - the name of the element to search for
>   metaClass - the metaclass of the element you are looking for. The strings to use for this parameter should be taken from the file metaclasses.txt in the Doc directory of the Rhapsody installation.

Returns:
>   the model element that was specified. Note that the element is always returned as an object of type IRPModelElement. So you will usually have to use casting, for example,
>   IRPPackage packageToUse = (IRPPackage)prj.findNestedElement("GreeterPackage", "Package");

---

findNestedElementRecursive

[IRPModelElement](#) **findNestedElementRecursive**(java.lang.String name,
                                           java.lang.String metaClass)

> Searches recursively for the specified model element. This method searches all of the levels below the current element. To search only the first level of elements below the current element, use the method findNestedElement.
>
> **Parameters:**
>> name - the name of the element to search for
>>
>> metaClass - the metaclass of the element you are looking for. The strings to use for this parameter should be taken from the file metaclasses.txt in the Doc directory of the Rhapsody installation.
>
> **Returns:**
>> the model element that was specified. Note that the element is always returned as an object of type IRPModelElement. So you will usually have to use casting, for example,
>> IRPPackage packageToUse =
>> (IRPPackage)prj.findNestedElementRecursive("GreeterPackage", "Package");

---

getAllTags

[IRPCollection](#) **getAllTags**()

> Returns a collection of all the element's tags.
> **Returns:**
>> a collection of IRPTag objects representing the element's tags

---

getAnnotations

[IRPCollection](#) **getAnnotations**()

> Returns all of the element's annotations. This includes comments, constraints, and requirements.
> **Returns:**
>> all of the element's annotations: comments, constraints, and requirements

---

getAssociationClasses

[IRPCollection](#) **getAssociationClasses**()

> Returns a collection of all the association classes directly beneath this model element. This method is only relevant for packages and classifiers.
> **Returns:**
>> all of the association classes directly beneath this model element

---

getBinaryID

byte[] **getBinaryID**()

Returns the GUID of the model element as an array of bytes, as opposed to the method getGUID, which returns the GUID as a string.
Returns:
the GUID of the model element as an array of bytes

getConstraints

IRPCollection **getConstraints**()

Returns all of the element's constraints.
Returns:
all of the element's constraints

getConstraintsByHim

IRPCollection **getConstraintsByHim**()

For internal use only.

getControlledFiles

IRPCollection **getControlledFiles**()

Returns a collection of all the element's controlled files.
Returns:
a collection of IRPControlledFile objects representing the element's controlled files

getDecorationStyle

java.lang.String **getDecorationStyle**()

Returns the name of the decoration style currently associated with the model element.
Returns:
the decoration style currently associated with the model element

getDependencies

IRPCollection **getDependencies**()

Returns all of the element's dependencies.
Returns:

all of the element's dependencies

getDescription

`java.lang.String` **`getDescription`**`()`

Returns the description defined for the element.
Returns:
the description for the element

getDescriptionHTML

`java.lang.String` **`getDescriptionHTML`**`()`

Returns HTML representation of the element description.
Returns:
HTML representation of the element description

getDescriptionPlainText

`java.lang.String` **`getDescriptionPlainText`**`()`

Returns the description defined for the element in plain text format.
Returns:
the description for the element in plain text format

getDescriptionRTF

`java.lang.String` **`getDescriptionRTF`**`()`

Returns the description defined for the element in RTF format.
Returns:
the description for the element in RTF format

getDisplayName

`java.lang.String` **`getDisplayName`**`()`

Returns the label of the model element.
Returns:
the label of the model element

getDisplayNameRTF

`java.lang.String` **`getDisplayNameRTF`**`()`

Returns the label of the model element as an RTF string.

Returns:

the label of the model element as an RTF string.

---

getErrorMessage

`java.lang.String` **`getErrorMessage`**`()`

Returns error message for last method called. If the last method completed successfully, then this method returns an empty string. To get the correct error message for a method, errorMessage() must be called immediately after the method is called.

Returns:

the error message for the last method called

---

getFullPathName

`java.lang.String` **`getFullPathName`**`()`

Returns the full path name of the model element. The format of the string returned is package::subpackage::class.

```
 // this code prints the full path name for each class in the Cameras package, including
      System.out.println("=============================================================
      IRPCollection allClassesInCamerasPackage = cameraPackage.getNestedElementsByMeta
      int numberOfClasses = allClassesInCamerasPackage.getCount();
      // note that when using getItem to get an item from an IRPCollection object, the
      IRPModelElement elementInCollection;
      for(int i = 1; i <numberOfClasses+1 ; i++) {
              elementInCollection = (IRPModelElement)allClassesInCamerasPackage.getItem
      System.out.println(elementInCollection.getFullPathName());
 }
```

Returns:

the full path name of the model element. The format of the string returned is package::subpackage::class.

---

getFullPathNameIn

`java.lang.String` **`getFullPathNameIn`**`()`

Retrieves the full path name of the element as a string in the following format: (class) in (package).

Returns:

the full path name of the element in the format: (class) in (package)

---

getGUID

`java.lang.String` **`getGUID`**`()`

Returns the GUID of the model element. In situations where you may have to carry out multiple searches for the same element, you can use the getGUID method to get the GUID of the element once, and then use the method IRPProject.findElementByGUID which performs a quicker search than the other "find" methods provided.

Returns:

the GUID of the model element

getHyperLinks

[IRPCollection](#) **`getHyperLinks`**`()`

Returns a collection of all the hyperlinks associated with the element.

Returns:

a collection of IRPHyperLink objects representing the hyperlinks associated with the element

getIconFileName

`java.lang.String` **`getIconFileName`**`()`

Returns the full path of the graphic file used to represent elements of this type in the browser, for example, D:\programs\rhapsody80\Share\PredefinedPictures\Icons\RhapsodyIcons_72.gif.

Returns:

the full path of the graphic file used to represent elements of this type in the browser

getInterfaceName

`java.lang.String` **`getInterfaceName`**`()`

Returns the name of the API interface corresponding to the current element, for example, IRPClass for a class element, IRPOperation for an operation element.

Returns:

the name of the API interface corresponding to the current element

getIsExternal

`int` **`getIsExternal`**`()`

Checks whether the element is an "external" element - corresponds to the value of the property UseAsExternal.

Returns:

1 if the element is an "external" element, 0 otherwise

getIsOfMetaClass

int **getIsOfMetaClass**(java.lang.String metaClass)

Indicates whether the model element is based on the metaclass provided as a parameter.

Parameters:

metaClass - The name of the metaclass to check for. The strings to use for this parameter should be taken from the file metaclasses.txt in the Doc directory of the Rhapsody installation.

Returns:

indication of whether the model element is based on the metaclass specified. 1 means that the model element is based on the metaclass specified, 0 means it is not based on that metaclass.

getIsShowDisplayName

int **getIsShowDisplayName**()

Checks whether the model element is configured to have its label displayed instead of its name whenever it is included in a diagram. This behavior is controlled by the General::Graphics::ShowLabels property.

Returns:

1 if the element is configured to have its label displayed instead of its name in diagrams, 0 otherwise

getIsUnresolved

int **getIsUnresolved**()

Checks if the element is an element that can't be resolved by Rhapsody.

Returns:

indication of whether the element is an unresolved element - 0 if the element can be resolved, 1 if the element is unresolved

getLocalTags

IRPCollection **getLocalTags**()

Returns a collection of the tags that were created locally for this model element.

Returns:

the tags that were created locally for the model element (collection of IRPTag elements)

getMainDiagram

IRPDiagram **getMainDiagram**()

Returns the "main" diagram for the element. This operation is valid only for packages, classes, actors, use cases, objects, and interfaces.

Returns:
> the "main" diagram for the element

---

getMetaClass

`java.lang.String` **`getMetaClass`**`()`

Gets the name of the metaclass on which the model element is based. Note that if the element is based on a New Term stereotype, the string returned here will be the metaclass on which it is based. To get the name of the New Term stereotype, use the method IRPModelElement.getUserDefinedMetaClass.

Returns:
> the name of the metaclasses on which the model element is based. The string returned will be one of the the metaclass names listed in the file metaclasses.txt in the Doc directory of the Rhapsody installation. For example, for an object of type IRPStereotype, the string "Stereotype" will be returned.

---

getName

`java.lang.String` **`getName`**`()`

Returns the name of the element.

Returns:
> the name of the element

---

getNestedElements

[IRPCollection](#) **`getNestedElements`**`()`

Gets a collection of all the model elements that are directly under the current element. Note that if you call this method on a package, the returned collection will not include functions, global variables, or global objects contained in the package because these are actually contained in a class called TopLevel. To get the functions, global variables, or global objects contained in a package, use the following IRPPackage methods: getGlobalFunctions(), getGlobalVariables(), and getGlobalObjects().

Returns:
> a collection of IRPModelElement objects representing all the model elements that are directly under the current element

```
IRPProject prj = app.openProject("l:\\temp\\_sample_code\\Unit_Tricks.rpy");
IRPPackage vehiclePackage = prj.addPackage("Vehicles");
vehiclePackage.addClass("Car");
vehiclePackage.addClass("Jeep");
vehiclePackage.addClass("Convertible");
prj.save();
IRPCollection elementsInVehiclesPackage = vehiclePackage.getNestedElements();
IRPModelElement elementInCollection;
System.out.println("The Vehicles package contains:");
for (int i = 1; i <= elementsInVehiclesPackage.getCount(); i++) {
        elementInCollection = (IRPModelElement)elementsInVehiclesPackage.getItem(i);
        System.out.println("\t" + elementInCollection.getName());
```

```
        }
```

getNestedElementsByMetaClass

IRPCollection **getNestedElementsByMetaClass**(java.lang.String metaClass,
                                        int recursive)

Retrieves all of the model elements of the specified type below the current element. The second argument can be used to specify whether the retrieval should be recursive.

```
  // this code retrieves all the classes in the Cameras package (including classes in su
                IRPCollection allClassesInCamerasPackage = cameraPackage.getNestedElement
                int numberOfClasses = allClassesInCamerasPackage.getCount();
                // note that when using getItem to get an item from an IRPCollection obj
                IRPModelElement elementInCollection;
                for(int i = 1; i <numberOfClasses+1 ; i++) {
                        elementInCollection = (IRPModelElement)allClassesInCamerasPackage
                        System.out.println(elementInCollection.getMetaClass() + ": " + e
                }
```

Parameters:
> metaClass - the type of elements that you want to retrieve. The strings to use for this parameter should be taken from the file metaclasses.txt in the Doc directory of the Rhapsody installation.
> recursive - Use 1 to specify that the retrieval should be recursive. Use 0 if you only want to retrieve the relevant elements from the first level below the current element.

Returns:
> a collection of the model elements of the specified type below the current element

getNestedElementsRecursive

IRPCollection **getNestedElementsRecursive**()

Returns a collection that consists of the current element and all of the model elements below it.

```
// this code retrieves all the items in the Cameras package, and prints their type and
IRPCollection allItemsInCameraPackage = cameraPackage.getNestedElementsRecursive();
int numberOfElements = allItemsInCameraPackage.getCount();
// note that when using getItem to get an item from an IRPCollection object, the index
IRPModelElement elementInCollection;
for(int i = 1; i <numberOfElements+1 ; i++) {
        elementInCollection = (IRPModelElement)allItemsInCameraPackage.getItem(i);
        System.out.println(elementInCollection.getMetaClass() + ": " + elementInCollecti
}
```

Returns:
> a collection consisting of the current element and all of the model elements below it

getNewTermStereotype

IRPStereotype **getNewTermStereotype**()

If a "new term" stereotype has been applied to the element, returns the stereotype.
Returns:
the "new term" stereotype that was applied to the element

getOSLCLinks

IRPCollection **getOSLCLinks**()

Returns a collection of all the element's OSLC links. Each item in the collection is a string that uses the following format: "Type=<<link type>>(newline)URL=<<linked item URL>>".
Returns:
all of the element's OSLC links
Throws:
RhapsodyRuntimeException

getOfTemplate

IRPModelElement **getOfTemplate**()

If the element is an instantiation of a template, this method returns the template that it instantiates.
Returns:
the template that this model element instantiates

getOverlayIconFileName

java.lang.String **getOverlayIconFileName**()

Returns the full path of the graphic file that is used as an overlay on this specific model element, on top of the regular icon that represent elements of this type in the browser.
Returns:
the full path of the graphic file that is used as an overlay on this specific model element, on top of the regular icon that represent elements of this type in the browser

getOverriddenProperties

IRPCollection **getOverriddenProperties**(int recursive)

Returns a collection of all the properties whose value was overridden for this model element. The collection consists of strings that use the format subject:metaclass:property:value.
Parameters:
recursive - use 1 to specify that the method should return all properties overridden for the element - from the level of the element itself all the way up to the project level, use 0 to

specify that the method should only return the properties that were overridden at the level of the element itself

Returns:

the properties whose value was overridden for this model element

---

getOverriddenPropertiesByPattern

[IRPCollection](#) **getOverriddenPropertiesByPattern**(java.lang.String pattern,
                                                int localyOverridenOnly,
                                                int withDefaultValues)

method getOverriddenPropertiesByPattern

Throws:

[RhapsodyRuntimeException](#)

---

getOwnedDependencies

[IRPCollection](#) **getOwnedDependencies**()

Returns all of the dependencies that are owned by the element.

Returns:

all of the dependencies that are owned by this element

Throws:

[RhapsodyRuntimeException](#)

---

getOwner

[IRPModelElement](#) **getOwner**()

Returns the model element that owns this model element.

Returns:

the model element that owns this model element

---

getProject

[IRPProject](#) **getProject**()

Returns the project that the current element belongs to.

Returns:

the project the current element belongs to

---

getPropertyValue

java.lang.String **getPropertyValue**(java.lang.String propertyKey)

Returns the value of the specified property for the model element.

Parameters:

> `propertyKey` - the property whose value should be returned. The syntax to use for this parameter is Subject.Metaclass.Property, for example, CG.Class.ActiveThreadName

Returns:

> the value of the specified property. If a value has not been set specifically for this element, the default value is returned (the value propagated from a higher level)

getPropertyValueConditional

```
java.lang.String getPropertyValueConditional(java.lang.String propertyKey,
                                              IRPCollection formalKey,
                                              IRPCollection actualValues)
```

Returns the value of the specified property for the model element, taking into account the collection of tokens specified and the collection of token values specified. For more information on using tokens in property values, see "Conditional Properties" in the Rational Rhapsody help.

Parameters:

> `propertyKey` - the property whose value should be returned. The syntax to use for this parameter is Subject.Metaclass.Property, for example, CG.Class.ActiveThreadName
>
> `formalKey` - the collection of tokens to take into account (collection of strings)
>
> `actualValues` - the collection of token values to take into account (collection of strings)

Returns:

> the value of the specified property, taking into account the tokens and token values specified. If a value has not been set specifically for this element, the default value is returned (the value propagated from a higher level)

getPropertyValueConditionalExplicit

```
java.lang.String getPropertyValueConditionalExplicit(java.lang.String propertyKey,
                                                      IRPCollection formalKey,
                                                      IRPCollection actualValues)
```

Returns the value of the specified property for the model element, if the default value was overridden, taking into account the collection of tokens specified and the collection of token values specified. For more information on using tokens in property values, see "Conditional Properties" in the Rational Rhapsody help. If a value has not been set explicitly for the model element, the method will not return the default value (like the getPropertyValueConditional method does). Rather, it will throw an exception.

Parameters:

> `propertyKey` - the property whose value should be returned. The syntax to use for this parameter is Subject.Metaclass.Property, for example, CG.Class.ActiveThreadName
>
> `formalKey` - the collection of tokens to take into account (collection of strings)
>
> `actualValues` - the collection of token values to take into account (collection of strings)

Returns:

> the value that was explicitly set for the model element for the specified property, taking into account the tokens and token values specified

getPropertyValueExplicit

java.lang.String **getPropertyValueExplicit**(java.lang.String propertyKey)

Returns the value of the specified property for the model element if the default value was overridden. If a value has not been set explicitly for the model element, it will not return the default value (like the getPropertyValue method does). Rather, it will throw an exception.

Parameters:
propertyKey - the property whose value should be returned. The syntax to use for this parameter is Subject.Metaclass.Property, for example, CG.Class.ActiveThreadName

Returns:
the value that was explicitly set for the model element for the specified property

getRedefines

IRPCollection **getRedefines**()

method getRedefines
Throws:
RhapsodyRuntimeException

getReferences

IRPCollection **getReferences**()

Returns a collection of all the model elements that point to this model element.
Returns:
all the model elements that point to this model element

getRemoteDependencies

IRPCollection **getRemoteDependencies**()

For Design Manager projects, returns a collection of all the dependencies that the model element has on remote elements.
Returns:
all the dependencies that the model element has on remote elements

getRemoteURI

java.lang.String **getRemoteURI**()

For elements that are remote resources, returns the URI of the resource.
Returns:
the URI of the remote resource. If the method is called for an element that is not a remote resource, an empty string is returned.

getRequirementTraceabilityHandle

int **getRequirementTraceabilityHandle**()

Returns the ID used by DOORS to refer to this requirement.
Returns:
the ID used by DOORS to refer to this requirement

getSaveUnit

IRPUnit **getSaveUnit**()

Returns the unit that the model element is saved in.
Returns:
the unit that the element is saved in

getStereotype

IRPStereotype **getStereotype**()

Deprecated. *Since Rhapsody now allows multiple stereotypes to be applied to a model element, the getStereotypes() method should be used instead.*

getStereotypes

IRPCollection **getStereotypes**()

Returns a collection of the stereotypes that have been applied to the element.
Returns:
the stereotypes that have been applied to the element

getTag

IRPTag **getTag**(java.lang.String name)

Returns the tag specified. This method can be used for both local tags and global tags.
Parameters:
name - the name of the tag to return
Returns:
the tag specified

getTemplateParameters

[IRPCollection](#) **getTemplateParameters**()

> For model elements that are templates, returns the template parameters.
> **Returns:**
> > the parameters of the template

---

getTi

[IRPTemplateInstantiation](#) **getTi**()

> For model elements that are template instantiations, returns an object that contains the template instantiation parameters.
> **Returns:**
> > object that contains the template instantiation parameters

---

getToolTipHTML

java.lang.String **getToolTipHTML**()

> Returns the HTML that would be used to display the tooltip for the element in the user interface.
> **Returns:**
> > the HTML that would be used to display the tooltip for the element in the user interface

---

getUserDefinedMetaClass

java.lang.String **getUserDefinedMetaClass**()

> Gets the name of the New Term on which the model element is based.
> **Returns:**
> > the name of the New Term on which the model element is based. The string returned will be the name of the New Term stereotype that you defined. To get the name of the metaclass on which the New Term is based, use the method IRPModelElement.getMetaClass().

---

hasNestedElements

int **hasNestedElements**()

> Checks whether the model element contains other elements.
> **Returns:**
> > 1 if the model element contains other elements, 0 otherwise

---

hasPanelWidget

int **hasPanelWidget**()

> Checks whether the model element is bound to a panel diagram widget.
> **Returns:**
> > 1 if the element is bound to a panel diagram widget, 0 otherwise

---

highLightElement

void **highLightElement**()

> Locates the element in the Rhapsody browser, and highlights the element in the diagram where it appears. Note that the element will be highlighted in the diagram only if it is the kind of element that can appear in only one diagram, for example, a state.

---

isATemplate

int **isATemplate**()

> Checks whether the model element is a template.
> **Returns:**
> > 1 if the element is a template, 0 otherwise

---

isDescriptionRTF

int **isDescriptionRTF**()

> Checks whether the description for the element is in RTF format.
> **Returns:**
> > 1 if the description is in RTF format, 0 otherwise

---

isDisplayNameRTF

int **isDisplayNameRTF**()

> Checks whether the label of the element is in RTF format.
> **Returns:**
> > 1 if the label is in RTF format, 0 otherwise

---

isModified

int **isModified**()

> Checks if the element was modified since the model was last saved.

Returns:
> 1 if the element was modified since the model was last saved, 0 if the element was not modified

---

isRemote

int **isRemote**()

> Checks whether the model element is a remote resource such as a DOORS/DNG requirement.
> **Returns:**
>> 1 if the element is a remote resource, 0 if not

---

locateInBrowser

int **locateInBrowser**()

> Locates the model element in the Rhapsody browser.
> **Returns:**
>> returns 1 if the element was located in the browser

---

lockOnDesignManager

void **lockOnDesignManager**()

> Locks the model element on the Design Manager server.

---

openFeaturesDialog

void **openFeaturesDialog**(int newDialog)

> Displays the information for the element in the Features window. Depending on the value of the parameter provided, opens a new Features window or uses an already-open Features window.
> **Parameters:**
>> newDialog - Use 1 to specify that the element information should be displayed in a new Features window. Use 0 to specify that the information should be displayed in a Features window that is already open or in a new window if there is no open Features window.

---

removeProperty

void **removeProperty**(java.lang.String propertyKey)

> Removes the value that was set for the specified property. This is equivalent to the "un-override" option in the Features window.
> **Parameters:**

propertyKey - the property whose value should be removed. The syntax to use for this parameter is Subject.Metaclass.Property, for example, CG.Class.ActiveThreadName

removeRedefines

void **removeRedefines**([IRPModelElement](#) removedRedefine)

method removeRedefines
Throws:
> [RhapsodyRuntimeException](#)

removeStereotype

void **removeStereotype**([IRPStereotype](#) stereotype)

Removes the specified stereotype from the element.
Parameters:
> stereotype - the stereotype to be removed from the element

setDecorationStyle

void **setDecorationStyle**(java.lang.String newVal)

Used to specify the decoration style that should now be associated with the model element.
Parameters:
> newVal - The decoration style that should now be associated with the model element. The value of the parameter must be one of the strings included in the value of the property Format::Decoration::StyleNames.

setDescription

void **setDescription**(java.lang.String description)

Sets the specified string as the description of the element.
Parameters:
> description - the string to use as the description of the element

setDescriptionAndHyperlinks

void **setDescriptionAndHyperlinks**(java.lang.String rtfText,
> [IRPCollection](#) targets)

Specifies an RTF string to use as the description for the element, and a collection of elements to which hyperlinks should be created.
Parameters:

rtfText - the string to use for the element description - must be in RTF format

targets - the collection of elements for which hyperlinks should be created

setDescriptionHTML

void **setDescriptionHTML**(java.lang.String descriptionHTML)

Specifies the HTML text to use for the description of the model element.
Parameters:
descriptionHTML - the HTML text to use for the description of the model element

setDescriptionRTF

void **setDescriptionRTF**(java.lang.String descriptionRTF)

Specifies the RTF string to use for the description of the model element.
Parameters:
descriptionRTF - the RTF string to use for the description of the model element

setDisplayName

void **setDisplayName**(java.lang.String displayName)

Specifies the text to use for the label of the model element.
Parameters:
displayName - the text to use for the label of the model element

setDisplayNameRTF

void **setDisplayNameRTF**(java.lang.String newVal)

Specifies the RTF string to use for the label of the model element.
Parameters:
newVal - the RTF string to use for the label of the model element

setGUID

void **setGUID**(java.lang.String gUID)

Sets a new GUID for the model element.
Parameters:
gUID - the new GUID that should be used for the model element

setIsShowDisplayName

void **setIsShowDisplayName**(int isShowDisplayName)

> Specifies whether the label of the element should be displayed instead of the element name whenever the element is used in a diagram. This method changes the value of the General::Graphics::ShowLabels property.
> **Parameters:**
> > isShowDisplayName - use 1 if you want the label of the element displayed, use 0 if you want the name of the element displayed

setMainDiagram

void **setMainDiagram**(IRPDiagram mainDiagram)

> Specifies the "main" diagram for the element. This operation is valid only for packages, classes, actors, use cases, objects, and interfaces.
> **Parameters:**
> > mainDiagram - the diagram to use as the "main" diagram for the element

setName

void **setName**(java.lang.String name)

> Sets the specified string as the name of the element.
> **Parameters:**
> > name - the string to use as the name of the element

setOfTemplate

void **setOfTemplate**(IRPModelElement ofTemplate)

> Makes the current model element a template instantiation of the specified template.
> **Parameters:**
> > ofTemplate - the template to use for the instantiation

setOwner

void **setOwner**(IRPModelElement owner)

> Specifies the model element that should be the owner of this element.
> **Parameters:**
> > owner - the model element that should be the owner of this element

setPropertyValue

```
void setPropertyValue(java.lang.String propertyKey,
                      java.lang.String propertyValue)
```

Sets the value of a property for the model element.
Parameters:
> `propertyKey` - the property whose value should be set. The syntax to use for this parameter is Subject.Metaclass.Property, for example, CG.Class.ActiveThreadName.
> `propertyValue` - the new value to use for the property. For boolean properties, use "True" or "False".

setRequirementTraceabilityHandle

```
void setRequirementTraceabilityHandle(int requirementTraceabilityHandle)
```

Sets a new ID to be used to reference this requirement
Parameters:
> `requirementTraceabilityHandle` - the new ID that should be used to reference this requirement

setStereotype

```
void setStereotype(IRPStereotype stereotype)
```

Applies the specified stereotype to the element.
Parameters:
> `stereotype` - the stereotype to be applied to the element

setTagContextValue

```
IRPTag setTagContextValue(IRPTag tag,
                          IRPCollection elements,
                          IRPCollection multiplicities)
```

Applies the specified tag to the model element, and sets the value of the tag to a specific instance of another model element.
Parameters:
> `tag` - the tag to apply to the model element
> `elements` - collection of model elements representing the full path to the element. This collection is used to set the value of the tag to the full path of the target element. The collection must consist of objects of type IRPModelElement.
> `multiplicities` - collection of the relevant indices for each of the model elements in the first collection (the "elements" parameter). This makes it possible to point to a specific instance of the target model element when multiplicity is greater than one. The collection must consist of integers provided as strings.

Returns:
> the tag created for the model element

setTagElementValue

IRPTag **setTagElementValue**(IRPTag tag,
                            IRPModelElement val)

Applies a tag whose type is a model element to the current element with the value specified. If the tag has already been applied to the current element, the method can be used to modify the value of the tag.

Parameters:
       tag - the tag to apply to the element
       val - the value to use for the tag applied

Returns:
       the tag created for the model element

setTagValue

IRPTag **setTagValue**(IRPTag tag,
                       java.lang.String val)

Applies the specified tag to the model element with the value specified. If the tag has already been applied to the model element, the method can be used to modify the value of the tag.

Parameters:
       tag - the tag to apply to the element
       val - the value to use for the tag applied

Returns:
       the tag created for the model element

setTi

void **setTi**(IRPTemplateInstantiation ti)

For internal use only.

synchronizeTemplateInstantiation

void **synchronizeTemplateInstantiation**()

After changes are made to a template, this method can be called on each instantiation of the template in order to update the instantiation to match the changes that were made to the template.

unlockOnDesignManager

void **unlockOnDesignManager**()

setTagContextValue   399

Unlocks the model element on the Design Manager server.

com.telelogic.rhapsody.core
Class IRPModelElement.OSLCLink

```
java.lang.Object
   └─com.telelogic.rhapsody.core.IRPModelElement.OSLCLink
```

Enclosing interface:
> IRPModelElement

---

public static final class **IRPModelElement.OSLCLink**
extends java.lang.Object

Constant values used with elements of this type

Nested Class Summary

static class	**IRPModelElement.OSLCLink.Types** This class contains values that specify OSLC Types

Constructor Summary

IRPModelElement.OSLCLink()

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

IRPModelElement.OSLCLink

public **IRPModelElement.OSLCLink**()

---

com.telelogic.rhapsody.core
Class IRPModelElement.OSLCLink.Types

```
java.lang.Object
  └─ com.telelogic.rhapsody.core.IRPModelElement.OSLCLink.Types
```

Enclosing class:
> IRPModelElement.OSLCLink

---

public static final class **IRPModelElement.OSLCLink.Types**
extends java.lang.Object

This class contains values that specify OSLC Types

Field Summary

static java.lang.String	**DERIVES** OSLC link type: Derives
static java.lang.String	**EXTERNAL** OSLC link type: External
static java.lang.String	**REFINE** OSLC link type: Refine
static java.lang.String	**SATISFY** OSLC Link Type: Satisfy
static java.lang.String	**TRACE** OSLC Link Type: Trace

Constructor Summary

IRPModelElement.OSLCLink.Types()	

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

EXTERNAL

`public static final java.lang.String` **`EXTERNAL`**

OSLC link type: External

See Also:
[Constant Field Values](#)

REFINE

`public static final java.lang.String` **`REFINE`**

OSLC link type: Refine

See Also:
[Constant Field Values](#)

DERIVES

`public static final java.lang.String` **`DERIVES`**

OSLC link type: Derives

See Also:
[Constant Field Values](#)

SATISFY

`public static final java.lang.String` **`SATISFY`**

OSLC Link Type: Satisfy

See Also:
[Constant Field Values](#)

TRACE

`public static final java.lang.String` **`TRACE`**

OSLC Link Type: Trace

See Also:
[Constant Field Values](#)

Constructor Detail

IRPModelElement.OSLCLink.Types

```
public IRPModelElement.OSLCLink.Types()
```

| **Package** | **Class** | **Use** | **Tree** | **Serialized** | **Deprecated** | **Index** | **Help** |

PREV CLASS   **NEXT CLASS**   **FRAMES**   **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPModule

All Superinterfaces:
> IRPInstance, IRPModelElement, IRPRelation, IRPUnit

---

public interface **IRPModule**
extends IRPInstance

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

Methods inherited from interface com.telelogic.rhapsody.core.IRPInstance
addRelationToTheWhole, getAttributeValue, getInLinks, getInstantiatedBy, getListOfInitializerArguments, getOutLinks, setAttributeValue, setExplicit, setImplicit, setInitializerArgumentValue, setInstantiatedBy

Methods inherited from interface com.telelogic.rhapsody.core.IRPRelation
addQualifier, getAssociationClass, getInverse, getIsNavigable, getIsSymmetric, getMultiplicity, getObjectAsObjectType, getOfClass, getOtherClass, getQualifier, getQualifiers, getQualifierType, getRelationLabel, getRelationLinkName, getRelationRoleName, getRelationType, getVisibility, isTypelessObject, makeUnidirect, removeQualifier, setInverse, setIsNavigable, setMultiplicity, setOfClass, setOtherClass, setQualifier, setQualifierType, setRelationLabel, setRelationLinkName, setRelationRoleName, setRelationType

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit
copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

---

com.telelogic.rhapsody.core
Interface IRPNode

All Superinterfaces:
> IRPClassifier, IRPModelElement, IRPUnit

```
public interface IRPNode
```
extends IRPClassifier

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPComponentInstance	**addComponentInstance**(java.lang.String name) method addComponentInstance
void	**deleteComponentInstance**(java.lang.String name) method deleteComponentInstance
IRPComponentInstance	**findComponentInstance**(java.lang.String name) method findComponentInstance
IRPCollection	**getComponentInstances**() get property componentInstances
java.lang.String	**getCPUtype**() get property CPUtype
void	**setCPUtype**(java.lang.String cPUtype) set property CPUtype

Methods inherited from interface com.telelogic.rhapsody.core.IRPClassifier
addActivityDiagram, addAttribute, addFlowItems, addFlows, addGeneralization, addOperation, addRelation, addRelationTo, addStatechart, addUnidirectionalRelation, addUnidirectionalRelationTo, deleteAttribute, deleteFlowItems, deleteFlows, deleteGeneralization, deleteOperation, deleteRelation, findAttribute, findBaseClassifier, findDerivedClassifier, findGeneralization, findInterfaceItem, findNestedClassifier, findNestedClassifierRecursive, findRelation, findTrigger,

Methods inherited from interface com.telelogic.rhapsody.core.**IRPClassifier**
getActivityDiagram, getAttributes, getAttributesIncludingBases, getBaseClassifiers, getBehavioralDiagrams, getDerivedClassifiers, getFlowItems, getFlows, getGeneralizations, getInterfaceItems, getInterfaceItemsIncludingBases, getLinks, getNestedClassifiers, getOperations, getPorts, getRelations, getRelationsIncludingBases, getSequenceDiagrams, getSourceArtifacts, getStatechart

Methods inherited from interface com.telelogic.rhapsody.core.**IRPUnit**
copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addComponentInstance

IRPComponentInstance **addComponentInstance**(java.lang.String name)

method addComponentInstance
Throws:
RhapsodyRuntimeException

deleteComponentInstance

void **deleteComponentInstance**(java.lang.String name)

method deleteComponentInstance
Throws:
RhapsodyRuntimeException

findComponentInstance

IRPComponentInstance **findComponentInstance**(java.lang.String name)

method findComponentInstance
Throws:
RhapsodyRuntimeException

getCPUtype

java.lang.String **getCPUtype**()

get property CPUtype
Throws:
RhapsodyRuntimeException

getComponentInstances

IRPCollection **getComponentInstances**()

get property componentInstances
Throws:
RhapsodyRuntimeException

setCPUtype

void **setCPUtype**(java.lang.String cPUtype)

set property CPUtype
Throws:
RhapsodyRuntimeException

com.telelogic.rhapsody.core

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPObjectModelDiagram

All Superinterfaces:
> IRPDiagram, IRPModelElement, IRPUnit

---

public interface **IRPObjectModelDiagram**
extends IRPDiagram

The IRPObjectModelDiagram interface represents object model diagrams in Rational Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement

IRPModelElement.OSLCLink

Method Summary

Methods inherited from interface com.telelogic.rhapsody.core.IRPDiagram

addFreeShapeByType, addImage, addNewEdgeByType, addNewEdgeForElement, addNewNodeByType, addNewNodeForElement, addTextBox, closeDiagram, completeRelations, getCorrespondingGraphicElements, getElementsInDiagram, getGraphicalElements, getLastVisualizationModifiedTime, getPicture, getPictureAs, getPictureAsDividedMetafiles, getPictureEx, getPicturesWithImageMap, isOpen, isShowDiagramFrame, openDiagram, openDiagramView, populateDiagram, removeGraphElements, setShowDiagramFrame

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype,

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

---

com.telelogic.rhapsody.core
Interface IRPObjectNode

All Superinterfaces:
> IRPModelElement, IRPState, IRPStateVertex

---

public interface **IRPObjectNode**
extends IRPState

The IRPObjectNode interface represents Object Node elements in activity diagrams.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

void	**addInState**(IRPModelElement val) Adds the specified state to the list of "In State" states for the object node.
java.lang.String	**getInState**() **Deprecated.** *Use getInStateList() instead.*
IRPCollection	**getInStateList**() Returns a collection of the "In State" states for the object node.
IRPModelElement	**getRepresents**() Returns the class/type that this object node represents.
void	**removeInState**(IRPModelElement val) Removes the specified state from the list of "In State" states for the object node.
void	**setInState**(java.lang.String inState) **Deprecated.** *Use addInState instead.*
void	**setRepresents**(IRPModelElement represents) Specifies the class/type that this object node should represent.

Methods inherited from interface com.telelogic.rhapsody.core.IRPState

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPState

addActivityFinal, addConnector, addInternalTransition, addState, addStaticReaction, addTerminationState, createDefaultTransition, createNestedStatechart, deleteConnector, deleteInternalTransition, deleteStaticReaction, getDefaultTransition, getEntryAction, getExitAction, getFullNameInStatechart, getInheritsFrom, getInternalTransitions, getIsOverridden, getIsReferenceActivity, getItsStatechart, getItsSwimlane, getLogicalStates, getNestedStatechart, getReferenceToActivity, getSendAction, getStateType, getStaticReactions, getSubStates, getSubStateVertices, getTheEntryAction, getTheExitAction, isAnd, isCompound, isLeaf, isRoot, isSendActionState, overrideInheritance, resetEntryActionInheritance, resetExitActionInheritance, setEntryAction, setExitAction, setInternalTransition, setItsSwimlane, setReferenceToActivity, setStateType, setStaticReaction, unoverrideInheritance

Methods inherited from interface com.telelogic.rhapsody.core.IRPStateVertex

addFlow, addTransition, deleteTransition, getInTransitions, getOutTransitions, getParent, setParent

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addInState

void **addInState**(<u>IRPModelElement</u> val)

> Adds the specified state to the list of "In State" states for the object node.
> **Parameters:**
>> val - the state to add to the list of "In State" states.

getInState

java.lang.String **getInState**()

> **Deprecated.** *Use getInStateList() instead.*

getInStateList

<u>IRPCollection</u> **getInStateList**()

> Returns a collection of the "In State" states for the object node.
> **Returns:**
>> the "In State" states defined for the object node

getRepresents

<u>IRPModelElement</u> **getRepresents**()

> Returns the class/type that this object node represents.
> **Returns:**
>> the class/type that this object node represents

removeInState

void **removeInState**(<u>IRPModelElement</u> val)

> Removes the specified state from the list of "In State" states for the object node.
> **Parameters:**
>> val - the state to remove from the list

setInState

void **setInState**(java.lang.String inState)

> **Deprecated.** *Use addInState instead.*

setRepresents

void **setRepresents**([IRPModelElement](#) represents)

> Specifies the class/type that this object node should represent.
> **Parameters:**
>> represents - the class/type that this object node should represent

---

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

com.telelogic.rhapsody.core
Interface IRPOperation

All Superinterfaces:
 IRPClassifier, IRPInterfaceItem, IRPModelElement, IRPUnit

---

public interface **IRPOperation**
extends IRPInterfaceItem

The IRPOperation interface represents operations of classes in Rational Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

void	**createAutoFlowChart**() Creates a flowchart for the operation.
void	**deleteArgument**(IRPArgument argument) Deletes a specific argument from the operation.
void	**deleteFlowchart**() Deletes the flowchart or activity defined for the operation.
java.lang.String	**getBody**() Returns the body of the operation.
IRPFlowchart	**getFlowchart**() Returns the flowchart or activity defined for the operation.
java.lang.String	**getImplementationSignature**() Returns the signature of the operation as it will appear in the generated code.
java.lang.String	**getInitializer**() For constructors, gets the initializer code that was defined for the operation.
int	**getIsAbstract**() Checks whether the operation was defined as abstract.
int	

Method Summary

	getIsCgDerived() Checks whether the operation is an operation that is automatically generated by Rhapsody.
int	**getIsConst**() For operations in C++ classes, checks whether the operation was defined as a constant member function.
int	**getIsCtor**() Checks whether the operation is a constructor.
int	**getIsDtor**() Checks whether the operation is a destructor.
int	**getIsFinal**() For operations in Java classes, checks whether the operation was defined as final.
int	**getIsInline**() Checks whether the code for the operation will be generated inline.
int	**getIsStatic**() Checks whether the operation was defined as static.
int	**getIsTrigger**() Checks whether the operation was defined as a triggered operation.
int	**getIsVirtual**() For operations in C++ or C# classes, checks whether the operation was defined as virtual.
[IRPClassifier](#)	**getReturns**() Gets the return type of the operation.
java.lang.String	**getReturnTypeDeclaration**() If an on-the-fly type is used as the return type of an operation, this method returns the declaration for the type.
java.lang.String	**getVisibility**() Gets the visibility specified for the operation.
void	**setBody**(java.lang.String body) Sets the body of an operation.
void	**setFlowchart**([IRPFlowchart](#) flowchart) Specifies a flowchart or activity for the operation.
void	**setInitializer**(java.lang.String initializer) For constructors, used to specify code for the initializer of the operation.
void	**setIsAbstract**(int isAbstract) Specifies whether an operation should be defined as abstract.
void	**setIsConst**(int isConst) For operations in C++ classes, used to specify whether an operation should be defined as a constant member function.

Method Summary

void	**setIsFinal**(int isFinal) For operations in Java classes, used to specify whether an operation should be defined as final.
void	**setIsStatic**(int isStatic) Specifies whether an operation should be defined as static.
void	**setIsVirtual**(int isVirtual) For operations in C++ or C# classes, used to specify whether an operation should be defined as virtual.
void	**setReturns**(IRPClassifier returns) Specifies the return type of the operation.
void	**setReturnTypeDeclaration**(java.lang.String newVal) Creates an on-the-fly type to use as the return type of the operation, using the declaration that you provide as a parameter.
void	**setVisibility**(java.lang.String visibility) set property visibility

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPInterfaceItem

addArgument, addArgumentBeforePosition, getArguments, getSignature, getSignatureNoArgNames, getSignatureNoArgTypes, matchOnSignature

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPClassifier

addActivityDiagram, addAttribute, addFlowItems, addFlows, addGeneralization, addOperation, addRelation, addRelationTo, addStatechart, addUnidirectionalRelation, addUnidirectionalRelationTo, deleteAttribute, deleteFlowItems, deleteFlows, deleteGeneralization, deleteOperation, deleteRelation, findAttribute, findBaseClassifier, findDerivedClassifier, findGeneralization, findInterfaceItem, findNestedClassifier, findNestedClassifierRecursive, findRelation, findTrigger, getActivityDiagram, getAttributes, getAttributesIncludingBases, getBaseClassifiers, getBehavioralDiagrams, getDerivedClassifiers, getFlowItems, getFlows, getGeneralizations, getInterfaceItems, getInterfaceItemsIncludingBases, getLinks, getNestedClassifiers, getOperations, getPorts, getRelations, getRelationsIncludingBases, getSequenceDiagrams, getSourceArtifacts, getStatechart

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink,

---

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

createAutoFlowChart

void **createAutoFlowChart**()

Creates a flowchart for the operation.

---

deleteArgument

void **deleteArgument**(IRPArgument argument)

Deletes a specific argument from the operation.
Parameters:
    argument - The argument that should be deleted

---

deleteFlowchart

void **deleteFlowchart**()

Deletes the flowchart or activity defined for the operation.

getBody

`java.lang.String` **`getBody`**`()`

>
> Returns the body of the operation.
> **Returns:**
> > the body of the operation

getFlowchart

[IRPFlowchart](#) **`getFlowchart`**`()`

>
> Returns the flowchart or activity defined for the operation.
> **Returns:**
> > the flowchart or activity defined for the operation

getImplementationSignature

`java.lang.String` **`getImplementationSignature`**`()`

>
> Returns the signature of the operation as it will appear in the generated code.
> **Returns:**
> > the signature of the operation as it will appear in the generated code

getInitializer

`java.lang.String` **`getInitializer`**`()`

>
> For constructors, gets the initializer code that was defined for the operation.
> **Returns:**
> > the initializer code that was defined for the operation

getIsAbstract

`int` **`getIsAbstract`**`()`

>
> Checks whether the operation was defined as abstract.
> **Returns:**
> > 1 if the operation was defined as abstract, 0 otherwise

getIsCgDerived

int **getIsCgDerived**()

> Checks whether the operation is an operation that is automatically generated by Rhapsody.
> **Returns:**
> > 1 if the operation is automatically generated by Rhapsody, 0 otherwise

getIsConst

int **getIsConst**()

> For operations in C++ classes, checks whether the operation was defined as a constant member function.
> **Returns:**
> > 1 if the operation was defined as a constant member function, 0 otherwise

getIsCtor

int **getIsCtor**()

> Checks whether the operation is a constructor.
> **Returns:**
> > 1 if the operation is a constructor, 0 otherwise

getIsDtor

int **getIsDtor**()

> Checks whether the operation is a destructor.
> **Returns:**
> > 1 if the operation is a destructor, 0 otherwise

getIsFinal

int **getIsFinal**()

> For operations in Java classes, checks whether the operation was defined as final.
> **Returns:**
> > 1 if the operation was defined as final, 0 otherwise

getIsInline

int **getIsInline**()

Checks whether the code for the operation will be generated inline.
Returns:
> 1 if the code for the operation will be generated inline, 0 otherwise

getIsStatic

int **getIsStatic**()

Checks whether the operation was defined as static.
Returns:
> 1 if the operation was defined as static, 0 otherwise

getIsTrigger

int **getIsTrigger**()

Checks whether the operation was defined as a triggered operation.
Returns:
> 1 if the operation is a triggered operation, 0 otherwise

getIsVirtual

int **getIsVirtual**()

For operations in C++ or C# classes, checks whether the operation was defined as virtual.
Returns:
> 1 if the operation was defined as virtual, 0 otherwise

getReturnTypeDeclaration

java.lang.String **getReturnTypeDeclaration**()

If an on-the-fly type is used as the return type of an operation, this method returns the declaration for the type.
Returns:
> the declaration for the return type

getReturns

IRPClassifier **getReturns**()

Gets the return type of the operation.
Returns:
> the return type of the operation

getVisibility

`java.lang.String` **`getVisibility`**`()`

> Gets the visibility specified for the operation.
>
> **Returns:**
>> the visibility specified for the operation

---

setBody

`void` **`setBody`**`(java.lang.String body)`

> Sets the body of an operation.
>
> **Parameters:**
>> `body` - The code to use for the body of the operation. Use \n to represent the line breaks, for example, `takePicture.setBody("openShutter();\ncloseShutter();");`

---

setFlowchart

`void` **`setFlowchart`**`(`[IRPFlowchart](#)` flowchart)`

> Specifies a flowchart or activity for the operation.
>
> **Parameters:**
>> `flowchart` - the flowchart or activity to use for the operation

---

setInitializer

`void` **`setInitializer`**`(java.lang.String initializer)`

> For constructors, used to specify code for the initializer of the operation.
>
> **Parameters:**
>> `initializer` - The code to use for the initializer of the operation

---

setIsAbstract

`void` **`setIsAbstract`**`(int isAbstract)`

> Specifies whether an operation should be defined as abstract.
>
> **Parameters:**
>> `isAbstract` - Use 1 to specify that the operation should be defined as abstract. Use 0 to specify that the operation should not be defined as abstract.

---

setIsConst

void **setIsConst**(int isConst)

For operations in C++ classes, used to specify whether an operation should be defined as a constant member function.
Parameters:
isConst - Use 1 to specify that the operation should be defined as a constant member function. Use 0 to specify that the operation should not be defined as a constant member function.

setIsFinal

void **setIsFinal**(int isFinal)

For operations in Java classes, used to specify whether an operation should be defined as final.
Parameters:
isFinal - Use 1 to specify that the operation should be defined as final. Use 0 to specify that the operation should not be defined as final.

setIsStatic

void **setIsStatic**(int isStatic)

Specifies whether an operation should be defined as static.
Parameters:
isStatic - Use 1 to specify that the operation should be defined as static. Use 0 to specify that the operation should not be defined as static.

setIsVirtual

void **setIsVirtual**(int isVirtual)

For operations in C++ or C# classes, used to specify whether an operation should be defined as virtual.
Parameters:
isVirtual - Use 1 to specify that the operation should be defined as virtual. Use 0 to specify that the operation should not be defined as virtual.

setReturnTypeDeclaration

void **setReturnTypeDeclaration**(java.lang.String newVal)

Creates an on-the-fly type to use as the return type of the operation, using the declaration that you provide as a parameter.
Parameters:

newVal - The declaration to use for the on-the-fly type that is to be created to use as the return type of the operation

setReturns

void **setReturns**([IRPClassifier](#) returns)

Specifies the return type of the operation.
Parameters:
returns - the return type to use for the operation

setVisibility

void **setVisibility**(java.lang.String visibility)

set property visibility
Throws:
[RhapsodyRuntimeException](#)

 com.telelogic.rhapsody.core
Interface IRPowListListener

public interface **IRPowListListener**

Method Summary

void	**dblClickNotify**(int nRow, int nCol, java.lang.String sContent) method DblClickNotify
java.lang.String	**getInterfaceName**() get property interfaceName
void	**setObjID**(java.lang.String bstrObjID) method SetObjID

Method Detail

dblClickNotify

```
void dblClickNotify(int nRow,
                    int nCol,
                    java.lang.String sContent)
```

method DblClickNotify
Throws:
RhapsodyRuntimeException

setObjID

```
void setObjID(java.lang.String bstrObjID)
```

method SetObjID
Throws:
RhapsodyRuntimeException

setVisibility 428

getInterfaceName

`java.lang.String` **`getInterfaceName`**`()`

get property interfaceName

Throws:

> RhapsodyRuntimeException

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | <u>METHOD</u>   DETAIL: FIELD | CONSTR | <u>METHOD</u>

com.telelogic.rhapsody.core
Interface IRPowPaneMgr

```
public interface IRPowPaneMgr
```

Method Summary

void	**<u>addTabNotify</u>**(int nType, int nSubType, java.lang.String sObjID, java.lang.String sTitle) method AddTabNotify
void	**<u>closeTabNotify</u>**(java.lang.String sObjID) method CloseTabNotify
java.lang.String	**<u>getInterfaceName</u>**() get property interfaceName
<u>IRPowListListener</u>	**<u>getOWListListener</u>**(java.lang.String sObjID) get list listener
<u>IRPowTextListener</u>	**<u>getOWTextListener</u>**(java.lang.String sObjID) get text listener

Method Detail

addTabNotify

```
void addTabNotify(int nType,
                  int nSubType,
                  java.lang.String sObjID,
                  java.lang.String sTitle)
```

     method AddTabNotify
     Throws:
            <u>RhapsodyRuntimeException</u>

closeTabNotify

void **closeTabNotify**(java.lang.String sObjID)

> method CloseTabNotify
> **Throws:**
>> RhapsodyRuntimeException

---

getInterfaceName

java.lang.String **getInterfaceName**()

> get property interfaceName
> **Throws:**
>> RhapsodyRuntimeException

---

getOWListListener

IRPowListListener **getOWListListener**(java.lang.String sObjID)

> get list listener
> **Throws:**
>> RhapsodyRuntimeException

---

getOWTextListener

IRPowTextListener **getOWTextListener**(java.lang.String sObjID)

> get text listener
> **Throws:**
>> RhapsodyRuntimeException

---

com.telelogic.rhapsody.core
Interface IRPowTextListener

public interface **IRPowTextListener**

Method Summary

void	**dblClickNotify**(int nLine, java.lang.String szLine) method DblClickNotify
java.lang.String	**getInterfaceName**() get property interfaceName
void	**setObjID**(java.lang.String bstrObjID) method SetObjID

Method Detail

dblClickNotify

void **dblClickNotify**(int nLine,
                  java.lang.String szLine)

> method DblClickNotify
> **Throws:**
> > RhapsodyRuntimeException

setObjID

void **setObjID**(java.lang.String bstrObjID)

> method SetObjID
> **Throws:**
> > RhapsodyRuntimeException

getInterfaceName

java.lang.String **getInterfaceName**()

> get property interfaceName
> **Throws:**
> > [RhapsodyRuntimeException](#)

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

com.telelogic.rhapsody.core
Interface IRPPackage

All Superinterfaces:
  IRPModelElement, IRPUnit

All Known Subinterfaces:
  IRPProfile, IRPProject

```
public interface IRPPackage
```
extends IRPUnit

The IRPPackage interface represents packages in Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPFlowchart	**addActivityDiagram**() Adds a new activity diagram to the package.
IRPActor	**addActor**(java.lang.String name) Adds a new actor to the package.
IRPClass	**addClass**(java.lang.String name) Adds a new class to the package.
IRPCollaborationDiagram	**addCollaborationDiagram**(java.lang.String name) Adds a new collaboration diagram to the package.
IRPComponentDiagram	**addComponentDiagram**(java.lang.String name) Adds a new component diagram to the package.
IRPDeploymentDiagram	**addDeploymentDiagram**(java.lang.String name) Adds a new deployment diagram to the package.
IRPEvent	**addEvent**(java.lang.String name) Adds a new event to the package.

Method Summary

[IRPFlowItem](#)	**addFlowItems**(java.lang.String name) Adds an item flow to the package.
[IRPFlow](#)	**addFlows**(java.lang.String name) Adds a flow to the package.
[IRPOperation](#)	**addGlobalFunction**(java.lang.String name) Adds a global function to the package.
[IRPRelation](#)	**addGlobalObject**(java.lang.String name, java.lang.String otherClassName, java.lang.String otherClassPackageName) Adds an Object to the package.
[IRPAttribute](#)	**addGlobalVariable**(java.lang.String name) Adds a global variable to the package.
[IRPRelation](#)	**addImplicitObject**(java.lang.String name) Adds an implicit object to the package.
[IRPLink](#)	**addLink**([IRPInstance](#) fromPart, [IRPInstance](#) toPart, [IRPRelation](#) assoc, [IRPPort](#) fromPort, [IRPPort](#) toPort) Creates a link between two objects in the package.
[IRPLink](#)	**addLinkBetweenSYSMLPorts**([IRPInstance](#) fromPart, [IRPInstance](#) toPart, [IRPRelation](#) assoc, [IRPSysMLPort](#) fromPort, [IRPSysMLPort](#) toPort) Creates a link between two objects.
[IRPModule](#)	**addModule**(java.lang.String name) Adds a new File element to the package.
[IRPPackage](#)	**addNestedPackage**(java.lang.String name) Adds a nested package to the package.
[IRPNode](#)	**addNode**(java.lang.String name) Adds a Node element to the package.
[IRPObjectModelDiagram](#)	**addObjectModelDiagram**(java.lang.String name) Adds a new object model diagram to the package.
[IRPPanelDiagram](#)	**addPanelDiagram**(java.lang.String name) Adds a new panel diagram to the package.
[IRPSequenceDiagram](#)	**addSequenceDiagram**(java.lang.String name) Adds a new sequence diagram to the package.
[IRPStatechart](#)	**addStatechart**() Adds a new statechart to the package.
[IRPTimingDiagram](#)	**addTimingDiagram**(java.lang.String name) Adds a new timing diagram to the package.
[IRPType](#)	**addType**(java.lang.String name) Adds a new type to the package.
[IRPUseCase](#)	

Method Summary

	addUseCase(java.lang.String name) Adds a new use case to the package.
IRPUseCaseDiagram	**addUseCaseDiagram**(java.lang.String name) Adds a new use case diagram to the package.
void	**deleteActor**(IRPActor actor) Deletes the specified actor.
void	**deleteClass**(IRPClass theClass) Deletes the specified class.
void	**deleteCollaborationDiagram**(java.lang.String name) Deletes the collaboration diagram with the specified name.
void	**deleteComponentDiagram**(java.lang.String name) Deletes the component diagram with the specified name.
void	**deleteDeploymentDiagram**(java.lang.String name) Deletes the deployment diagram with the specified name.
void	**deleteEvent**(IRPEvent event) Deletes the specified event.
void	**deleteFlowItems**(IRPFlowItem pItem) Deletes the specified item flow.
void	**deleteFlows**(IRPFlow pFlow) Deletes the specified flow.
void	**deleteGlobalFunction**(IRPOperation operation) Deletes the specified global function.
void	**deleteGlobalObject**(IRPRelation relation) Deletes the specified object.
void	**deleteGlobalVariable**(IRPAttribute attribute) Deletes the specified global variable.
void	**deleteNode**(java.lang.String name) Deletes the Node element with the specified name.
void	**deleteObjectModelDiagram**(java.lang.String name) Deletes the object model diagram with the specified name.
void	**deletePackage**() Deletes the package.
void	**deletePanelDiagram**(java.lang.String name) Deletes the panel diagram with the specified name.
void	**deleteSequenceDiagram**(java.lang.String name) Deletes the sequence diagram with the specified name.
void	**deleteTimingDiagram**(java.lang.String name) Deletes the timing diagram with the specified name.

Method Summary

void	**deleteType**([IRPType](#) type) Deletes the specified type.
void	**deleteUseCase**([IRPUseCase](#) useCase) Deletes the specified use case.
void	**deleteUseCaseDiagram**(java.lang.String name) Deletes the use case diagram with the specified name.
[IRPActor](#)	**findActor**(java.lang.String name) Returns the actor with the specified name.
[IRPModelElement](#)	**findAllByName**(java.lang.String name, java.lang.String metaClass) Searches the package for a model element of the specified type with the specified name.
[IRPClass](#)	**findClass**(java.lang.String name) Returns the class with the specified name.
[IRPEvent](#)	**findEvent**(java.lang.String name) Returns the event with the specified name.
[IRPOperation](#)	**findGlobalFunction**(java.lang.String name) Returns the global function with the specified name.
[IRPRelation](#)	**findGlobalObject**(java.lang.String name) Returns the Object with the specified name.
[IRPAttribute](#)	**findGlobalVariable**(java.lang.String name) Returns the global variable with the specified name.
[IRPNode](#)	**findNode**(java.lang.String name) Returns the Node element with the specified name.
[IRPType](#)	**findType**(java.lang.String name) Returns the type with the specified name.
[IRPCollection](#)	**findUsage**([IRPModelElement](#) objToFind) Returns a collection of the elements in the current package that are related to the specified model element.
[IRPUseCase](#)	**findUseCase**(java.lang.String name) Returns the use case with the specified name.
[IRPCollection](#)	**getActors**() Returns a collection of all the actors in the package.
[IRPCollection](#)	**getAllNestedElements**() Returns a collection of all the model elements that are directly under the current package, including functions, global variables, and global objects.
[IRPCollection](#)	**getBehavioralDiagrams**() Returns a collection of all the activity diagrams in the package.
[IRPCollection](#)	**getClasses**() Returns a collection of all the classes in the package.

Method Summary

[IRPCollection](#)	**getCollaborationDiagrams**() Returns a collection of all the collaboration diagrams in the package.
[IRPCollection](#)	**getComponentDiagrams**() Returns a collection of all the component diagrams in the package.
[IRPCollection](#)	**getDeploymentDiagrams**() Returns a collection of all the deployment diagrams in the package.
[IRPCollection](#)	**getEvents**() Returns a collection of all the events in the package.
int	**getEventsBaseId**() Returns the start number used for assigning IDs to events in the package.
[IRPCollection](#)	**getFlowItems**() Returns a collection of all the item flows in the package.
[IRPCollection](#)	**getFlows**() Returns a collection of all the flows in the package.
[IRPCollection](#)	**getGlobalFunctions**() Returns a collection of all the global functions in the package.
[IRPCollection](#)	**getGlobalObjects**() Returns a collection of all the Objects in the package.
[IRPCollection](#)	**getGlobalVariables**() Returns a collection of all the global variables in the package.
[IRPCollection](#)	**getInstanceSpecifications**() Returns a collection of all the instance specifications in the package.
[IRPCollection](#)	**getLinks**() Returns a collection of all the Links in the package.
[IRPCollection](#)	**getModules**() Returns a collection of all the File elements in the package.
java.lang.String	**getNamespace**() getNamespace
[IRPCollection](#)	**getNestedClassifiers**() Returns a collection of all the classifiers in the package.
[IRPCollection](#)	**getNestedComponents**() Returns a collection of all the Components in the package.
[IRPCollection](#)	**getNodes**() Returns a collection of all the Node elements in the package.
[IRPCollection](#)	**getObjectModelDiagrams**() Returns a collection of all the object model diagrams in the package.
[IRPCollection](#)	**getPackages**() Returns a collection of all the nested packages in the package.

Method Summary

[IRPCollection](#)	**getPanelDiagrams**() Returns a collection of all the panel diagrams in the package.
java.lang.String	**getRemoteRequirementsPopulateMode**() Returns the mode that was selected for loading remote requirements in the collection.
int	**getSavedInSeperateDirectory**() Checks whether the package is configured to be saved in a separate directory.
[IRPCollection](#)	**getSequenceDiagrams**() Returns a collection of all the sequence diagrams in the package.
[IRPCollection](#)	**getSourceArtifacts**() Gets the source artifacts for the package.
[IRPCollection](#)	**getTimingDiagrams**() Returns a collection of all the timing diagrams in the package.
[IRPCollection](#)	**getTypes**() Returns a collection of all the types in the package.
[IRPCollection](#)	**getUseCaseDiagrams**() Returns a collection of all the use case diagrams in the package.
[IRPCollection](#)	**getUseCases**() Returns a collection of all the use cases in the package.
[IRPCollection](#)	**getUserDefinedStereotypes**() Returns a collection of all the user-defined stereotypes in the package.
void	**populateRemoteRequirements**() For Design Manager projects, populates the package with the remote requirements that model elements do not yet have dependencies upon.
int	**reCalculateEventsBaseId**() If you are using Rational Rhapsody's default numbering scheme for event IDs, then a certain amount of IDs are reserved for each package.
void	**setRemoteRequirementsPopulateMode**(java.lang.String populateMode) For collections of remote requirements, you can use setRemoteRequirementsPopulateMode to specify which requirements in the collection should be loaded when you open the model - all the requirements, only the requirements that have OSLC links to model elements, or none of the requirements.
void	**setSavedInSeperateDirectory**(int savedInSeperateDirectory) Specifies whether the package should be saved in a separate directory.

Methods inherited from interface com.telelogic.rhapsody.core.[IRPUnit](#)

[copyToAnotherProject](#), [getAddToModelMode](#), [getCMHeader](#), [getCMState](#), [getCurrentDirectory](#), [getFilename](#), [getIncludeInNextLoad](#), [getIsStub](#), [getLanguage](#), [getLastModifiedTime](#), [getNestedSaveUnits](#), [getNestedSaveUnitsCount](#), [getStructureDiagrams](#), [isReadOnly](#),

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addActivityDiagram

IRPFlowchart **addActivityDiagram**()

Adds a new activity diagram to the package.
Returns:
the activity diagram that was created

addActor

IRPActor **addActor**(java.lang.String name)

Adds a new actor to the package.

Parameters:

name - the name to use for the new actor

Returns:

the actor that was created

---

addClass

IRPClass **addClass**(java.lang.String name)

Adds a new class to the package.

Parameters:

name - the name to use for the new class

Returns:

the class that was created

---

addCollaborationDiagram

IRPCollaborationDiagram **addCollaborationDiagram**(java.lang.String name)

Adds a new collaboration diagram to the package.

Parameters:

name - the name to use for the new collaboration diagram

Returns:

the collaboration diagram that was created

---

addComponentDiagram

IRPComponentDiagram **addComponentDiagram**(java.lang.String name)

Adds a new component diagram to the package.

Parameters:

name - the name to use for the new component diagram

Returns:

the component diagram that was created

---

addDeploymentDiagram

IRPDeploymentDiagram **addDeploymentDiagram**(java.lang.String name)

Adds a new deployment diagram to the package.

Parameters:

name - the name to use for the new deployment diagram

Returns:

the deployment diagram that was created

---

addEvent

IRPEvent **addEvent**(java.lang.String name)

>
> Adds a new event to the package.
> **Parameters:**
> > name - the name to use for the new event
> **Returns:**
> > the event that was created

addFlowItems

IRPFlowItem **addFlowItems**(java.lang.String name)

>
> Adds an item flow to the package.
> **Parameters:**
> > name - the name to use for the new item flow
> **Returns:**
> > the item flow created

addFlows

IRPFlow **addFlows**(java.lang.String name)

>
> Adds a flow to the package.
> **Parameters:**
> > name - the name to use for the new flow
> **Returns:**
> > the flow created

addGlobalFunction

IRPOperation **addGlobalFunction**(java.lang.String name)

>
> Adds a global function to the package.
> **Parameters:**
> > name - the name to use for the new function
> **Returns:**
> > the function created

addGlobalObject

IRPRelation **addGlobalObject**(java.lang.String name,
                                java.lang.String otherClassName,
                                java.lang.String otherClassPackageName)

Adds an Object to the package. This method is for adding instances of existing classes. To add an implicit object, use the method addImplicitObject.
Parameters:
> name - the name to use for the new object
>
> otherClassName - the name of the class that the new object should be an instance of
>
> otherClassPackageName - the name of the package that contains the class. You must specify this argument even if you are adding the object to the package that contains the class you are instantiating

Returns:
> the object that was created

addGlobalVariable

IRPAttribute **addGlobalVariable**(java.lang.String name)

Adds a global variable to the package.
Parameters:
> name - the name to use for the variable

Returns:
> the variable created

addImplicitObject

IRPRelation **addImplicitObject**(java.lang.String name)

Adds an implicit object to the package. This is relevant only for C and C++ models.
Parameters:
> name - the name to use for the new object

Returns:
> the object that was created

addLink

IRPLink **addLink**(IRPInstance fromPart,
                IRPInstance toPart,
                IRPRelation assoc,
                IRPPort fromPort,
                IRPPort toPort)

Creates a link between two objects in the package. In addition to specifying the two objects, you must specify the association that the link should represent, or, alternatively, the two ports that should be used for the link. If you provide the two ports as arguments, you should use Null for the association argument. Similarly, if you specify an association, you should use Null for the two port arguments. Note that if you are not specifying the two ports, you must provide an association as an argument even if there is only one relevant association.
Parameters:
> fromPart - the "from" object for the link
>
> toPart - the "to" object for the link

assoc - the association that the link should represent

fromPort - the "from" port for the link

toPort - the "to" port for the link

Returns:

the link created

---

addLinkBetweenSYSMLPorts

IRPLink **addLinkBetweenSYSMLPorts**(IRPInstance fromPart,
                                     IRPInstance toPart,
                                     IRPRelation assoc,
                                     IRPSysMLPort fromPort,
                                     IRPSysMLPort toPort)

Creates a link between two objects. In addition to specifying the two objects, you must specify the association that the link should represent, or, alternatively, the two flow ports that should be used for the link. If you provide the two flow ports as arguments, you should use Null for the association argument. Similarly, if you specify an association, you should use Null for the two flow port arguments. Note that if you are not specifying the two flow ports, you must provide an association as an argument even if there is only one relevant association.

Parameters:

fromPart - the "from" object for the link

toPart - the "to" object for the link

assoc - the association that the link should represent

fromPort - the "from" flow port for the link

toPort - the "to" flow port for the link

Returns:

the link created

Throws:

RhapsodyRuntimeException

---

addModule

IRPModule **addModule**(java.lang.String name)

Adds a new File element to the package.

Parameters:

name - the name to use for the new File

Returns:

the File element that was created

---

addNestedPackage

IRPPackage **addNestedPackage**(java.lang.String name)

Adds a nested package to the package.

Parameters:

name - the name to use for the new package

Returns:
>   the package created

---

addNode

IRPNode **addNode**(java.lang.String name)

>   Adds a Node element to the package.
>   **Parameters:**
> >   name - the name to use for the new Node element
>
>   **Returns:**
> >   the Node element created

---

addObjectModelDiagram

IRPObjectModelDiagram **addObjectModelDiagram**(java.lang.String name)

>   Adds a new object model diagram to the package.
>   **Parameters:**
> >   name - the name to use for the new object model diagram
>
>   **Returns:**
> >   the object model diagram that was created

---

addPanelDiagram

IRPPanelDiagram **addPanelDiagram**(java.lang.String name)

>   Adds a new panel diagram to the package.
>   **Parameters:**
> >   name - the name to use for the new panel diagram
>
>   **Returns:**
> >   the panel diagram that was created

---

addSequenceDiagram

IRPSequenceDiagram **addSequenceDiagram**(java.lang.String name)

>   Adds a new sequence diagram to the package.
>   **Parameters:**
> >   name - the name to use for the new sequence diagram
>
>   **Returns:**
> >   the sequence diagram that was created

---

addStatechart

[IRPStatechart](#) **addStatechart**()

>
> Adds a new statechart to the package.
> **Returns:**
>> the statechart that was created

---

addTimingDiagram

[IRPTimingDiagram](#) **addTimingDiagram**(java.lang.String name)

>
> Adds a new timing diagram to the package.
> **Parameters:**
>> name - the name to use for the new timing diagram
> **Returns:**
>> the timing diagram that was created

---

addType

[IRPType](#) **addType**(java.lang.String name)

>
> Adds a new type to the package.
> **Parameters:**
>> name - the name to use for the new type
> **Returns:**
>> the type that was created

---

addUseCase

[IRPUseCase](#) **addUseCase**(java.lang.String name)

>
> Adds a new use case to the package.
> **Parameters:**
>> name - the name to use for the new use case
> **Returns:**
>> the use case that was created

---

addUseCaseDiagram

[IRPUseCaseDiagram](#) **addUseCaseDiagram**(java.lang.String name)

>
> Adds a new use case diagram to the package.
> **Parameters:**
>> name - the name to use for the new use case diagram
> **Returns:**

the use case diagram that was created

deleteActor

void **deleteActor**([IRPActor](#) actor)

Deletes the specified actor.
Parameters:
actor - that actor that should be deleted

deleteClass

void **deleteClass**([IRPClass](#) theClass)

Deletes the specified class.
Parameters:
theClass - the class that should be deleted

deleteCollaborationDiagram

void **deleteCollaborationDiagram**(java.lang.String name)

Deletes the collaboration diagram with the specified name.
Parameters:
name - the name of the collaboration diagram to delete

deleteComponentDiagram

void **deleteComponentDiagram**(java.lang.String name)

Deletes the component diagram with the specified name.
Parameters:
name - the name of the component diagram to delete

deleteDeploymentDiagram

void **deleteDeploymentDiagram**(java.lang.String name)

Deletes the deployment diagram with the specified name.
Parameters:
name - the name of the deployment diagram to delete

deleteEvent

void **deleteEvent**(IRPEvent event)

> Deletes the specified event.
> **Parameters:**
> > event - the event that should be deleted

deleteFlowItems

void **deleteFlowItems**(IRPFlowItem pItem)

> Deletes the specified item flow.
> **Parameters:**
> > pItem - the item flow that should be deleted

deleteFlows

void **deleteFlows**(IRPFlow pFlow)

> Deletes the specified flow.
> **Parameters:**
> > pFlow - the flow that should be deleted

deleteGlobalFunction

void **deleteGlobalFunction**(IRPOperation operation)

> Deletes the specified global function.
> **Parameters:**
> > operation - the global function that should be deleted

deleteGlobalObject

void **deleteGlobalObject**(IRPRelation relation)

> Deletes the specified object.
> **Parameters:**
> > relation - the object that should be deleted

deleteGlobalVariable

void **deleteGlobalVariable**(IRPAttribute attribute)

> Deletes the specified global variable.

Parameters:
> `attribute` - the global variable that should be deleted

deleteNode

void **deleteNode**(java.lang.String name)

> Deletes the Node element with the specified name.
> **Parameters:**
>> `name` - the name of the node to delete

deleteObjectModelDiagram

void **deleteObjectModelDiagram**(java.lang.String name)

> Deletes the object model diagram with the specified name.
> **Parameters:**
>> `name` - the name of the object model diagram to delete

deletePackage

void **deletePackage**()

> Deletes the package.

deletePanelDiagram

void **deletePanelDiagram**(java.lang.String name)

> Deletes the panel diagram with the specified name.
> **Parameters:**
>> `name` - the name of the panel diagram to delete

deleteSequenceDiagram

void **deleteSequenceDiagram**(java.lang.String name)

> Deletes the sequence diagram with the specified name.
> **Parameters:**
>> `name` - the name of the sequence diagram to delete

deleteTimingDiagram

void **deleteTimingDiagram**(java.lang.String name)

Deletes the timing diagram with the specified name.
Parameters:
　　name - the name of the timing diagram to delete

---

deleteType

void **deleteType**(IRPType type)

Deletes the specified type.
Parameters:
　　type - the type that should be deleted

---

deleteUseCase

void **deleteUseCase**(IRPUseCase useCase)

Deletes the specified use case.
Parameters:
　　useCase - the use case that should be deleted

---

deleteUseCaseDiagram

void **deleteUseCaseDiagram**(java.lang.String name)

Deletes the use case diagram with the specified name.
Parameters:
　　name - the name of the use case diagram to delete

---

findActor

IRPActor **findActor**(java.lang.String name)

Returns the actor with the specified name.
Parameters:
　　name - the name of the actor to return
Returns:
　　the actor with the name specified

---

findAllByName

```
IRPModelElement findAllByName(java.lang.String name,
                              java.lang.String metaClass)
```

Searches the package for a model element of the specified type with the specified name. Note that the search is carried out recursively if the package contains nested packages. In cases where there are multiple elements that meet the search criteria, the first such element encountered will be returned.
Parameters:
>   name - the name of the element to find
>   metaClass - the metaclass of the element to find

Returns:
>   the first element found that satisfies the search criteria

findClass

```
IRPClass findClass(java.lang.String name)
```

Returns the class with the specified name.
Parameters:
>   name - the name of the class to return

Returns:
>   the class with the specified name

findEvent

```
IRPEvent findEvent(java.lang.String name)
```

Returns the event with the specified name.
Parameters:
>   name - the name of the event to return

Returns:
>   the event with the specified name

findGlobalFunction

```
IRPOperation findGlobalFunction(java.lang.String name)
```

Returns the global function with the specified name.
Parameters:
>   name - the name of the function to return

Returns:
>   the global function with the specified name

findGlobalObject

[IRPRelation](#) **findGlobalObject**(java.lang.String name)

> Returns the Object with the specified name.
> **Parameters:**
>> name - the name of the Object to return
> **Returns:**
>> the Object with the specified name

findGlobalVariable

[IRPAttribute](#) **findGlobalVariable**(java.lang.String name)

> Returns the global variable with the specified name.
> **Parameters:**
>> name - the name of the variable to return
> **Returns:**
>> the global variable with the specified name

findNode

[IRPNode](#) **findNode**(java.lang.String name)

> Returns the Node element with the specified name.
> **Parameters:**
>> name - the name of the node to return
> **Returns:**
>> the Node element with the specified name

findType

[IRPType](#) **findType**(java.lang.String name)

> Returns the type with the specified name.
> **Parameters:**
>> name - the name of the type to return
> **Returns:**
>> the type with the specified name

findUsage

[IRPCollection](#) **findUsage**([IRPModelElement](#) objToFind)

> Returns a collection of the elements in the current package that are related to the specified model
> element. Note that the type of relations searched for depends upon the type of the element specified.

For a more comprehensive list of references to the element, use the method
IRPModelElement.getReferences(). (Keep in mind that getReferences() searches the entire model, not
just the current package.)
Parameters:
>   objToFind - the element whose references you want to find
Returns:
>   the elements in the current package that are related to the specified model element

---

findUseCase

IRPUseCase **findUseCase**(java.lang.String name)

>   Returns the use case with the specified name.
>   **Parameters:**
>   >   name - the name of the use case to return
>   **Returns:**
>   >   the use case with the specified name

---

getActors

IRPCollection **getActors**()

>   Returns a collection of all the actors in the package.
>   **Returns:**
>   >   all the actors in the package

---

getAllNestedElements

IRPCollection **getAllNestedElements**()

>   Returns a collection of all the model elements that are directly under the current package, including
>   functions, global variables, and global objects.
>   **Returns:**
>   >   collection of all the model elements that are directly under the current package, including
>   >   functions, global variables, and global objects

---

getBehavioralDiagrams

IRPCollection **getBehavioralDiagrams**()

>   Returns a collection of all the activity diagrams in the package. Note that this includes only the
>   activity diagrams directly under the package, not diagrams belonging to classes in the package.
>   **Returns:**
>   >   all the activity diagrams directly under the package

---

getClasses

[IRPCollection](#) **getClasses**()

>> Returns a collection of all the classes in the package.
>> **Returns:**
>>> all the classes in the package

getCollaborationDiagrams

[IRPCollection](#) **getCollaborationDiagrams**()

>> Returns a collection of all the collaboration diagrams in the package.
>> **Returns:**
>>> all the collaboration diagrams in the package

getComponentDiagrams

[IRPCollection](#) **getComponentDiagrams**()

>> Returns a collection of all the component diagrams in the package.
>> **Returns:**
>>> all the component diagrams in the package

getDeploymentDiagrams

[IRPCollection](#) **getDeploymentDiagrams**()

>> Returns a collection of all the deployment diagrams in the package.
>> **Returns:**
>>> all the deployment diagrams in the package

getEvents

[IRPCollection](#) **getEvents**()

>> Returns a collection of all the events in the package.
>> **Returns:**
>>> all the events in the package

getEventsBaseId

int **getEventsBaseId**()

Returns the start number used for assigning IDs to events in the package. This value is controlled by the property EventsBaseID.

Returns:

the start number used for assigning IDs to events in the package

getFlowItems

IRPCollection **getFlowItems**()

Returns a collection of all the item flows in the package.

Returns:

all the item flows in the package

getFlows

IRPCollection **getFlows**()

Returns a collection of all the flows in the package.

Returns:

all the flows in the package

getGlobalFunctions

IRPCollection **getGlobalFunctions**()

Returns a collection of all the global functions in the package.

Returns:

all the global functions in the package

getGlobalObjects

IRPCollection **getGlobalObjects**()

Returns a collection of all the Objects in the package.

Returns:

all the Objects in the package

getGlobalVariables

IRPCollection **getGlobalVariables**()

Returns a collection of all the global variables in the package.

Returns:

all the global variables in the package

getInstanceSpecifications

[IRPCollection](#) **getInstanceSpecifications**()

Returns a collection of all the instance specifications in the package.
Returns:
all the instance specifications in the package

getLinks

[IRPCollection](#) **getLinks**()

Returns a collection of all the Links in the package.
Returns:
all the Links in the package

getModules

[IRPCollection](#) **getModules**()

Returns a collection of all the File elements in the package.
Returns:
all the File elements in the package

getNamespace

java.lang.String **getNamespace**()

getNamespace
Throws:
[RhapsodyRuntimeException](#)

getNestedClassifiers

[IRPCollection](#) **getNestedClassifiers**()

Returns a collection of all the classifiers in the package.
Returns:
all the classifiers in the package

getNestedComponents

[IRPCollection](#) **getNestedComponents**()

Returns a collection of all the Components in the package.

Returns:

all the Components in the package

getNodes

`IRPCollection` **getNodes**`()`

Returns a collection of all the Node elements in the package.
Returns:

all the Node elements in the package

getObjectModelDiagrams

`IRPCollection` **getObjectModelDiagrams**`()`

Returns a collection of all the object model diagrams in the package.
Returns:

all the object model diagrams in the package

getPackages

`IRPCollection` **getPackages**`()`

Returns a collection of all the nested packages in the package.
Returns:

all the nested packages in the package

getPanelDiagrams

`IRPCollection` **getPanelDiagrams**`()`

Returns a collection of all the panel diagrams in the package.
Returns:

all the panel diagrams in the package

getRemoteRequirementsPopulateMode

`java.lang.String` **getRemoteRequirementsPopulateMode**`()`

Returns the mode that was selected for loading remote requirements in the collection.
Returns:

the mode that was selected for loading remote requirements in the collection - will be one of
the following values: "All", "Linked", "None"
Throws:

`RhapsodyRuntimeException`

getNestedComponents 457

getSavedInSeperateDirectory

int **getSavedInSeperateDirectory**()

     Checks whether the package is configured to be saved in a separate directory.
     Returns:
          1 if the package is configured to be saved in a separate directory, 0 otherwise

getSequenceDiagrams

IRPCollection **getSequenceDiagrams**()

     Returns a collection of all the sequence diagrams in the package.
     Returns:
          all the sequence diagrams in the package

getSourceArtifacts

IRPCollection **getSourceArtifacts**()

     Gets the source artifacts for the package.
     Returns:
          the source artifacts for the package, as a collection of IRPFile objects

getTimingDiagrams

IRPCollection **getTimingDiagrams**()

     Returns a collection of all the timing diagrams in the package.
     Returns:
          all the timing diagrams in the package

getTypes

IRPCollection **getTypes**()

     Returns a collection of all the types in the package.
     Returns:
          all the types in the package

getUseCaseDiagrams

IRPCollection **getUseCaseDiagrams**()

>Returns a collection of all the use case diagrams in the package.
>**Returns:**
>>all the use case diagrams in the package

getUseCases

IRPCollection **getUseCases**()

>Returns a collection of all the use cases in the package.
>**Returns:**
>>all the use cases in the package

getUserDefinedStereotypes

IRPCollection **getUserDefinedStereotypes**()

>Returns a collection of all the user-defined stereotypes in the package.
>**Returns:**
>>all the user-defined stereotypes in the package

populateRemoteRequirements

void **populateRemoteRequirements**()

>For Design Manager projects, populates the package with the remote requirements that model elements do not yet have dependencies upon. This method corresponds to the "populate with all existing requirements" option that the UI provides for "Remote Resource" packages. Once these requirements have been added to the package, you can add dependencies to these requirements by using the "link to remote requirement" option.

reCalculateEventsBaseId

int **reCalculateEventsBaseId**()

>If you are using Rational Rhapsody's default numbering scheme for event IDs, then a certain amount of IDs are reserved for each package. As a result, there are situations where the IDs used for events in a given package may not be continuous. In cases like this, you can use the method reCalculateEventsBaseId to have the event ID numbering recalculated so that event IDs are continuous for all events in the package.
>**Returns:**
>>the new start number for event IDs in the package

setRemoteRequirementsPopulateMode

void **setRemoteRequirementsPopulateMode**(java.lang.String populateMode)

> For collections of remote requirements, you can use setRemoteRequirementsPopulateMode to specify which requirements in the collection should be loaded when you open the model - all the requirements, only the requirements that have OSLC links to model elements, or none of the requirements.
>
> **Parameters:**
> > populateMode - can be one of the following values: "All", "Linked", "None"
>
> **Throws:**
> > RhapsodyRuntimeException

setSavedInSeperateDirectory

void **setSavedInSeperateDirectory**(int savedInSeperateDirectory)

> Specifies whether the package should be saved in a separate directory.
>
> **Parameters:**
> > savedInSeperateDirectory - Use 1 to specify that the package should be saved in a separate directory. Use 0 to specify that the package should not be saved in a separate directory.

Package | **Class** | **Use** | **Tree** | **Serialized** | **Deprecated** | **Index** | **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPPanelDiagram

All Superinterfaces:
> IRPDiagram, IRPModelElement, IRPUnit

---

public interface **IRPPanelDiagram**
extends IRPDiagram

The IRPPanelDiagram interface represents panel diagrams in Rational Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

Methods inherited from interface com.telelogic.rhapsody.core.IRPDiagram
addFreeShapeByType, addImage, addNewEdgeByType, addNewEdgeForElement, addNewNodeByType, addNewNodeForElement, addTextBox, closeDiagram, completeRelations, getCorrespondingGraphicElements, getElementsInDiagram, getGraphicalElements, getLastVisualizationModifiedTime, getPicture, getPictureAs, getPictureAsDividedMetafiles, getPictureEx, getPicturesWithImageMap, isOpen, isShowDiagramFrame, openDiagram, openDiagramView, populateDiagram, removeGraphElements, setShowDiagramFrame

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit
copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype,

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

---

com.telelogic.rhapsody.core
Interface IRPPin

All Superinterfaces:
> IRPConnector, IRPModelElement, IRPStateVertex

```
public interface IRPPin
```
extends IRPConnector

The IRPPin interface represents action pins added to actions, or activity parameters added to action blocks, in an activity diagram. To add an action pin to an action, use IRPState.addConnector, for example:
```
action1.addConnector("InPin"), action1.addConnector("OutPin"), or
action1.addConnector("InOutPin").
```

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement	
IRPModelElement.OSLCLink	

Method Summary

int	**getIsParameter**() 　　　Checks whether the element is an activity parameter or an action pin.
java.lang.String	**getPinDirection**() 　　　Returns the direction of the pin/parameter: In, Out, or InOut.
IRPClassifier	**getPinType**() 　　　Returns the type of the value held by the pin/parameter.
void	**setIsParameter**(int isParameter) 　　　Specifies whether the element should be an activity parameter or an action pin.
void	**setPinDirection**(java.lang.String pinDirection) 　　　Specifies the direction of the pin/parameter.
void	**setPinType**(IRPClassifier pinType) 　　　Specifies the type to use for the value held by the pin/parameter.

Methods inherited from interface com.telelogic.rhapsody.core.IRPConnector

Methods inherited from interface com.telelogic.rhapsody.core.IRPConnector

createDefaultTransition, getConnectorType, getDerivedInEdges, getDerivedOutEdge, getItsSwimlane, getOfState, isConditionConnector, isDiagramConnector, isForkConnector, isHistoryConnector, isJoinConnector, isJunctionConnector, isStubConnector, isTerminationConnector, setItsSwimlane, setOfState

Methods inherited from interface com.telelogic.rhapsody.core.IRPStateVertex

addFlow, addTransition, deleteTransition, getInTransitions, getOutTransitions, getParent, setParent

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getIsParameter

```
int getIsParameter()
```

> Checks whether the element is an activity parameter or an action pin.
> **Returns:**
> > 1 if the element is an activity parameter, 0 if the element is an action pin.

getPinDirection

java.lang.String **getPinDirection**()

> Returns the direction of the pin/parameter: In, Out, or InOut.
> **Returns:**
>> the direction of the pin/parameter

getPinType

IRPClassifier **getPinType**()

> Returns the type of the value held by the pin/parameter.
> **Returns:**
>> the type of the value held by the pin/parameter

setIsParameter

void **setIsParameter**(int isParameter)

> Specifies whether the element should be an activity parameter or an action pin.
> **Parameters:**
>> isParameter - use 1 if you want the element to be an activity parameter, use 0 if you want the element to be an action pin

setPinDirection

void **setPinDirection**(java.lang.String pinDirection)

> Specifies the direction of the pin/parameter.
> **Parameters:**
>> pinDirection - the direction that should be used for the pin/parameter. The valid strings for this parameter are: In, Out, and InOut

setPinType

void **setPinType**(IRPClassifier pinType)

> Specifies the type to use for the value held by the pin/parameter.
> **Parameters:**
>> pinType - the type to use for the value held by the pin/parameter

Package **Class** **[Use](Use)** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**

getPinDirection   465

com.telelogic.rhapsody.core
Interface IRPPlugInWindow

public interface **IRPPlugInWindow**

Method Summary

void	**destroyWindow**() Destroy window
int	**getDocking**() Get docking mode
java.lang.String	**getInterfaceName**() get property interfaceName
java.lang.String	**getPosString**() Get position string
long	**getWindowHandle**() Get window handle
void	**setDocking**(int nDockPos) Set docking mode.
void	**setPosString**(java.lang.String sPos) Set position string
void	**setTitle**(java.lang.String sTitle) Set window title
void	**showWindow**(int nShow) Show or hide window

Method Detail

destroyWindow

void **destroyWindow**()

Destroy window
Throws:

getDocking

int **getDocking**()

> Get docking mode
> **Throws:**
> > RhapsodyRuntimeException

getPosString

java.lang.String **getPosString**()

> Get position string
> **Throws:**
> > RhapsodyRuntimeException

getWindowHandle

long **getWindowHandle**()

> Get window handle
> **Throws:**
> > RhapsodyRuntimeException

setDocking

void **setDocking**(int nDockPos)

> Set docking mode. 0=floating, 1=top, 2=left, 3=right, 4=bottom
> **Throws:**
> > RhapsodyRuntimeException

setPosString

void **setPosString**(java.lang.String sPos)

> Set position string
> **Throws:**
> > RhapsodyRuntimeException

setTitle

void **setTitle**(java.lang.String sTitle)

> Set window title
> **Throws:**
> > [RhapsodyRuntimeException](#)

showWindow

void **showWindow**(int nShow)

> Show or hide window
> **Throws:**
> > [RhapsodyRuntimeException](#)

getInterfaceName

java.lang.String **getInterfaceName**()

> get property interfaceName
> **Throws:**
> > [RhapsodyRuntimeException](#)

[Package](#)   **Class**   **[Use](#)**   **[Tree](#)**   **[Serialized](#)**   **[Deprecated](#)**   **[Index](#)**   **[Help](#)**

[PREV CLASS](#)   **[NEXT CLASS](#)**   **[FRAMES](#)**   **[NO FRAMES](#)**   **[All Classes](#)**

SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#)   DETAIL: FIELD | CONSTR | [METHOD](#)

com.telelogic.rhapsody.core
Interface IRPPort

All Superinterfaces:
> IRPInstance, IRPModelElement, IRPRelation, IRPUnit

public interface **IRPPort**
extends IRPInstance

The IRPPort interface represents ports in Rational Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

void	**addProvidedInterface**(IRPClass newVal) method addRProvidedInterface
void	**addRequiredInterface**(IRPClass newVal) method addRequiredInterface
IRPClass	**getContract**() get property contract
int	**getIsBehavioral**() get property isBehavioral
int	**getIsReversed**() get property isReversed
IRPCollection	**getProvidedInterfaces**() get property providedInterfaces
IRPCollection	**getRequiredInterfaces**() get property requiredInterfaces
void	**removeProvidedInterface**(IRPClass newVal) method removeProvidedInterface
void	

Method Summary

	removeRequiredInterface(IRPClass newVal) method removeRequiredInterface
void	**setContract**(IRPClass contract) set property contract
void	**setIsBehavioral**(int isBehavioral) set property isBehavioral
void	**setIsReversed**(int isReversed) set property isReversed

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPInstance

addRelationToTheWhole, getAttributeValue, getInLinks, getInstantiatedBy, getListOfInitializerArguments, getOutLinks, setAttributeValue, setExplicit, setImplicit, setInitializerArgumentValue, setInstantiatedBy

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPRelation

addQualifier, getAssociationClass, getInverse, getIsNavigable, getIsSymmetric, getMultiplicity, getObjectAsObjectType, getOfClass, getOtherClass, getQualifier, getQualifiers, getQualifierType, getRelationLabel, getRelationLinkName, getRelationRoleName, getRelationType, getVisibility, isTypelessObject, makeUnidirect, removeQualifier, setInverse, setIsNavigable, setMultiplicity, setOfClass, setOtherClass, setQualifier, setQualifierType, setRelationLabel, setRelationLinkName, setRelationRoleName, setRelationType

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit,

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addProvidedInterface

void **addProvidedInterface**(IRPClass newVal)

> method addRProvidedInterface
> **Throws:**
> > RhapsodyRuntimeException

addRequiredInterface

void **addRequiredInterface**(IRPClass newVal)

> method addRequiredInterface
> **Throws:**
> > RhapsodyRuntimeException

getContract

IRPClass **getContract**()

> get property contract
> **Throws:**
> > RhapsodyRuntimeException

getIsBehavioral

int **getIsBehavioral**()

> get property isBehavioral
> **Throws:**

RhapsodyRuntimeException

---

getIsReversed

int **getIsReversed**()

get property isReversed
Throws:
RhapsodyRuntimeException

---

getProvidedInterfaces

IRPCollection **getProvidedInterfaces**()

get property providedInterfaces
Throws:
RhapsodyRuntimeException

---

getRequiredInterfaces

IRPCollection **getRequiredInterfaces**()

get property requiredInterfaces
Throws:
RhapsodyRuntimeException

---

removeProvidedInterface

void **removeProvidedInterface**(IRPClass newVal)

method removeProvidedInterface
Throws:
RhapsodyRuntimeException

---

removeRequiredInterface

void **removeRequiredInterface**(IRPClass newVal)

method removeRequiredInterface
Throws:
RhapsodyRuntimeException

---

getIsBehavioral 473

setContract

void **setContract**(IRPClass contract)

> set property contract
> **Throws:**
>> RhapsodyRuntimeException

---

setIsBehavioral

void **setIsBehavioral**(int isBehavioral)

> set property isBehavioral
> **Throws:**
>> RhapsodyRuntimeException

---

setIsReversed

void **setIsReversed**(int isReversed)

> set property isReversed
> **Throws:**
>> RhapsodyRuntimeException

Package	**Class**	**[Use](Use)**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**	
PREV CLASS **NEXT CLASS**					**FRAMES** **NO FRAMES** **All Classes**			
SUMMARY: NESTED	FIELD	CONSTR	METHOD				DETAIL: FIELD	CONSTR	METHOD			

com.telelogic.rhapsody.core
Interface IRPProfile

All Superinterfaces:
> IRPModelElement, IRPPackage, IRPUnit

```
public interface IRPProfile
```
extends IRPPackage

The IRPProfile interface represents profiles in Rational Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

Methods inherited from interface com.telelogic.rhapsody.core.IRPPackage
addActivityDiagram, addActor, addClass, addCollaborationDiagram, addComponentDiagram, addDeploymentDiagram, addEvent, addFlowItems, addFlows, addGlobalFunction, addGlobalObject, addGlobalVariable, addImplicitObject, addLink, addLinkBetweenSYSMLPorts, addModule, addNestedPackage, addNode, addObjectModelDiagram, addPanelDiagram, addSequenceDiagram, addStatechart, addTimingDiagram, addType, addUseCase, addUseCaseDiagram, deleteActor, deleteClass, deleteCollaborationDiagram, deleteComponentDiagram, deleteDeploymentDiagram, deleteEvent, deleteFlowItems, deleteFlows, deleteGlobalFunction, deleteGlobalObject, deleteGlobalVariable, deleteNode, deleteObjectModelDiagram, deletePackage, deletePanelDiagram, deleteSequenceDiagram, deleteTimingDiagram, deleteType, deleteUseCase, deleteUseCaseDiagram, findActor, findAllByName, findClass, findEvent, findGlobalFunction, findGlobalObject, findGlobalVariable, findNode, findType, findUsage, findUseCase, getActors, getAllNestedElements, getBehavioralDiagrams, getClasses, getCollaborationDiagrams, getComponentDiagrams, getDeploymentDiagrams, getEvents, getEventsBaseId, getFlowItems, getFlows, getGlobalFunctions, getGlobalObjects, getGlobalVariables, getInstanceSpecifications, getLinks, getModules, getNamespace, getNestedClassifiers, getNestedComponents, getNodes, getObjectModelDiagrams, getPackages, getPanelDiagrams, getRemoteRequirementsPopulateMode, getSavedInSeperateDirectory, getSequenceDiagrams, getSourceArtifacts, getTimingDiagrams, getTypes, getUseCaseDiagrams, getUseCases, getUserDefinedStereotypes, populateRemoteRequirements, reCalculateEventsBaseId, setRemoteRequirementsPopulateMode, setSavedInSeperateDirectory

setIsReversed   475

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

---

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPProgressBar

public interface **IRPProgressBar**

Method Summary

java.lang.String	**getInterfaceName**() get property interfaceName
void	**reset**() method reset
void	**tick**(int amount) method tick

Method Detail

getInterfaceName

java.lang.String **getInterfaceName**()

> get property interfaceName
> **Throws:**
> > RhapsodyRuntimeException

reset

void **reset**()

> method reset
> **Throws:**
> > RhapsodyRuntimeException

tick

```
void tick(int amount)
```

> method tick
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

com.telelogic.rhapsody.core
Interface IRPProject

All Superinterfaces:
> IRPModelElement, IRPPackage, IRPUnit

```
public interface IRPProject
```
extends <u>IRPPackage</u>

The IRPProject interface represents Rational Rhapsody projects.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.<u>IRPModelElement</u>
<u>IRPModelElement.OSLCLink</u>

Method Summary

<u>IRPComponent</u>	**addComponent**(java.lang.String name) Adds a new Component to the project.
void	**addCustomViewOnBrowser**(<u>IRPPackage</u> customView) Applies the specified custom view to the model browser.
void	**addCustomViewOnDiagram**(<u>IRPDiagram</u> diagram, <u>IRPPackage</u> customView) Applies the specified custom view to the specified diagram.
<u>IRPPackage</u>	**addPackage**(java.lang.String name) Adds a new package to the project.
<u>IRPProfile</u>	**addProfile**(java.lang.String name) Adds a new profile to the project.
void	**addSpellCheckerResult**(java.lang.String value) For internal use only.
void	**allowAutoSave**(int allow) Can be used to temporarily disable autosaving of the model regardless of the current value of the property General::Model::AutoSaveInterval.
void	**allowNonUniqueNames**(int allow) For internal use only.

Method Summary

void	**applyBrowserCustomViewsOnDiagrams**(int newVal) Applies the custom views applied to the browser to all diagrams as well.
void	**applyRoundtripDiffMerge**(int magicNumber, IRPProject shadowModel, IRPCollection filesToUpdate) For internal use only.
void	**becomeActiveProject**() Makes this project the active project in Rhapsody.
void	**checkEventsBaseIdsSolveCollisions**() Checks the values of the event base IDs for all packages in the model, detects collisions between the IDs, and resolves any incorrect values and collisions.
void	**cleanUnresolvedElements**(IRPModelElement rootElement) Removes any unresolved elements from the model, starting at the level of the specified element and working downward.
void	**close**() Closes the project.
void	**closeCSVFile**(java.lang.String fullCSVFileName) Closes the tab in the Output window for the specified csv file.
void	**deleteComponent**(IRPComponent component) Deletes the specified Component.
void	**endTransactionOfNoCGInterest**() For internal use only.
IRPComponent	**findComponent**(java.lang.String name) Returns the Component with the specified name.
IRPModelElement	**findElementByBinaryID**(byte[] theID) Retrieves a model element based on its binary ID.
IRPModelElement	**findElementByFileName**(java.lang.String theFolderName, java.lang.String theFileName) Returns the top-level element in the specified Rhapsody unit file.
IRPModelElement	**findElementByGUID**(java.lang.String theGUID) Retrieves a model element based on its GUID.
IRPCollection	**findElementsWithOSLCLink**(java.lang.String type, java.lang.String purl) Returns a collection of all the model elements that have an OSLC link of the specified type to the specified target element.
void	**gatewayExportToXML**(java.lang.String filename, java.lang.String params) For internal use only.
void	**gatewayExportToXML2**(java.lang.String filename, java.lang.String params, IRPProject proj) For internal use only.
void	**generateReport**(java.lang.String modelscope, java.lang.String templatename, java.lang.String docType, java.lang.String filename, int showDocument, int silentMode)

Method Summary

	Generates a ReporterPLUS report for the model.
IRPComponent	**getActiveComponent**() Returns the active component.
IRPConfiguration	**getActiveConfiguration**() Returns the active configuration.
IRPCollection	**getActiveCustomViewsOnBrowser**() Returns a collection of the custom views currently applied to the browser.
IRPCollection	**getActiveCustomViewsOnDiagram**(*IRPDiagram* diagram) Returns a collection of the custom views currently applied to the specified diagram.
IRPCollection	**getAllStereotypes**() Returns a collection of all the stereotypes in the project.
IRPPackage	**getCgSimplifiedModelPackage**() Returns the package that contains the simplified model.
IRPCollection	**getCodeGeneratedFiles**() Returns a collection of filenames for the code files that will be generated for the current active component if you select the "regenerate" option.
IRPCollection	**getComponents**() Returns a collection of all the components in the project.
java.lang.String	**getDefaultDirectoryScheme**() Returns the project's default directory scheme with regard to packages.
IRPCollaboration	**getNewCollaboration**() Creates a new IRPCollaboration object that can be used to create a sequence diagram.
IRPProgressBar	**getNewProgressBar**(int amount, java.lang.String name) method getNewProgressBar
int	**getNotifyPluginOnElementsChanged**() Checks whether plugins will be notified when model elements are modified.
IRPCollection	**getProfiles**() Returns a collection of all the profiles in the project.
IRPCollection	**getRemoteResourcePackages**() For Design Manager projects, returns the packages of remote requirements (the "Remote Resource Packages").
IRPCollection	**getRequirementsByID**(java.lang.String requirementID, int returnFirstFoundOnly) Returns all of the requirements that have the specified ID.
IRPProject	**getRoundtripShadowModel**(int magicNumber) For internal use only.
IRPModelElement	**highlightFromCode**(java.lang.String filename, int lineNumber) Highlights in the Rhapsody browser the model element associated with the

Method Summary

	specified line of code.
void	**importPackageFromRose**(java.lang.String projectName, java.lang.String packageName, java.lang.String logFileName) Imports the specified Rose package.
void	**importProjectFromRose**(java.lang.String projectName, java.lang.String logFileName) Imports the specified Rose project.
int	**isActivelyManaged**() Checks whether the project is an actively-managed Design Manager project.
int	**isModifiedRecursive**() Checks whether any part of the project has been modified and the project needs to be saved.
void	**locateInIDE**(IRPConfiguration config, java.lang.String filename, int lineNumber) For projects that use Rhapsody's integration with Eclipse or Visual Studio, you can use the locateInIDE method to have the IDE highlight a specific line in a specific source file.
void	**moveToDesignManager**(java.lang.String userName, java.lang.String password, java.lang.String serverURL, java.lang.String projectAreaName, java.lang.String streamName) Moves the model to a Rhapsody Design Manager server.
void	**moveToDesignManagerAfterLogin**(java.lang.String serverURL, java.lang.String projectAreaName, java.lang.String streamName) Moves the model to a Rhapsody Design Manager server.
void	**openCSVFile**(java.lang.String fullCSVFileName, int reserved) Displays the content of the specified csv file in a new tab in the Output window.
void	**recalculateEventsBaseIds**() If you are using Rational Rhapsody's default numbering scheme for event IDs, then a certain amount of IDs are reserved for each package.
void	**reloadCSVFile**(java.lang.String fullCSVFileName) Reloads the content of the specified csv file in a tab in the Output window.
int	**remove**() Removes the project from the Rhapsody workspace.
void	**removeCustomViewOnBrowser**(IRPPackage customView) Removes the specified custom view from the model browser.
void	**removeCustomViewOnDiagram**(IRPDiagram diagram, IRPPackage customView) Removes the specified custom view from the specified diagram.
void	**save**() Saves the project.
void	**saveAs**(java.lang.String filename) Saves the project using the specified path.

Method Summary

void	**saveAsPrevVersion**(java.lang.String filename, java.lang.String prevVersion) Saves the project, using the format of a previous version of Rhapsody.
void	**setActiveComponent**(IRPComponent activeComponent) Sets the specified component as the active component for the project.
void	**setActiveComponent**(java.lang.String name) Sets the specified component as the active component for the project.
void	**setActiveConfiguration**(IRPConfiguration activeConfiguration) Sets the specified configuration to be the active configuration of the project.
void	**setActiveConfiguration**(java.lang.String name) Sets the specified configuration to be the active configuration of the project.
void	**setDefaultDirectoryScheme**(java.lang.String defaultDirectoryScheme) Set's the project's default directory scheme with regard to packages.
void	**setNotifyPluginOnElementsChanged**(int val) For plugins that use the callback API, you must call the method setNotifyPluginOnElementsChanged if you want the plugin to be notified when model elements are modified.
void	**setObjectExplicit**(IRPInstance pInstance) Changes the specified object to an explicit object.
void	**setObjectImplicit**(IRPInstance pInstance) Changes the specified object to an implicit object.
void	**setUseUniqueStereotypeAndRefCache**(int useUniqueStereotypeAndRefCache) This method can be used to specify that all of the stereotypes in the model should be cached to allow quicker retrieval.
void	**setWaitDialogWatchdogValue**(java.lang.String value) The method setWaitDialogWatchdogValue provides a mechanism that allows an external process to inform Rhapsody that the process has ended or crashed.
void	**startTransactionOfNoCGInterest**() For internal use only.

Methods inherited from interface com.telelogic.rhapsody.core.IRPPackage

addActivityDiagram, addActor, addClass, addCollaborationDiagram, addComponentDiagram, addDeploymentDiagram, addEvent, addFlowItems, addFlows, addGlobalFunction, addGlobalObject, addGlobalVariable, addImplicitObject, addLink, addLinkBetweenSYSMLPorts, addModule, addNestedPackage, addNode, addObjectModelDiagram, addPanelDiagram, addSequenceDiagram, addStatechart, addTimingDiagram, addType, addUseCase, addUseCaseDiagram, deleteActor, deleteClass, deleteCollaborationDiagram, deleteComponentDiagram, deleteDeploymentDiagram, deleteEvent, deleteFlowItems, deleteFlows, deleteGlobalFunction, deleteGlobalObject, deleteGlobalVariable, deleteNode, deleteObjectModelDiagram, deletePackage, deletePanelDiagram, deleteSequenceDiagram, deleteTimingDiagram, deleteType, deleteUseCase, deleteUseCaseDiagram, findActor, findAllByName, findClass, findEvent, findGlobalFunction, findGlobalObject, findGlobalVariable, findNode, findType, findUsage, findUseCase, getActors, getAllNestedElements, getBehavioralDiagrams,

Methods inherited from interface com.telelogic.rhapsody.core.IRPPackage

getClasses, getCollaborationDiagrams, getComponentDiagrams, getDeploymentDiagrams, getEvents, getEventsBaseId, getFlowItems, getFlows, getGlobalFunctions, getGlobalObjects, getGlobalVariables, getInstanceSpecifications, getLinks, getModules, getNamespace, getNestedClassifiers, getNestedComponents, getNodes, getObjectModelDiagrams, getPackages, getPanelDiagrams, getRemoteRequirementsPopulateMode, getSavedInSeperateDirectory, getSequenceDiagrams, getSourceArtifacts, getTimingDiagrams, getTypes, getUseCaseDiagrams, getUseCases, getUserDefinedStereotypes, populateRemoteRequirements, reCalculateEventsBaseId, setRemoteRequirementsPopulateMode, setSavedInSeperateDirectory

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

gatewayExportToXML

```
void gatewayExportToXML(java.lang.String filename,
                        java.lang.String params)
```

For internal use only.

---

gatewayExportToXML2

```
void gatewayExportToXML2(java.lang.String filename,
                         java.lang.String params,
                         IRPProject proj)
```

For internal use only.

---

generateReport

```
void generateReport(java.lang.String modelscope,
                    java.lang.String templatename,
                    java.lang.String docType,
                    java.lang.String filename,
                    int showDocument,
                    int silentMode)
```

Generates a ReporterPLUS report for the model. (When this method is used to generate a report, the Rhapsody model is saved before the report is generated.)

```
// Sample code:
IRPProject currentProject = app.activeProject();
currentProject.generateReport("", "C:\Rhapsody\reporterplus\Templates\Class.tpl", "html
```

Parameters:

> modelscope - the name of the package for which the report should be generated. If empty, a report is generated for the entire model. (This is similar to the "scope" command-line option for ReporterPLUS.)
>
> templatename - the name of the template to use. If empty, then the ReporterPLUS report generation wizard will be launched and it will display the name of the last template used.
>
> docType - the type of output to generate (doc, html, ppt, txt). If empty, the ReporterPLUS report generation wizard will be launched and it will display the last output type used.
>
> filename - the filename to use for the generated report. If empty, the ReporterPLUS report generation wizard will be displayed and it will display the filename of the last generated report.
>
> showDocument - In general, the user will be asked if they want to view the report after generation only if they have requested this by selecting View > Options > Ask to open after generating report from the main menu in ReporterPLUS. However, if the user has specified silent generation mode using the parameter silentMode, this parameter can be used to request that the generated document be displayed. To display the report, set this parameter to 1, otherwise use 0.
>
> silentMode - If the template name, document type, or output file name has not been specified using the appropriate parameter, the ReporterPLUS report generation wizard is

displayed so the user can provide the missing information. This is the behavior if this parameter is set to 0. If you want to prevent the wizard from being launched in such cases, you can specify silent generation mode by setting this parameter to 1. If set to silent mode, no report will be generated if one or more of the above parameters was not provided. (The report generation status dialog is displayed regardless of the value of this parameter.)

addComponent

IRPComponent **addComponent**(java.lang.String name)

Adds a new Component to the project.
Parameters:
    name - the name to use for the new Component
Returns:
    the Component that was created

addCustomViewOnBrowser

void **addCustomViewOnBrowser**(IRPPackage customView)

Applies the specified custom view to the model browser.
Parameters:
    customView - the custom view that should be applied to the browser
Throws:
    RhapsodyRuntimeException

addCustomViewOnDiagram

void **addCustomViewOnDiagram**(IRPDiagram diagram,
                       IRPPackage customView)

Applies the specified custom view to the specified diagram.
Parameters:
    diagram - the diagram to which the custom view should be applied
    customView - the custom view that should be applied to the diagram
Throws:
    RhapsodyRuntimeException

addPackage

IRPPackage **addPackage**(java.lang.String name)

Adds a new package to the project.
Parameters:
    name - the name to use for the new package
Returns:

the package that was created

addProfile

[IRPProfile](#) **addProfile**(java.lang.String name)

Adds a new profile to the project.
Parameters:
    name - the name to use for the new profile
Returns:
    the profile that was created

addSpellCheckerResult

void **addSpellCheckerResult**(java.lang.String value)

For internal use only.

allowAutoSave

void **allowAutoSave**(int allow)

Can be used to temporarily disable autosaving of the model regardless of the current value of the property General::Model::AutoSaveInterval.
Parameters:
    allow - Use 0 to turn off autosave, use 1 to turn autosave on

allowNonUniqueNames

void **allowNonUniqueNames**(int allow)

For internal use only.

applyBrowserCustomViewsOnDiagrams

void **applyBrowserCustomViewsOnDiagrams**(int newVal)

Applies the custom views applied to the browser to all diagrams as well.
Parameters:
    newVal - Use 1 to apply the custom views to all diagrams, 0 to remove the custom views from the diagrams
Throws:
    [RhapsodyRuntimeException](#)

applyRoundtripDiffMerge

```
void applyRoundtripDiffMerge(int magicNumber,
                             IRPProject shadowModel,
                             IRPCollection filesToUpdate)
```

For internal use only.

---

becomeActiveProject

```
void becomeActiveProject()
```

Makes this project the active project in Rhapsody. For use when you have multiple projects open in Rhapsody.

---

checkEventsBaseIdsSolveCollisions

```
void checkEventsBaseIdsSolveCollisions()
```

Checks the values of the event base IDs for all packages in the model, detects collisions between the IDs, and resolves any incorrect values and collisions.

---

cleanUnresolvedElements

```
void cleanUnresolvedElements(IRPModelElement rootElement)
```

Removes any unresolved elements from the model, starting at the level of the specified element and working downward.
Parameters:
        rootElement - the element below which Rhapsody should remove all unresolved elements

---

close

```
void close()
```

Closes the project.

---

closeCSVFile

```
void closeCSVFile(java.lang.String fullCSVFileName)
```

Closes the tab in the Output window for the specified csv file.
Parameters:
        fullCSVFileName - the path of the csv file that should be closed

---

deleteComponent

void **deleteComponent**(IRPComponent component)

> Deletes the specified Component.
> **Parameters:**
> > component - the Component that should be deleted

---

endTransactionOfNoCGInterest

void **endTransactionOfNoCGInterest**()

> For internal use only.

---

findComponent

IRPComponent **findComponent**(java.lang.String name)

> Returns the Component with the specified name.
> **Parameters:**
> > name - the name of the Component to return
> **Returns:**
> > the Component with the specified name

---

findElementByBinaryID

IRPModelElement **findElementByBinaryID**(byte[] theID)

> Retrieves a model element based on its binary ID. This operation can be used in conjunction with the operation IRPModelElement.getBinaryID, which returns the binary ID of the element. In some situations, findElementByBinaryID is faster than the operation IRPProject.findElementByGUID.
> **Parameters:**
> > theID - the binary ID for the model element that should be retrieved
> **Returns:**
> > the model element with the specified binary ID

---

findElementByFileName

IRPModelElement **findElementByFileName**(java.lang.String theFolderName,
                                          java.lang.String theFileName)

> Returns the top-level element in the specified Rhapsody unit file. For example, the top-level package is returned for an sbs file, and the class element is returned for a cls file.
> **Parameters:**
> > theFolderName - the full path of the folder that contains the unit file
> > theFileName - the filename for the unit file

Returns:
> the top-level element in the specified Rhapsody unit file

Throws:
> RhapsodyRuntimeException

---

findElementByGUID

IRPModelElement **findElementByGUID**(java.lang.String theGUID)

> Retrieves a model element based on its GUID.
> **Parameters:**
>> `theGUID` - the GUID for the model element that should be retrieved
>
> **Returns:**
>> the model element with the specified GUID

---

findElementsWithOSLCLink

IRPCollection **findElementsWithOSLCLink**(java.lang.String type,
                                    java.lang.String purl)

> Returns a collection of all the model elements that have an OSLC link of the specified type to the specified target element.
> **Parameters:**
>> `type` - the link type of the OSLC link. Must be one of the typed defined in
>> IRPModelElement.OSLCLink.Types. You can also use "*" to represent all of the types.
>> `purl` - the URL for the link's target element. You can use "*" to find all the model elements that have any link of the specified type.
>
> **Returns:**
>> collection of the model elements that have an OSLC link of the specified type to the specified target element
>
> **Throws:**
>> RhapsodyRuntimeException

---

getActiveComponent

IRPComponent **getActiveComponent**()

> Returns the active component.
> **Returns:**
>> the active component

---

getActiveConfiguration

IRPConfiguration **getActiveConfiguration**()

> Returns the active configuration.

Returns:

the active configuration

getActiveCustomViewsOnBrowser

IRPCollection **getActiveCustomViewsOnBrowser**()

Returns a collection of the custom views currently applied to the browser.
Returns:

the custom views currently applied to the browser
Throws:

RhapsodyRuntimeException

getActiveCustomViewsOnDiagram

IRPCollection **getActiveCustomViewsOnDiagram**(IRPDiagram diagram)

Returns a collection of the custom views currently applied to the specified diagram.
Parameters:

diagram - the diagram whose custom views should be returned
Returns:

the custom views currently applied to the specified diagram
Throws:

RhapsodyRuntimeException

getAllStereotypes

IRPCollection **getAllStereotypes**()

Returns a collection of all the stereotypes in the project.
Returns:

all the stereotypes in the project

getCgSimplifiedModelPackage

IRPPackage **getCgSimplifiedModelPackage**()

Returns the package that contains the simplified model.
Returns:

the package that contains the simplified model

getCodeGeneratedFiles

IRPCollection **getCodeGeneratedFiles**()

getActiveConfiguration 491

Returns a collection of filenames for the code files that will be generated for the current active component if you select the "regenerate" option. Note that this does not refer to which files were actually generated the last time that code generation was carried out.

Returns:
> collection of filenames for the code files that will be generated for the current active component if you select the "regenerate" option

---

getComponents

[IRPCollection](#) **getComponents**()

> Returns a collection of all the components in the project.
> **Returns:**
>> all the components in the project

---

getDefaultDirectoryScheme

java.lang.String **getDefaultDirectoryScheme**()

> Returns the project's default directory scheme with regard to packages. "Flat" means that all new units are saved in a single directory. "PackageAsDirectory" means that a new directory is created for each package in the model. This setting is controlled by the property General::Model::DefaultDirectoryScheme.
> **Returns:**
>> the project's default directory scheme with regard to packages - Flat or PackageAsDirectory

---

getNewCollaboration

[IRPCollaboration](#) **getNewCollaboration**()

> Creates a new IRPCollaboration object that can be used to create a sequence diagram.
> **Returns:**
>> the IRPCollaboration object that was created

---

getNewProgressBar

[IRPProgressBar](#) **getNewProgressBar**(int amount,
                                java.lang.String name)

> method getNewProgressBar
> **Throws:**
>> [RhapsodyRuntimeException](#)

---

getNotifyPluginOnElementsChanged

int **getNotifyPluginOnElementsChanged**()

>Checks whether plugins will be notified when model elements are modified.
>**Returns:**
>>1 if plugins are to be notified when model elements are modified, 0 otherwise.
>**Throws:**
>>RhapsodyRuntimeException

getProfiles

IRPCollection **getProfiles**()

>Returns a collection of all the profiles in the project.
>**Returns:**
>>all the profiles in the project

getRemoteResourcePackages

IRPCollection **getRemoteResourcePackages**()

>For Design Manager projects, returns the packages of remote requirements (the "Remote Resource Packages"). The collection returned consists or IRPPackage objects. You can then cycle through the individual packages to access the individual requirements.
>**Returns:**
>>the packages of remote requirements

getRequirementsByID

IRPCollection **getRequirementsByID**(java.lang.String requirementID,
                                     int returnFirstFoundOnly)

>Returns all of the requirements that have the specified ID. This refers to the ID field in the Features dialog for requirements. For models where you know that only one requirement can have a given ID, you can use the second parameter to specify that only the first matching requirement should be returned - this option can be useful in large models.
>**Parameters:**
>>requirementID - the ID to use for the search
>>returnFirstFoundOnly - use 1 if you want the method to return only the first matching requirement, 0 if you want the method to return all matching requirements
>**Returns:**
>>all of the requirements that have the specified ID
>**Throws:**
>>RhapsodyRuntimeException

getRoundtripShadowModel

IRPProject **getRoundtripShadowModel**(int magicNumber)

For internal use only.

---

highlightFromCode

IRPModelElement **highlightFromCode**(java.lang.String filename,
                                     int lineNumber)

Highlights in the Rhapsody browser the model element associated with the specified line of code.

```
 IRPProject prj = app.openProject("d:\\temp\\_sample_code\\First_Project.rpy");
 prj.highlightFromCode("d:\\temp\\_sample_code\\DefaultComponent\\DefaultConfig\\class_0.h", 42);
```

Parameters:
> filename - the absolute path for the relevant source file
> lineNumber - the line number in the file

Returns:
> the model element associated with the specified line of code

Throws:
> RhapsodyRuntimeException

---

importPackageFromRose

void **importPackageFromRose**(java.lang.String projectName,
                              java.lang.String packageName,
                              java.lang.String logFileName)

Imports the specified Rose package. Beginning in release 8.1.4, this method is no longer supported.

Parameters:
> projectName - the Rose project from which the package should be imported (path that
> includes the name of the *.mdl file)
> packageName - the name of the Rose package to import
> logFileName - the file to use for logging the import process

---

importProjectFromRose

void **importProjectFromRose**(java.lang.String projectName,
                              java.lang.String logFileName)

Imports the specified Rose project. Beginning in release 8.1.4, this method is no longer supported.

Parameters:
> projectName - the Rose project to import (path that includes the name of the *.mdl file)
> logFileName - the file to use for logging the import process

---

isActivelyManaged

int **isActivelyManaged**()

Checks whether the project is an actively-managed Design Manager project.

Returns:

1 if the project is an actively-managed DM project, 0 otherwise

Throws:

[RhapsodyRuntimeException](#)

---

isModifiedRecursive

int **isModifiedRecursive**()

Checks whether any part of the project has been modified and the project needs to be saved.

Returns:

1 if any part of the project has been modified, 0 if no changes have been made

---

locateInIDE

```
void locateInIDE(IRPConfiguration config,
                 java.lang.String filename,
                 int lineNumber)
```

For projects that use Rhapsody's integration with Eclipse or Visual Studio, you can use the locateInIDE method to have the IDE highlight a specific line in a specific source file.

Parameters:

config - the Rhapsody configuration (of type Eclipse or Visual Studio configuration) that contains the generated source file

filename - the name of the file that should be opened in the IDE

lineNumber - the line number of the line that should be highlighted

Throws:

[RhapsodyRuntimeException](#)

---

moveToDesignManager

```
void moveToDesignManager(java.lang.String userName,
                         java.lang.String password,
                         java.lang.String serverURL,
                         java.lang.String projectAreaName,
                         java.lang.String streamName)
```

Moves the model to a Rhapsody Design Manager server. Note that this operation is only intended for the initial move of a model to the DM server. It cannot be used to replace an existing model on the server.

Parameters:

userName - the username for logging in to the Design Manager server

password - the password for logging in to the Design Manager server

serverURL - the URL of the Design Manager server

projectAreaName - the name of the Project Area to use for this model

streamName - the name of the Stream to use for this model

moveToDesignManagerAfterLogin

```
void moveToDesignManagerAfterLogin(java.lang.String serverURL,
                                   java.lang.String projectAreaName,
                                   java.lang.String streamName)
```

Moves the model to a Rhapsody Design Manager server. Before calling this method, you must call one of the DM login methods in IRPApplication, such as loginToDesignManagerWithUsername. Note that this operation is only intended for the initial move of a model to the DM server. It cannot be used to replace an existing model on the server.

Parameters:

serverURL - the URL of the Design Manager server

projectAreaName - the name of the Project Area to use for this model

streamName - the name of the Stream to use for this model

Throws:

RhapsodyRuntimeException

openCSVFile

```
void openCSVFile(java.lang.String fullCSVFileName,
                 int reserved)
```

Displays the content of the specified csv file in a new tab in the Output window.

Parameters:

fullCSVFileName - the path of the csv file that should be displayed

reserved - this parameter has no effect, you can use any integer

recalculateEventsBaseIds

```
void recalculateEventsBaseIds()
```

If you are using Rational Rhapsody's default numbering scheme for event IDs, then a certain amount of IDs are reserved for each package. As a result, there are situations where the IDs used for events in a given package may not be continuous. In cases like this, you can use the method recalculateEventsBaseIds to have the event ID numbering recalculated so that event IDs are continuous for all events within each package in the project.

reloadCSVFile

```
void reloadCSVFile(java.lang.String fullCSVFileName)
```

Reloads the content of the specified csv file in a tab in the Output window.

Parameters:

`fullCSVFileName` - the path of the csv file that should be reloaded

---

remove

`int` **`remove`**`()`

Removes the project from the Rhapsody workspace.

Returns:

1 if the project was removed successfully, 0 otherwise

Throws:

RhapsodyRuntimeException

---

removeCustomViewOnBrowser

`void` **`removeCustomViewOnBrowser`**`(`IRPPackage` customView)`

Removes the specified custom view from the model browser.

Parameters:

`customView` - the custom view that should be removed from the browser

Throws:

RhapsodyRuntimeException

---

removeCustomViewOnDiagram

`void` **`removeCustomViewOnDiagram`**`(`IRPDiagram` diagram,`
                                  IRPPackage` customView)`

Removes the specified custom view from the specified diagram.

Parameters:

`diagram` - the diagram that the custom view should be removed from

`customView` - the custom view that should be removed from the diagram

Throws:

RhapsodyRuntimeException

---

save

`void` **`save`**`()`

Saves the project.

---

saveAs

`void` **`saveAs`**`(java.lang.String filename)`

Saves the project using the specified path.

Parameters:
>   `filename` - the path to use for saving the project

saveAsPrevVersion

```
void saveAsPrevVersion(java.lang.String filename,
                       java.lang.String prevVersion)
```

Saves the project, using the format of a previous version of Rhapsody.
Parameters:
>   `filename` - the path to use for saving the project
>   `prevVersion` - the Rhapsody version whose format you want to use for saving the project. The valid strings for this parameter are those that are used in the Save As dialog in the user interface, for example, "7.6.1".

setActiveComponent

```
void setActiveComponent(java.lang.String name)
```

Sets the specified component as the active component for the project. Note that there are two versions of this method. The first takes a String parameter, which is the name of the component that should be made the active component. The second version takes an object of type IRPComponent, which is the component that should be made the active component.
Parameters:
>   `activeComponent` - the name of the component that should be set as the active component for the project. The string should represent the location of the component in the project hierarchy, using a double colon (::) as the delimiter, for example, Default::subpackage_1::component_a::subcomponent_b. If the component is not contained in a package or within another component, you can just use the name of the component.

setActiveComponent

```
void setActiveComponent(IRPComponent activeComponent)
```

Sets the specified component as the active component for the project. Note that there are two versions of this method. The first takes a String parameter, which is the name of the component that should be made the active component. The second version takes an object of type IRPComponent, which is the component that should be made the active component.
Parameters:
>   `activeComponent` - the name of the component that should be set as the active component for the project. The string should represent the location of the component in the project hierarchy, using a double colon (::) as the delimiter, for example, Default::subpackage_1::component_a::subcomponent_b. If the component is not contained in a package or within another component, you can just use the name of the component.

setActiveConfiguration

void **setActiveConfiguration**(java.lang.String name)

Sets the specified configuration to be the active configuration of the project. The configuration must belong to the active component. Note that there are two versions of this method. The first takes a String parameter, which is the name of the configuration that should be made the active configuration. The second version takes an object of type IRPConfiguration, which is the configuration that should be made the active configuration.

Parameters:

activeConfiguration - the name of the configuration to set as the active configuration

setActiveConfiguration

void **setActiveConfiguration**(IRPConfiguration activeConfiguration)

Sets the specified configuration to be the active configuration of the project. The configuration must belong to the active component. Note that there are two versions of this method. The first takes a String parameter, which is the name of the configuration that should be made the active configuration. The second version takes an object of type IRPConfiguration, which is the configuration that should be made the active configuration.

Parameters:

activeConfiguration - the name of the configuration to set as the active configuration

setDefaultDirectoryScheme

void **setDefaultDirectoryScheme**(java.lang.String defaultDirectoryScheme)

Set's the project's default directory scheme with regard to packages. This setting is controlled by the property General::Model::DefaultDirectoryScheme.

Parameters:

defaultDirectoryScheme - the default directory scheme to use for packages in the model. The valid values for this parameter are: "Flat" - all new units are saved in a single directory, and "PackageAsDirectory" - a new directory is created for each package in the model.

setNotifyPluginOnElementsChanged

void **setNotifyPluginOnElementsChanged**(int val)

For plugins that use the callback API, you must call the method setNotifyPluginOnElementsChanged if you want the plugin to be notified when model elements are modified.

Parameters:

val - Use 1 to specify that the plugin should be notified when an element is modified. Use 0 to specify that the plugin should not be notified when elements are modified.

Throws:

RhapsodyRuntimeException

setObjectExplicit

void **setObjectExplicit**([IRPInstance](#) pInstance)

Changes the specified object to an explicit object. As a result, a class is added to the model with the name [object name]_Class. This method corresponds to the Expose Class option in the pop-up menu of the Rhapsody model browser.
Parameters:
pInstance - the object that should be changed to explicit

setObjectImplicit

void **setObjectImplicit**([IRPInstance](#) pInstance)

Changes the specified object to an implicit object.
Parameters:
pInstance - the object that should be changed to implicit

setUseUniqueStereotypeAndRefCache

void **setUseUniqueStereotypeAndRefCache**(int useUniqueStereotypeAndRefCache)

This method can be used to specify that all of the stereotypes in the model should be cached to allow quicker retrieval. This can be beneficial when working with profiles that contain a very large number of stereotypes.
Parameters:
useUniqueStereotypeAndRefCache - use 1 to turn on stereotype caching, use 0 to turn off stereotype caching
Throws:
[RhapsodyRuntimeException](#)

setWaitDialogWatchdogValue

void **setWaitDialogWatchdogValue**(java.lang.String value)

The method setWaitDialogWatchdogValue provides a mechanism that allows an external process to inform Rhapsody that the process has ended or crashed. Call this method with any non-blank value to notify Rhapsody that the process is running. Rhapsody then displays a message dialog indicating that it is waiting for the process to complete. The user plugin must continue calling this method repeatedly to indicate that it has not finished. The interval for calling the function must be less than two minutes. If the method is not called for two minutes, Rhapsody assumes the process has crashed, and it closes the dialog and lets Rhapsody continue. When the process has completed, call the method again with an empty string as the argument. This informs Rhapsody that the process is done.
Parameters:
value - use non-blank string to indicate to Rhapsody that the external process is still running, use blank string to indicate to Rhapsody that the process has completed
Throws:

RhapsodyRuntimeException

startTransactionOfNoCGInterest

void **startTransactionOfNoCGInterest**()

> For internal use only.

com.telelogic.rhapsody.core
Interface IRPRelation

All Superinterfaces:
　　　IRPModelElement, IRPUnit

All Known Subinterfaces:
　　　IRPInstance, IRPModule, IRPPort, IRPSysMLPort

---

public interface **IRPRelation**
extends IRPUnit

Represents a relationship between two classes.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

void	**addQualifier**(IRPModelElement pVal) 　　　method addQualifier
IRPAssociationClass	**getAssociationClass**() 　　　method getAssociationClass
IRPRelation	**getInverse**() 　　　get property inverse
int	**getIsNavigable**() 　　　get property isNavigable
int	**getIsSymmetric**() 　　　get property isSymmetric
java.lang.String	**getMultiplicity**() 　　　get property multiplicity
IRPClass	**getObjectAsObjectType**() 　　　get property ObjectAsObjectType

Method Summary

[IRPClassifier](#)	**getOfClass**() get property ofClass
[IRPClassifier](#)	**getOtherClass**() Gets the class that this class is related to via this relation.
java.lang.String	**getQualifier**() get property qualifier
[IRPCollection](#)	**getQualifiers**() method getQualifiers
[IRPClassifier](#)	**getQualifierType**() For associations that use qualifiers, returns the type of the qualifier.
java.lang.String	**getRelationLabel**() get property relationLabel
java.lang.String	**getRelationLinkName**() get property relationLinkName
java.lang.String	**getRelationRoleName**() get property relationRoleName
java.lang.String	**getRelationType**() get property relationType
java.lang.String	**getVisibility**() get property visibility
int	**isTypelessObject**() method isTypelessObject
void	**makeUnidirect**() method makeUnidirect
void	**removeQualifier**([IRPModelElement](#) pVal) method removeQualifier
void	**setInverse**(java.lang.String roleName, java.lang.String linkType) property setInverse
void	**setIsNavigable**(int isNavigable) set property isNavigable
void	**setMultiplicity**(java.lang.String multiplicity) set property multiplicity
void	**setOfClass**([IRPClassifier](#) ofClass) set property ofClass
void	**setOtherClass**([IRPClassifier](#) otherClass) set property otherClass
void	**setQualifier**(java.lang.String qualifier) set property qualifier

Method Summary

void	**setQualifierType**(IRPClassifier pVal) Sets the type to use for the qualifier for the association.
void	**setRelationLabel**(java.lang.String relationLabel) set property relationLabel
void	**setRelationLinkName**(java.lang.String relationLinkName) set property relationLinkName
void	**setRelationRoleName**(java.lang.String relationRoleName) set property relationRoleName
void	**setRelationType**(java.lang.String relationType) set property relationType

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addQualifier

void **addQualifier**(IRPModelElement pVal)

> method addQualifier
> **Throws:**
> > RhapsodyRuntimeException

---

getAssociationClass

IRPAssociationClass **getAssociationClass**()

> method getAssociationClass
> **Throws:**
> > RhapsodyRuntimeException

---

getInverse

IRPRelation **getInverse**()

> get property inverse
> **Throws:**
> > RhapsodyRuntimeException

---

getIsNavigable

int **getIsNavigable**()

> get property isNavigable
> **Throws:**
> > RhapsodyRuntimeException

---

getIsSymmetric

int **getIsSymmetric**()

> get property isSymmetric
> **Throws:**
> > RhapsodyRuntimeException

---

getMultiplicity

java.lang.String **getMultiplicity**()

>   get property multiplicity
>   **Throws:**
>   RhapsodyRuntimeException

getObjectAsObjectType

IRPClass **getObjectAsObjectType**()

>   get property ObjectAsObjectType
>   **Throws:**
>   RhapsodyRuntimeException

getOfClass

IRPClassifier **getOfClass**()

>   get property ofClass
>   **Throws:**
>   RhapsodyRuntimeException

getOtherClass

IRPClassifier **getOtherClass**()

>   Gets the class that this class is related to via this relation.
>   **Returns:**
>   the class that this class is related to via this relation

getQualifier

java.lang.String **getQualifier**()

>   get property qualifier
>   **Throws:**
>   RhapsodyRuntimeException

getQualifierType

IRPClassifier **getQualifierType**()

>   For associations that use qualifiers, returns the type of the qualifier.

getMultiplicity   506

Returns:
the type of the qualifier that is used for the association

getQualifiers

[IRPCollection](#) **getQualifiers**()

method getQualifiers
Throws:
[RhapsodyRuntimeException](#)

getRelationLabel

java.lang.String **getRelationLabel**()

get property relationLabel
Throws:
[RhapsodyRuntimeException](#)

getRelationLinkName

java.lang.String **getRelationLinkName**()

get property relationLinkName
Throws:
[RhapsodyRuntimeException](#)

getRelationRoleName

java.lang.String **getRelationRoleName**()

get property relationRoleName
Throws:
[RhapsodyRuntimeException](#)

getRelationType

java.lang.String **getRelationType**()

get property relationType
Throws:
[RhapsodyRuntimeException](#)

getVisibility

java.lang.String **getVisibility**()

> get property visibility
> **Throws:**
>> RhapsodyRuntimeException

isTypelessObject

int **isTypelessObject**()

> method isTypelessObject
> **Throws:**
>> RhapsodyRuntimeException

makeUnidirect

void **makeUnidirect**()

> method makeUnidirect
> **Throws:**
>> RhapsodyRuntimeException

removeQualifier

void **removeQualifier**(IRPModelElement pVal)

> method removeQualifier
> **Throws:**
>> RhapsodyRuntimeException

setInverse

void **setInverse**(java.lang.String roleName,
                java.lang.String linkType)

> property setInverse
> **Throws:**
>> RhapsodyRuntimeException

setIsNavigable

void **setIsNavigable**(int isNavigable)

> set property isNavigable

Throws:

[RhapsodyRuntimeException](#)

setMultiplicity

void **setMultiplicity**(java.lang.String multiplicity)

set property multiplicity
Throws:

[RhapsodyRuntimeException](#)

setOfClass

void **setOfClass**([IRPClassifier](#) ofClass)

set property ofClass
Throws:

[RhapsodyRuntimeException](#)

setOtherClass

void **setOtherClass**([IRPClassifier](#) otherClass)

set property otherClass
Throws:

[RhapsodyRuntimeException](#)

setQualifier

void **setQualifier**(java.lang.String qualifier)

set property qualifier
Throws:

[RhapsodyRuntimeException](#)

setQualifierType

void **setQualifierType**([IRPClassifier](#) pVal)

Sets the type to use for the qualifier for the association.
Parameters:

pVal - the type to use for the qualifier for the association

setRelationLabel

void **setRelationLabel**(java.lang.String relationLabel)

> set property relationLabel
> **Throws:**
> > [RhapsodyRuntimeException](#)

setRelationLinkName

void **setRelationLinkName**(java.lang.String relationLinkName)

> set property relationLinkName
> **Throws:**
> > [RhapsodyRuntimeException](#)

setRelationRoleName

void **setRelationRoleName**(java.lang.String relationRoleName)

> set property relationRoleName
> **Throws:**
> > [RhapsodyRuntimeException](#)

setRelationType

void **setRelationType**(java.lang.String relationType)

> set property relationType
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

com.telelogic.rhapsody.core
Interface IRPRequirement

All Superinterfaces:
    IRPAnnotation, IRPModelElement, IRPUnit

---

public interface **IRPRequirement**
extends IRPAnnotation

The IRPRequirement interface represents requirements in a Rational Rhapsody model.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement

IRPModelElement.OSLCLink

Method Summary

java.lang.String	**getRequirementID**() Returns the ID that was set for the requirement.
void	**setRequirementID**(java.lang.String requirementID) Sets the ID for the requirement.

Methods inherited from interface com.telelogic.rhapsody.core.IRPAnnotation

addAnchor, getAnchoredByMe, getBody, getSpecification, getSpecificationRTF, isSpecificationRTF, removeAnchor, setBody, setSpecification, setSpecificationRTF

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

setRelationType   511

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getRequirementID

```
java.lang.String getRequirementID()
```

> Returns the ID that was set for the requirement.
> **Returns:**
> > the ID for the requirement

---

setRequirementID

```
void setRequirementID(java.lang.String requirementID)
```

> Sets the ID for the requirement.
> **Parameters:**
> > requirementID - the ID to use for the requirement

---

com.telelogic.rhapsody.core

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

 com.telelogic.rhapsody.core
Interface IRPRhapsodyServer

```
public interface IRPRhapsodyServer
```

Method Summary

IRPApplication	**getApplication**() getApplication
IRPApplication	**getHiddenApplication**() getHiddenApplication
java.lang.String	**getInterfaceName**() get property interfaceName
IRPApplication	**getUninitializedApplication**() getUninitializedApplication
void	**initializeApplication**(IRPApplication pVal) initializeApplication

Method Detail

getApplication

IRPApplication **getApplication**()

> getApplication
> **Throws:**
>   RhapsodyRuntimeException

---

getHiddenApplication

IRPApplication **getHiddenApplication**()

> getHiddenApplication
> **Throws:**
>   RhapsodyRuntimeException

getInterfaceName

java.lang.String **getInterfaceName**()

> get property interfaceName
> **Throws:**
> > RhapsodyRuntimeException

getUninitializedApplication

IRPApplication **getUninitializedApplication**()

> getUninitializedApplication
> **Throws:**
> > RhapsodyRuntimeException

initializeApplication

void **initializeApplication**(IRPApplication pVal)

> initializeApplication
> **Throws:**
> > RhapsodyRuntimeException

com.telelogic.rhapsody.core
Interface IRPRoundTrip

public interface **IRPRoundTrip**

Method Summary

java.lang.String	**getInterfaceName**() get property interfaceName
IRPCollection	**roundtripFile**(java.lang.String filename, int reGenerateFile) roundtrip file

Method Detail

getInterfaceName

java.lang.String **getInterfaceName**()

> get property interfaceName
> **Throws:**
> > RhapsodyRuntimeException

roundtripFile

IRPCollection **roundtripFile**(java.lang.String filename,
                            int reGenerateFile)

> roundtrip file
> **Throws:**
> > RhapsodyRuntimeException

com.telelogic.rhapsody.core
Interface IRPSearchManager

public interface **IRPSearchManager**

Method Summary

IRPSearchQuery	**createSearchQuery**() method Create search query
java.lang.String	**getInterfaceName**() get property interfaceName
IRPCollection	**search**(IRPSearchQuery pSearchQuery) method Search
void	**searchAndShowResults**(IRPSearchQuery pSearchQuery) method SearchAndShowResults
void	**searchAsync**(IRPSearchQuery pSearchQuery) method SearchAsync

Method Detail

createSearchQuery

IRPSearchQuery **createSearchQuery**()

> method Create search query
> **Throws:**
> > RhapsodyRuntimeException

---

search

IRPCollection **search**(IRPSearchQuery pSearchQuery)

> method Search
> **Throws:**
> > RhapsodyRuntimeException

searchAndShowResults

void **searchAndShowResults**([IRPSearchQuery](#) pSearchQuery)

> method SearchAndShowResults
> **Throws:**
> > [RhapsodyRuntimeException](#)

searchAsync

void **searchAsync**([IRPSearchQuery](#) pSearchQuery)

> method SearchAsync
> **Throws:**
> > [RhapsodyRuntimeException](#)

getInterfaceName

java.lang.String **getInterfaceName**()

> get property interfaceName
> **Throws:**
> > [RhapsodyRuntimeException](#)

[Package](#) Class [Use](#) [Tree](#) [Serialized](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)   **[FRAMES](#) [NO FRAMES](#) [All Classes](#)**
SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#)   DETAIL: FIELD | CONSTR | [METHOD](#)

search   518

com.telelogic.rhapsody.core
Interface IRPSearchQuery

public interface **IRPSearchQuery**

The IRPSearchQuery interface represents the search criteria objects that are used by IRPSearchManager to carry out searches.

Nested Class Summary

static class	**IRPSearchQuery.References**
static class	**IRPSearchQuery.SearchInField** Constant values used with elements of this type
static class	**IRPSearchQuery.SubQueriesOperator**
static class	**IRPSearchQuery.UnresolvedKind**
static class	**IRPSearchQuery.ViewsToSearch**

Method Summary

int	**addDiagramToViewsList**(IRPDiagram view) Adds the specified diagram to the list of views to be searched for the search text.
void	**addFilterElementType**(java.lang.String elementType) Adds an element type to the list of element types that the search should be applied to.
void	**addFilterSearchInField**(java.lang.String searchInField) Adds an element field to the list of element fields that the search should be applied to, for example, element name or element description.
void	**addFilterStereotype**(IRPStereotype stereotype) Specifies that the search should be limited to model elements with a specific stereotype applied to them.
void	**addFilterSubQuery**(IRPTableLayout subQuery, int useWithNotOperator) Adds a subquery to the list of subqueries specified for the search.

getInterfaceName   519

Method Summary

int	**addMatrixToViewsList**(IRPMatrixView view) Adds the specified matrix to the list of views to be searched for the search text.
void	**addSearchScope**(IRPModelElement scopeElement) Adds an element to the scope for the search.
int	**addTableToViewsList**(IRPTableView view) Adds the specified table to the list of views to be searched for the search text.
IRPCollection	**getFilterElementTypes**() Returns the element types that are to be searched for the search text.
int	**getFilterReferenceIncludeReferencedElementsInSearchResults**() Checks whether the reference search criterion specified that the referenced elements included in the search criterion should also be displayed in the search results.
java.lang.String	**getFilterReferenceNameOfReferencedElements**() Returns the model element name that was specified for the reference criterion that was defined.
int	**getFilterReferenceNumberOfReferences**() Returns the number of references that was specified as a search criterion.
java.lang.String	**getFilterReferenceQuantityOperator**() When the search criterion includes a specific number of references, this method returns a value that indicates whether the criterion was exactly that number of references, less than that number, or more than that number.
java.lang.String	**getFilterReferenceRelationKind**() Returns the type of reference used in the search criterion, for example, aggregates or incoming relations.
java.lang.String	**getFilterReferenceStereotypeOfReferencedElements**() Returns the stereotype that was specified for the reference criterion that was defined.
java.lang.String	**getFilterReferenceTypeOfReferencedElements**() Returns the model element type that was specified for the reference criterion that was defined.
IRPCollection	**getFilterSearchInFields**() Returns the list of element fields that the search is to be applied to.
IRPCollection	**getFilterStereotypes**() Returns the names of the stereotypes that were specified as search criteria.
IRPCollection	**getFilterSubQueries**() Returns the subqueries that were specified for the search.
java.lang.String	**getFilterSubQueriesOperator**() Returns indication of how the specified subqueries are to be combined in the search
int	**getFilterSubQueryUseWithNotOperator**(IRPTableLayout subQuery) Checks whether the NOT operator was specified for the specified subquery.

Method Summary

char	**getFilterTagFindAs**() Returns the type of search that was specified for the tag name and tag value search criteria - regular text, wildcard, regular expression, or empty string.
int	**getFilterTagMatchCase**() Checks whether an exact match was specified for the tag name and tag value search criteria, in terms of upper and lower case.
int	**getFilterTagMatchWholeWord**() Checks whether a whole word match was specified for the tag name and tag value search criteria
java.lang.String	**getFilterTagName**() Returns the tag name specified as a criterion for the search
java.lang.String	**getFilterTagValue**() Returns the tag value specified as a criterion for the search
int	**getFilterUnitsOnly**() Checks whether the search is limited to model elements that are saved units.
java.lang.String	**getFilterUnresolvedKind**() Returns the method that was specified for handling unresolved elements in the search.
int	**getIncludeDescendants**() Checks whether the scope of the search is to include the descendants of the elements specified for the scope.
java.lang.String	**getInterfaceName**() Returns the name of the interface (IRPSearchQuery).
int	**getMatchCase**() Checks whether an exact match was specified for the query in terms of upper and lower case.
int	**getMatchSpecifiedCriteria**() Checks whether the query is to return the model elements that match the criteria specified, or the model elements that do not match the criteria specified.
int	**getMatchWholeWord**() Checks whether a whole word match was specified for the search.
char	**getSearchFindAsOption**() Returns the type of search that was specified for the search text - regular text, wildcard, regular expression, or empty string.
IRPCollection	**getSearchScopeElements**() Returns a collection of the model elements that constitute the scope for the search.
IRPModelElement	**getSearchScopeObject**() **Deprecated.** *This method, used to return the scope specified for the search, was introduced when Rhapsody only allowed you to specify a single element as the scope. Now that Rhapsody allows you to specify a list of such elements, you should use the method* *getSearchScopeElements()*.

Method Summary

java.lang.String	**getSearchText**() Returns the text that was specified as the text to search for.
[IRPModelElement](#)	**getView**(int Index) Retrieves the specified item from the list of tables, matrices, and diagrams that are to be searched.
int	**getViewCount**() Returns the number of views in the list of views that are to be searched.
int	**getViewIncludeModelElements**() Checks whether the query specifies that the search results should also include model elements that were found by the search but are not referenced in any of the views that you specified.
java.lang.String	**getViewsToSearch**() Returns indication of which views (diagrams, tables, and matrices) are supposed to be searched.
void	**loadFromQuery**([IRPTableLayout](#) query) Loads the settings from the specified query into the search query object.
void	**removeFilterElementTypes**() Removes any element type filters that you defined to limit the search to certain element types.
void	**removeFilterReferences**() Removes reference search criterion that was defined for the search query.
void	**removeFilterSearchInFields**() Removes any element field filters that you defined to limit the search to certain element fields, for example, model element descriptions.
void	**removeFilterStereotypes**() Removes any stereotype filter that was defined to limit the search to model elements that have certain stereotypes applied to them.
void	**removeFilterSubQueries**() Removes the subquery criteria that were specified for the search.
int	**removeFilterSubQuery**([IRPTableLayout](#) subQuery) Removes the specified subquery from the search.
void	**removeFilterTag**() Removes the tag name and tag value criteria that were defined for the search query.
int	**removeSearchScopeElement**([IRPModelElement](#) scopeElement) Removes the specified model element from the scope for the search.
void	**removeView**(int Index) Removes the specified view from the list of views to be searched for the search text.
void	**resetSearchScope**()

Method Summary

	Resets the search scope to include the entire project, or all projects if multiple projects are open.
IRPTableLayout	**saveAsQuery**(IRPPackage queryOwner) Saves the search query object that you defined as a query in your model.
void	**setFilterReference**(java.lang.String quantityOperator, int numberOfReferences, java.lang.String relationKind, java.lang.String typeOfReferencedElements, java.lang.String stereotypeOfReferencedElements, java.lang.String nameOfReferencedElements, int includeReferencedElementsInSearchResults) Sets criteria for the search based on an element's references.
void	**setFilterSubQueriesOperator**(java.lang.String filterSubQueriesOperator) Specify how the various subqueries specified should be combined - as an AND operation or an OR operation
void	**setFilterTag**(java.lang.String tagName, java.lang.String tagValue, int matchCase, int matchWholeWord, char findAs) Sets tag name and tag value criteria for the search query.
void	**setFilterUnitsOnly**(int filterUnitsOnly) Specifies whether the search should be limited to model elements that are saved units.
void	**setFilterUnresolvedKind**(java.lang.String filterUnresolvedKind) Specifies how unresolved elements should be handled in the search.
void	**setIncludeDescendants**(int includeDescendants) Specifies whether the scope for the search should include the descendants of the elements specified for the scope, for example, the subpackages of a package that was added to the scope.
void	**setMatchCase**(int matchCase) Specifies whether the search should require an exact match in terms of upper and lower case.
void	**setMatchSpecifiedCriteria**(int matchSpecifiedCriteria) Specifies whether the query should return the model elements that match the criteria specified, or the model elements that do not match the criteria specified.
void	**setMatchWholeWord**(int matchWholeWord) Specifies whether the search should require whole word matches.
void	**setSearchFindAsOption**(char searchFindAsOption) Sets the type of search that should be used for the search text - regular text, wildcard, regular expression, or empty string.
void	**setSearchScopeObject**(IRPModelElement searchScopeObject) **Deprecated.** *This method, used to set the scope for the search, was introduced when Rhapsody only allowed you to specify a single element as the scope. Now that Rhapsody allows you to specify a list of such elements, you should use the method addSearchScope(com.telelogic.rhapsody.core.IRPModelElement).*
void	

Method Summary

	setSearchText(java.lang.String searchText) Specifies the text that should be searched for.
void	**setViewIncludeModelElements**(int viewIncludeModelElements) Specifies whether the search results should also include model elements that were found by the search but are not referenced in any of the views that you specified.
void	**setViewsToSearch**(java.lang.String viewsToSearch) Specifies which views (tables, matrices, and diagrams) should be searched - all, none, all open, or just the views that were specified with the methods addDiagramToViewsList, addTableToViewsList, and addMatrixToViewsList.

Method Detail

addDiagramToViewsList

int **addDiagramToViewsList**(IRPDiagram view)

Adds the specified diagram to the list of views to be searched for the search text. Note that the list of views to search will be used only if you call the method IRPSearchQuery.setViewsToSearch, providing IRPSearchQuery.ViewsToSearch.DETAILED as the argument.

Parameters:
        view - the diagram to add to the list of views to search
Returns:
        the location of the new item in the list of views. Note that the order of the list can change when a view is added or removed, so the returned index can only be used if you have not made additional changes to the list since adding the item.
Throws:
        RhapsodyRuntimeException

---

addFilterElementType

void **addFilterElementType**(java.lang.String elementType)

Adds an element type to the list of element types that the search should be applied to. Note that the purpose of this method is to limit the search to certain element types. If you do not call this method at all, then the search will be applied to all model element types.

Parameters:
        elementType - element type to add to the list of element types to search. The strings to use for this parameter can be found in the file metaclasses.txt in the Doc directory of the Rhapsody installation. For this parameter, you can also use any "new terms" in your project.
Throws:
        RhapsodyRuntimeException

---

addFilterSearchInField

void **addFilterSearchInField**(java.lang.String searchInField)

Adds an element field to the list of element fields that the search should be applied to, for example, element name or element description. Note that the purpose of this method is to limit the search to certain element fields. If you do not call this method at all, then the search will be applied to all model element fields.

Parameters:

searchInField - element field to add to the list of element fields to search. The value of this parameter should be one of the constants defined in the class IRPSearchQuery.SearchInField. For example, use IRPSearchQuery.SearchInField.NAME for the name of the model element, and IRPSearchQuery.SearchInField.DESCRIPTION for the description of the model element.

addFilterStereotype

void **addFilterStereotype**(IRPStereotype stereotype)

Specifies that the search should be limited to model elements with a specific stereotype applied to them. Note that you can call this method multiple times to specify that the search should be limited to elements that have a certain group of stereotypes.

Parameters:

stereotype - the stereotype to use as a search criterion. Use null if you want to search for model elements that do not have any stereotypes applied to them.

Throws:

RhapsodyRuntimeException

addFilterSubQuery

void **addFilterSubQuery**(IRPTableLayout subQuery,
                     int useWithNotOperator)

Adds a subquery to the list of subqueries specified for the search.

Parameters:

subQuery - the subquery to add for the search
useWithNotOperator - use 1 if you want the NOT operator to be used for the specified subquery, 0 otherwise

Throws:

RhapsodyRuntimeException

addMatrixToViewsList

int **addMatrixToViewsList**(IRPMatrixView view)

Adds the specified matrix to the list of views to be searched for the search text. Note that the list of views to search will be used only if you call the method IRPSearchQuery.setViewsToSearch, providing IRPSearchQuery.ViewsToSearch.DETAILED as the argument.

Parameters:

> `view` - the matrix to add to the list of views to search

Returns:

> the location of the new item in the list of views. Note that the order of the list can change when a view is added or removed, so the returned index can only be used if you have not made additional changes to the list since adding the item.

Throws:

addSearchScope

void **addSearchScope**(IRPModelElement scopeElement)

Adds an element to the scope for the search. You can call this method multiple times to include different parts of the model in a search.

Parameters:

> `scopeElement` - model element that represents a part of the model that should be searched, for example, a specific package

Throws:

> RhapsodyRuntimeException

addTableToViewsList

int **addTableToViewsList**(IRPTableView view)

Adds the specified table to the list of views to be searched for the search text. Note that the list of views to search will be used only if you call the method IRPSearchQuery.setViewsToSearch, providing IRPSearchQuery.ViewsToSearch.DETAILED as the argument.

Parameters:

> `view` - the table to add to the list of views to search

Returns:

> the location of the new item in the list of views. Note that the order of the list can change when a view is added or removed, so the returned index can only be used if you have not made additional changes to the list since adding the item.

Throws:

> RhapsodyRuntimeException

getFilterElementTypes

IRPCollection **getFilterElementTypes**()

Returns the element types that are to be searched for the search text. Note that this method will return element types only if you used the method addFilterElementType to limit the search to certain element types. If you did not call the method addFilterElementType, then the search is applied to all element types, and getFilterElementType will return an empty collection.

Returns:

> the element types that are to be searched

Throws:

getFilterReferenceIncludeReferencedElementsInSearchResults

int **getFilterReferenceIncludeReferencedElementsInSearchResults**()

Checks whether the reference search criterion specified that the referenced elements included in the search criterion should also be displayed in the search results.

Returns:

1 if the reference search criterion specified that the referenced elements included in the search criterion should also be displayed in the search results, 0 otherwise

Throws:

RhapsodyRuntimeException

getFilterReferenceNameOfReferencedElements

java.lang.String **getFilterReferenceNameOfReferencedElements**()

Returns the model element name that was specified for the reference criterion that was defined.

Returns:

the model element name that was specified for the reference criterion that was defined

Throws:

RhapsodyRuntimeException

getFilterReferenceNumberOfReferences

int **getFilterReferenceNumberOfReferences**()

Returns the number of references that was specified as a search criterion.

Returns:

the number of references that was specified as a search criterion

Throws:

RhapsodyRuntimeException

getFilterReferenceQuantityOperator

java.lang.String **getFilterReferenceQuantityOperator**()

When the search criterion includes a specific number of references, this method returns a value that indicates whether the criterion was exactly that number of references, less than that number, or more than that number.

Returns:

value that indicates whether the search criterion was an exact number of references, less than a specific number of references, or more than a specific number of references. The value returned will be one of the values defined in
IRPSearchQuery.References.QuantityOperator.

Throws:

RhapsodyRuntimeException

getFilterReferenceRelationKind

java.lang.String **getFilterReferenceRelationKind**()

> Returns the type of reference used in the search criterion, for example, aggregates or incoming relations.
> **Returns:**
>> the type of reference used in the search criterion - one of the values defined in IRPSearchQuery.References.RelationKind.
> **Throws:**
>> RhapsodyRuntimeException

getFilterReferenceStereotypeOfReferencedElements

java.lang.String **getFilterReferenceStereotypeOfReferencedElements**()

> Returns the stereotype that was specified for the reference criterion that was defined.
> **Returns:**
>> the stereotype that was specified for the reference criterion that was defined
> **Throws:**
>> RhapsodyRuntimeException

getFilterReferenceTypeOfReferencedElements

java.lang.String **getFilterReferenceTypeOfReferencedElements**()

> Returns the model element type that was specified for the reference criterion that was defined.
> **Returns:**
>> the model element type that was specified for the reference criterion that was defined.
> **Throws:**
>> RhapsodyRuntimeException

getFilterSearchInFields

IRPCollection **getFilterSearchInFields**()

> Returns the list of element fields that the search is to be applied to.
> **Returns:**
>> the list of element fields that the search is to be applied to. The collection returned will consist of constants defined in the class IRPSearchQuery.SearchInField. For example, IRPSearchQuery.SearchInField.NAME for the name of the model element, and IRPSearchQuery.SearchInField.DESCRIPTION for the description of the model element.
> **Throws:**
>> RhapsodyRuntimeException

getFilterStereotypes

IRPCollection **getFilterStereotypes**()

> Returns the names of the stereotypes that were specified as search criteria.
> **Returns:**
>> the names of the stereotypes that were specified as search criteria
> **Throws:**
>> RhapsodyRuntimeException

getFilterSubQueries

IRPCollection **getFilterSubQueries**()

> Returns the subqueries that were specified for the search.
> **Returns:**
>> the subqueries that were specified for the search
> **Throws:**
>> RhapsodyRuntimeException

getFilterSubQueryUseWithNotOperator

int **getFilterSubQueryUseWithNotOperator**(IRPTableLayout subQuery)

> Checks whether the NOT operator was specified for the specified subquery.
> **Parameters:**
>> subQuery - the subquery to be checked
> **Returns:**
>> 1 if the NOT operator was specified for the subquery, 0 otherwise
> **Throws:**
>> RhapsodyRuntimeException

getFilterTagFindAs

char **getFilterTagFindAs**()

> Returns the type of search that was specified for the tag name and tag value search criteria - regular text, wildcard, regular expression, or empty string.
> **Returns:**
>> the type of search that was specified for the tag name and tag value search criteria - will be one of the constants defined in the class SearchFindAsEnum, for example SearchFindAsEnum.RP_SEARCH_WILDCARD for a wildcard search or SearchFindAsEnum.RP_SEARCH_REGEX for a regular expression search.
> **Throws:**
>> RhapsodyRuntimeException

getFilterTagMatchCase

int **getFilterTagMatchCase**()

Checks whether an exact match was specified for the tag name and tag value search criteria, in terms of upper and lower case.
Returns:
1 if an exact match was specified for the tag criteria in terms of upper and lower case, 0 otherwise
Throws:
RhapsodyRuntimeException

getFilterTagMatchWholeWord

int **getFilterTagMatchWholeWord**()

Checks whether a whole word match was specified for the tag name and tag value search criteria
Returns:
1 if whole word match was specified for the tag criteria, 0 otherwise
Throws:
RhapsodyRuntimeException

getFilterTagName

java.lang.String **getFilterTagName**()

Returns the tag name specified as a criterion for the search
Returns:
the tag name specified as a criterion for the search
Throws:
RhapsodyRuntimeException

getFilterTagValue

java.lang.String **getFilterTagValue**()

Returns the tag value specified as a criterion for the search
Returns:
the tag value specified as a criterion for the search
Throws:
RhapsodyRuntimeException

getSearchScopeElements

IRPCollection **getSearchScopeElements**()

Returns a collection of the model elements that constitute the scope for the search.

Returns:

the model elements that constitute the scope for the search

Throws:

RhapsodyRuntimeException

---

getView

IRPModelElement **getView**(int Index)

Retrieves the specified item from the list of tables, matrices, and diagrams that are to be searched.

Parameters:

Index - the index of the view to retrieve. Note that the index of the first view in the list is 0.

Returns:

the specified item from the list of tables, matrices, and diagrams that are to be searched

Throws:

RhapsodyRuntimeException

---

getViewCount

int **getViewCount**()

Returns the number of views in the list of views that are to be searched.

Returns:

the number of views in the list of views that are to be searched

Throws:

RhapsodyRuntimeException

---

loadFromQuery

void **loadFromQuery**(IRPTableLayout query)

Loads the settings from the specified query into the search query object.

Parameters:

query - the query element whose settings should be loaded into the search query object

Throws:

RhapsodyRuntimeException

---

removeFilterElementTypes

void **removeFilterElementTypes**()

Removes any element type filters that you defined to limit the search to certain element types. After calling this method, the search will be applied to all model element types.

Throws:

RhapsodyRuntimeException

---

removeFilterReferences

void **removeFilterReferences**()

> Removes reference search criterion that was defined for the search query.
> **Throws:**
> > [RhapsodyRuntimeException](#)

removeFilterSearchInFields

void **removeFilterSearchInFields**()

> Removes any element field filters that you defined to limit the search to certain element fields, for example, model element descriptions. After calling this method, the search will be applied to all model element fields.
> **Throws:**
> > [RhapsodyRuntimeException](#)

removeFilterStereotypes

void **removeFilterStereotypes**()

> Removes any stereotype filter that was defined to limit the search to model elements that have certain stereotypes applied to them.
> **Throws:**
> > [RhapsodyRuntimeException](#)

removeFilterSubQueries

void **removeFilterSubQueries**()

> Removes the subquery criteria that were specified for the search.
> **Throws:**
> > [RhapsodyRuntimeException](#)

removeFilterSubQuery

int **removeFilterSubQuery**([IRPTableLayout](#) subQuery)

> Removes the specified subquery from the search.
> **Parameters:**
> > subQuery - the subquery that should be removed from the list of subqueries for the search
> **Throws:**
> > [RhapsodyRuntimeException](#)

removeFilterTag

void **removeFilterTag**()

Removes the tag name and tag value criteria that were defined for the search query.
Throws:
    RhapsodyRuntimeException

removeSearchScopeElement

int **removeSearchScopeElement**(IRPModelElement scopeElement)

Removes the specified model element from the scope for the search.
Parameters:
    scopeElement - the model element that should be removed from the scope of the search
Throws:
    RhapsodyRuntimeException

removeView

void **removeView**(int Index)

Removes the specified view from the list of views to be searched for the search text. This method can be used in conjunction with getViewCount and getView to loop through the views in the list and remove a specific one.
Parameters:
    Index - the index of the view in the list of views to search
Throws:
    RhapsodyRuntimeException

resetSearchScope

void **resetSearchScope**()

Resets the search scope to include the entire project, or all projects if multiple projects are open.
Throws:
    RhapsodyRuntimeException

saveAsQuery

IRPTableLayout **saveAsQuery**(IRPPackage queryOwner)

Saves the search query object that you defined as a query in your model.
Parameters:
    queryOwner - the model element under which the new query should be created
Returns:
    the new query element that was created

Throws:

setFilterReference

```
void setFilterReference(java.lang.String quantityOperator,
                        int numberOfReferences,
                        java.lang.String relationKind,
                        java.lang.String typeOfReferencedElements,
                        java.lang.String stereotypeOfReferencedElements,
                        java.lang.String nameOfReferencedElements,
                        int includeReferencedElementsInSearchResults)
```

Sets criteria for the search based on an element's references.

Parameters:

quantityOperator - if you are specifying a number of references as a criterion, use one of the values defined in IRPSearchQuery.References.QuantityOperator to specify whether the criterion should be exactly that number of references, less than that number, or more than that number

numberOfReferences - the number of references that should be used as a search criterion

relationKind - use one of the values defined in IRPSearchQuery.References.RelationKind to specify the type of references that are to be used as a search criterion, for example, aggregates or incoming relations

typeOfReferencedElements - can be used to specify a model element type to further limit the reference criterion, for example, find model elements that have aggregates of type "Attribute". The strings to use for this parameter can be found in the file metaclasses.txt in the Doc directory of the Rhapsody installation. You can also use the names of any "new terms" in your project.

stereotypeOfReferencedElements - use this parameter to specify that the reference criterion should be limited to references to elements that have a specific stereotype applied to them

nameOfReferencedElements - use this parameter to specify that the reference criterion should be limited to references to elements with a specific name

includeReferencedElementsInSearchResults - use 1 to specify that the referenced elements included in the search criterion should also be displayed in the search results, 0 otherwise. For example, if you searched for classes that have aggregates of type "Attribute" with the stereotype Web Managed applied to them, and you used 1 for this parameter, the results will list the classes found as well as their attributes that have the Web Managed stereotype.

setFilterTag

```
void setFilterTag(java.lang.String tagName,
                  java.lang.String tagValue,
                  int matchCase,
                  int matchWholeWord,
                  char findAs)
```

Sets tag name and tag value criteria for the search query.

Parameters:

tagName - the text to use for the tag name criterion

tagValue - the text to use for the tag value criterion

matchCase - use 1 to require an exact match for the tag name and tag value search criteria, in terms of upper and lower case, use 0 otherwise

matchWholeWord - use 1 to require a whole word match for the tag name and tag value search criteria, use 0 otherwise

findAs - use one of the constants defined in the class SearchFindAsEnum to indicate the type of search that should be used for the tag name and tag value search criteria. For example, use SearchFindAsEnum.RP_SEARCH_WILDCARD for a wildcard search or SearchFindAsEnum.RP_SEARCH_REGEX for a regular expression search. If you want to search for elements that have an empty string for a tag value, use SearchFindAsEnum.RP_SEARCH_EMPTY_ONLY.

Throws:
> RhapsodyRuntimeException

---

getFilterSubQueriesOperator

java.lang.String **getFilterSubQueriesOperator**()

> Returns indication of how the specified subqueries are to be combined in the search
> **Returns:**
>> indication of how the subqueries are to be combined in the search - will be one of the constants defined in the class IRPSearchQuery.SubQueriesOperator, for example IRPSearchQuery.SubQueriesOperator.AND.

---

getFilterUnitsOnly

int **getFilterUnitsOnly**()

> Checks whether the search is limited to model elements that are saved units.
> **Returns:**
>> 1 if the search is limited to saved units, 0 otherwise
> **Throws:**
>> RhapsodyRuntimeException

---

getFilterUnresolvedKind

java.lang.String **getFilterUnresolvedKind**()

> Returns the method that was specified for handling unresolved elements in the search. The value returned will be one of the constants from the class IRPSearchQuery.UnresolvedKind.
> **Returns:**
>> the method that was specified for handling unresolved elements in the search

---

getIncludeDescendants

int **getIncludeDescendants**()

Checks whether the scope of the search is to include the descendants of the elements specified for the scope.

Returns:

1 if the scope of the search is to include the descendants of the elements specified for the scope, 0 otherwise

Throws:

RhapsodyRuntimeException

getInterfaceName

java.lang.String **getInterfaceName**()

Returns the name of the interface (IRPSearchQuery).

Returns:

the name of the interface (IRPSearchQuery)

Throws:

RhapsodyRuntimeException

getMatchCase

int **getMatchCase**()

Checks whether an exact match was specified for the query in terms of upper and lower case.

Returns:

1 if an exact match was specified in terms of upper and lower case, 0 otherwise

Throws:

RhapsodyRuntimeException

getMatchSpecifiedCriteria

int **getMatchSpecifiedCriteria**()

Checks whether the query is to return the model elements that match the criteria specified, or the model elements that do not match the criteria specified.

Returns:

1 if the query is to return the model elements that match the criteria specified, 0 if the query is to return the model elements that do not match the specified criteria

Throws:

RhapsodyRuntimeException

getMatchWholeWord

int **getMatchWholeWord**()

Checks whether a whole word match was specified for the search.
Returns:
1 if a whole word match was specified, 0 otherwise
Throws:
<u>RhapsodyRuntimeException</u>

getSearchFindAsOption

char **getSearchFindAsOption**()

Returns the type of search that was specified for the search text - regular text, wildcard, regular expression, or empty string.
Returns:
the type of search that was specified for the search text - will be one of the constants defined in the class <u>SearchFindAsEnum</u>, for example SearchFindAsEnum.RP_SEARCH_WILDCARD for a wildcard search or SearchFindAsEnum.RP_SEARCH_REGEX for a regular expression search.
Throws:
<u>RhapsodyRuntimeException</u>

getSearchScopeObject

<u>IRPModelElement</u> **getSearchScopeObject**()

Deprecated. *This method, used to return the scope specified for the search, was introduced when Rhapsody only allowed you to specify a single element as the scope. Now that Rhapsody allows you to specify a list of such elements, you should use the method <u>getSearchScopeElements()</u>.*

getSearchText

java.lang.String **getSearchText**()

Returns the text that was specified as the text to search for.
Returns:
the text to search for
Throws:
<u>RhapsodyRuntimeException</u>

getViewIncludeModelElements

int **getViewIncludeModelElements**()

Checks whether the query specifies that the search results should also include model elements that were found by the search but are not referenced in any of the views that you specified.

Returns:

> 1 if the query specified that the search results should also include model elements that were found by the search but are not referenced in any of the views that you specified, 0 otherwise

Throws:

> RhapsodyRuntimeException

getViewsToSearch

java.lang.String **getViewsToSearch**()

Returns indication of which views (diagrams, tables, and matrices) are supposed to be searched.

Returns:

> indication of which views are supposed to be searched - will be one of the constants defined in the class IRPSearchQuery.ViewsToSearch, for example IRPSearchQuery.ViewsToSearch.OPEN.

setFilterSubQueriesOperator

void **setFilterSubQueriesOperator**(java.lang.String filterSubQueriesOperator)

Specify how the various subqueries specified should be combined - as an AND operation or an OR operation

Parameters:

> filterSubQueriesOperator - use one of the constants defined in the class IRPSearchQuery.SubQueriesOperator to indicate how the specified subqueries should be combined, for example IRPSearchQuery.SubQueriesOperator.AND

setFilterUnitsOnly

void **setFilterUnitsOnly**(int filterUnitsOnly)

Specifies whether the search should be limited to model elements that are saved units.

Parameters:

> filterUnitsOnly - use 1 to specify that the search should be limited to model elements that are saved units, 0 otherwise

Throws:

> RhapsodyRuntimeException

setFilterUnresolvedKind

void **setFilterUnresolvedKind**(java.lang.String filterUnresolvedKind)

Specifies how unresolved elements should be handled in the search.

Parameters:

> filterUnresolvedKind - how unresolved elements should be handled in the search. The value of the parameter should be one of the constants from the class <u>IRPSearchQuery.UnresolvedKind</u>.

---

setIncludeDescendants

void **setIncludeDescendants**(int includeDescendants)

> Specifies whether the scope for the search should include the descendants of the elements specified for the scope, for example, the subpackages of a package that was added to the scope.
> **Parameters:**
> > includeDescendants - use 1 if you want the search scope to include the descendants of the specified elements, 0 otherwise
> **Throws:**
> > <u>RhapsodyRuntimeException</u>

---

setMatchCase

void **setMatchCase**(int matchCase)

> Specifies whether the search should require an exact match in terms of upper and lower case.
> **Parameters:**
> > matchCase - use 1 to specify that an exact match is required in terms of upper and lower case, 0 otherwise
> **Throws:**
> > <u>RhapsodyRuntimeException</u>

---

setMatchSpecifiedCriteria

void **setMatchSpecifiedCriteria**(int matchSpecifiedCriteria)

> Specifies whether the query should return the model elements that match the criteria specified, or the model elements that do not match the criteria specified.
> **Parameters:**
> > matchSpecifiedCriteria - use 1 if you want the query to return the model elements that match the criteria specified, use 0 if you want the query to return the model elements that do not match the criteria specified
> **Throws:**
> > <u>RhapsodyRuntimeException</u>

---

setMatchWholeWord

void **setMatchWholeWord**(int matchWholeWord)

> Specifies whether the search should require whole word matches.
> **Parameters:**

---

matchWholeWord - use 1 to specify that a whole word match is required, 0 otherwise
Throws:
RhapsodyRuntimeException

setSearchFindAsOption

void **setSearchFindAsOption**(char searchFindAsOption)

Sets the type of search that should be used for the search text - regular text, wildcard, regular expression, or empty string.
Parameters:
searchFindAsOption - use one of the constants defined in the class SearchFindAsEnum to indicate the type of search that should be used for the search text. For example, use SearchFindAsEnum.RP_SEARCH_WILDCARD for a wildcard search or SearchFindAsEnum.RP_SEARCH_REGEX for a regular expression search. If you want to search for elements that have an empty string in certain fields, use SearchFindAsEnum.RP_SEARCH_EMPTY_ONLY.
Throws:
RhapsodyRuntimeException

setSearchScopeObject

void **setSearchScopeObject**(IRPModelElement searchScopeObject)

Deprecated. *This method, used to set the scope for the search, was introduced when Rhapsody only allowed you to specify a single element as the scope. Now that Rhapsody allows you to specify a list of such elements, you should use the method*
addSearchScope(com.telelogic.rhapsody.core.IRPModelElement).

setSearchText

void **setSearchText**(java.lang.String searchText)

Specifies the text that should be searched for.
Parameters:
searchText - the text that should be searched for
Throws:
RhapsodyRuntimeException

setViewIncludeModelElements

void **setViewIncludeModelElements**(int viewIncludeModelElements)

Specifies whether the search results should also include model elements that were found by the search but are not referenced in any of the views that you specified.
Parameters:

setMatchWholeWord

> `viewIncludeModelElements` - use 1 to specify that the search results should also include model elements that were found by the search but are not referenced in any of the views that you specified, 0 otherwise

Throws:
> [RhapsodyRuntimeException](#)

---

setViewsToSearch

void **setViewsToSearch**(java.lang.String viewsToSearch)

> Specifies which views (tables, matrices, and diagrams) should be searched - all, none, all open, or just the views that were specified with the methods addDiagramToViewsList, addTableToViewsList, and addMatrixToViewsList.
>
> **Parameters:**
>> `viewsToSearch` - use one of the constants defined in the class [IRPSearchQuery.ViewsToSearch](#) to indicate which views should be searched, for example IRPSearchQuery.ViewsToSearch.ALL
>
> **Throws:**
>> [RhapsodyRuntimeException](#)

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

PREV CLASS **NEXT CLASS**   **FRAMES**   **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Class IRPSearchQuery.References

```
java.lang.Object
  └─com.telelogic.rhapsody.core.IRPSearchQuery.References
```

Enclosing interface:
> IRPSearchQuery

---

public static final class **IRPSearchQuery.References**
extends java.lang.Object

Nested Class Summary

static class	**IRPSearchQuery.References.QuantityOperator**
static class	**IRPSearchQuery.References.RelationKind**

Constructor Summary

IRPSearchQuery.References()	

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

IRPSearchQuery.References

```
public IRPSearchQuery.References()
```

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

PREV CLASS **NEXT CLASS**

SUMMARY: NESTED | FIELD | CONSTR | METHOD

FRAMES **NO FRAMES** **All Classes**

DETAIL: FIELD | CONSTR | METHOD

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Class IRPSearchQuery.References.QuantityOperator

```
java.lang.Object
   └ com.telelogic.rhapsody.core.IRPSearchQuery.References.QuantityOperator
```

Enclosing class:
> IRPSearchQuery.References

---

public static final class **IRPSearchQuery.References.QuantityOperator**
extends java.lang.Object

Field Summary

static java.lang.String	**EXACTLY**
static java.lang.String	**LESS_THAN**
static java.lang.String	**MORE_THAN**

Constructor Summary

IRPSearchQuery.References.QuantityOperator()

Method Summary

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

MORE_THAN

public static final java.lang.String **MORE_THAN**

>   **See Also:**
>   [Constant Field Values](#)

LESS_THAN

public static final java.lang.String **LESS_THAN**

>   **See Also:**
>   [Constant Field Values](#)

EXACTLY

public static final java.lang.String **EXACTLY**

>   **See Also:**
>   [Constant Field Values](#)

Constructor Detail

IRPSearchQuery.References.QuantityOperator

public **IRPSearchQuery.References.QuantityOperator**()

com.telelogic.rhapsody.core
Class IRPSearchQuery.References.RelationKind

```
java.lang.Object
  └com.telelogic.rhapsody.core.IRPSearchQuery.References.RelationKind
```

Enclosing class:
>   IRPSearchQuery.References

---

public static final class **IRPSearchQuery.References.RelationKind**
extends java.lang.Object

Field Summary

static java.lang.String	**AGGREGATE**
static java.lang.String	**DIAGRAM_ELEMENT**
static java.lang.String	**INCOMING_RELATION**
static java.lang.String	**OUTGOING_RELATION**
static java.lang.String	**REFERENCE**
static java.lang.String	**UNDEFINED_RELATION**

Constructor Summary

IRPSearchQuery.References.RelationKind()

Method Summary

Methods inherited from class java.lang.Object

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

Field Detail

UNDEFINED_RELATION

public static final java.lang.String **UNDEFINED_RELATION**

> **See Also:**
> [Constant Field Values](#)

INCOMING_RELATION

public static final java.lang.String **INCOMING_RELATION**

> **See Also:**
> [Constant Field Values](#)

OUTGOING_RELATION

public static final java.lang.String **OUTGOING_RELATION**

> **See Also:**
> [Constant Field Values](#)

AGGREGATE

public static final java.lang.String **AGGREGATE**

> **See Also:**
> [Constant Field Values](#)

REFERENCE

public static final java.lang.String **REFERENCE**

> **See Also:**
> [Constant Field Values](#)

DIAGRAM_ELEMENT

`public static final java.lang.String `**`DIAGRAM_ELEMENT`**

See Also:
[Constant Field Values](#)

Constructor Detail

IRPSearchQuery.References.RelationKind

`public `**`IRPSearchQuery.References.RelationKind`**`()`

com.telelogic.rhapsody.core
Class IRPSearchQuery.SearchInField

```
java.lang.Object
  └─com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField
```

Enclosing interface:
    IRPSearchQuery

---

public static final class **IRPSearchQuery.SearchInField**
extends java.lang.Object

Constant values used with elements of this type

Field Summary

static java.lang.String	**COMMENT_SPECIFICATION**
static java.lang.String	**CONFIGURATION_INITIALIZATION**
static java.lang.String	**CONSTRAINT_SPECIFICATION**
static java.lang.String	**DESCRIPTIONS**
static java.lang.String	**ENUMERATION_LITERAL_VALUE**
static java.lang.String	**GROUP_ALL**
static java.lang.String	**GROUP_CODE**
static java.lang.String	**GROUP_ELEMENT_NAME**
static java.lang.String	**GROUP_OTHER_TEXT**
static java.lang.String	**INITIAL_VALUE**

Field Summary

static java.lang.String	**LABEL**
static java.lang.String	**LOCALLY_OVERRIDDEN_PROPERTY**
static java.lang.String	**MULTIPLICITY**
static java.lang.String	**NAME**
static java.lang.String	**NOTES_AND_TEXT**
static java.lang.String	**OPERATION_BODIES**
static java.lang.String	**REQUIREMENT_ID**
static java.lang.String	**REQUIREMENT_SPECIFICATION**
static java.lang.String	**TAG_VALUE**
static java.lang.String	**TEXT_FRAGMENT**
static java.lang.String	**TRANSITION_LABEL**
static java.lang.String	**TYPE_DECLARATIONS_AND_REFERENCES**

Constructor Summary

IRPSearchQuery.SearchInField()	

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

OPERATION_BODIES

public static final java.lang.String **OPERATION_BODIES**

>**See Also:**
>   Constant Field Values

TRANSITION_LABEL

public static final java.lang.String **TRANSITION_LABEL**

>**See Also:**
>   Constant Field Values

TAG_VALUE

public static final java.lang.String **TAG_VALUE**

>**See Also:**
>   Constant Field Values

TYPE_DECLARATIONS_AND_REFERENCES

public static final java.lang.String **TYPE_DECLARATIONS_AND_REFERENCES**

>**See Also:**
>   Constant Field Values

CONFIGURATION_INITIALIZATION

public static final java.lang.String **CONFIGURATION_INITIALIZATION**

>**See Also:**
>   Constant Field Values

MULTIPLICITY

public static final java.lang.String **MULTIPLICITY**

>**See Also:**
>   Constant Field Values

LOCALLY_OVERRIDDEN_PROPERTY

public static final java.lang.String **LOCALLY_OVERRIDDEN_PROPERTY**

See Also:
Constant Field Values

DESCRIPTIONS

public static final java.lang.String **DESCRIPTIONS**

See Also:
Constant Field Values

COMMENT_SPECIFICATION

public static final java.lang.String **COMMENT_SPECIFICATION**

See Also:
Constant Field Values

CONSTRAINT_SPECIFICATION

public static final java.lang.String **CONSTRAINT_SPECIFICATION**

See Also:
Constant Field Values

REQUIREMENT_SPECIFICATION

public static final java.lang.String **REQUIREMENT_SPECIFICATION**

See Also:
Constant Field Values

NOTES_AND_TEXT

public static final java.lang.String **NOTES_AND_TEXT**

See Also:
Constant Field Values

LABEL

public static final java.lang.String **LABEL**

> **See Also:**
> [Constant Field Values](#)

INITIAL_VALUE

public static final java.lang.String **INITIAL_VALUE**

> **See Also:**
> [Constant Field Values](#)

ENUMERATION_LITERAL_VALUE

public static final java.lang.String **ENUMERATION_LITERAL_VALUE**

> **See Also:**
> [Constant Field Values](#)

REQUIREMENT_ID

public static final java.lang.String **REQUIREMENT_ID**

> **See Also:**
> [Constant Field Values](#)

NAME

public static final java.lang.String **NAME**

> **See Also:**
> [Constant Field Values](#)

TEXT_FRAGMENT

public static final java.lang.String **TEXT_FRAGMENT**

> **See Also:**
> [Constant Field Values](#)

GROUP_ELEMENT_NAME

public static final java.lang.String **GROUP_ELEMENT_NAME**

> **See Also:**
> [Constant Field Values](#)

GROUP_CODE

public static final java.lang.String **GROUP_CODE**

> **See Also:**
> [Constant Field Values](#)

GROUP_OTHER_TEXT

public static final java.lang.String **GROUP_OTHER_TEXT**

> **See Also:**
> [Constant Field Values](#)

GROUP_ALL

public static final java.lang.String **GROUP_ALL**

> **See Also:**
> [Constant Field Values](#)

Constructor Detail

IRPSearchQuery.SearchInField

public **IRPSearchQuery.SearchInField**()

---

com.telelogic.rhapsody.core
Class IRPSearchQuery.SubQueriesOperator

```
java.lang.Object
   └─com.telelogic.rhapsody.core.IRPSearchQuery.SubQueriesOperator
```

Enclosing interface:
> IRPSearchQuery

---

public static final class **IRPSearchQuery.SubQueriesOperator**
extends java.lang.Object

Field Summary

static java.lang.String	**AND**
static java.lang.String	**OR**

Constructor Summary

IRPSearchQuery.SubQueriesOperator()	

Method Summary

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

AND

```
public static final java.lang.String AND
```

> **See Also:**
> [Constant Field Values](#)

OR

```
public static final java.lang.String OR
```

> **See Also:**
> [Constant Field Values](#)

Constructor Detail

IRPSearchQuery.SubQueriesOperator

```
public IRPSearchQuery.SubQueriesOperator()
```

---

com.telelogic.rhapsody.core
Class IRPSearchQuery.UnresolvedKind

```
java.lang.Object
  └─com.telelogic.rhapsody.core.IRPSearchQuery.UnresolvedKind
```

Enclosing interface:
> IRPSearchQuery

---

public static final class **IRPSearchQuery.UnresolvedKind**
extends java.lang.Object

Field Summary

static java.lang.String	**IGNORE_UNRESOLVED**
static java.lang.String	**ONLY_UNRESOLVED_OR_UNLOADED**
static java.lang.String	**SHOW_UNRESOLVED**

Constructor Summary

IRPSearchQuery.UnresolvedKind()	

Method Summary

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

IGNORE_UNRESOLVED

`public static final java.lang.String` **`IGNORE_UNRESOLVED`**

See Also:
[Constant Field Values](#)

SHOW_UNRESOLVED

`public static final java.lang.String` **`SHOW_UNRESOLVED`**

See Also:
[Constant Field Values](#)

ONLY_UNRESOLVED_OR_UNLOADED

`public static final java.lang.String` **`ONLY_UNRESOLVED_OR_UNLOADED`**

See Also:
[Constant Field Values](#)

Constructor Detail

IRPSearchQuery.UnresolvedKind

`public` **`IRPSearchQuery.UnresolvedKind`**`()`

---

com.telelogic.rhapsody.core
Class IRPSearchQuery.ViewsToSearch

```
java.lang.Object
  └─com.telelogic.rhapsody.core.IRPSearchQuery.ViewsToSearch
```

Enclosing interface:
> IRPSearchQuery

---

public static final class **IRPSearchQuery.ViewsToSearch**
extends java.lang.Object

Field Summary

static java.lang.String	**ALL**
static java.lang.String	**DETAILED**
static java.lang.String	**NONE**
static java.lang.String	**OPEN**

Constructor Summary

IRPSearchQuery.ViewsToSearch()	

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

IRPSearchQuery.UnresolvedKind   559

Field Detail

NONE

`public static final java.lang.String **NONE**`

> **See Also:**
> [Constant Field Values](#)

OPEN

`public static final java.lang.String **OPEN**`

> **See Also:**
> [Constant Field Values](#)

ALL

`public static final java.lang.String **ALL**`

> **See Also:**
> [Constant Field Values](#)

DETAILED

`public static final java.lang.String **DETAILED**`

> **See Also:**
> [Constant Field Values](#)

Constructor Detail

IRPSearchQuery.ViewsToSearch

`public **IRPSearchQuery.ViewsToSearch**()`

[Package](#)   **Class**   **[Use](#)**   **[Tree](#)**   **[Serialized](#)**   **[Deprecated](#)**   **[Index](#)**   **[Help](#)**

[PREV CLASS](#)   **[NEXT CLASS](#)**   **[FRAMES](#)**   **[NO FRAMES](#)**   **[All Classes](#)**

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)   DETAIL: [FIELD](#) | [CONSTR](#) | METHOD

com.telelogic.rhapsody.coreClass IRPSearchQuery.ViewsToSearch   560

com.telelogic.rhapsody.core
Interface IRPSearchResult

public interface **IRPSearchResult**

Method Summary

java.lang.String	**getInterfaceName**() get property interfaceName
java.lang.String	**getMatchedField**() get property matchedField
IRPCollection	**getMatchedFields**() get property matchedFields
IRPModelElement	**getMatchedObject**() get property matchedObject
java.lang.String	**getName**() get property name

Method Detail

getInterfaceName

java.lang.String **getInterfaceName**()

> get property interfaceName
> **Throws:**
> > RhapsodyRuntimeException

---

getMatchedField

java.lang.String **getMatchedField**()

> get property matchedField
> **Throws:**
> > RhapsodyRuntimeException

getMatchedFields

IRPCollection **getMatchedFields**()

> get property matchedFields
> **Throws:**
> > RhapsodyRuntimeException

getMatchedObject

IRPModelElement **getMatchedObject**()

> get property matchedObject
> **Throws:**
> > RhapsodyRuntimeException

getName

java.lang.String **getName**()

> get property name
> **Throws:**
> > RhapsodyRuntimeException

com.telelogic.rhapsody.core
Interface IRPSelection

```
public interface IRPSelection
```

Method Summary

int	**canCopy**() Check if COPY operation is available for current selection
int	**canCut**() Check if CUT operation is available for current selection
int	**canDelete**() Check if DELETE operation is available for current selection
int	**canPaste**() Check if PASTE operation is available for current selection
int	**copySelected**() Perform COPY operation under current selection
int	**cutSelected**() Perform CUT operation under current selection
int	**deleteSelected**() Perform DELETE operation under current selection
java.lang.String	**getInterfaceName**() get property interfaceName
int	**pasteSelected**() Perform PASTE operation under current selection

Method Detail

canCopy

```
int canCopy()
```

> Check if COPY operation is available for current selection
> **Throws:**

getName   563

RhapsodyRuntimeException

canCut

int **canCut**()

Check if CUT operation is available for current selection
Throws:
RhapsodyRuntimeException

canDelete

int **canDelete**()

Check if DELETE operation is available for current selection
Throws:
RhapsodyRuntimeException

canPaste

int **canPaste**()

Check if PASTE operation is available for current selection
Throws:
RhapsodyRuntimeException

copySelected

int **copySelected**()

Perform COPY operation under current selection
Throws:
RhapsodyRuntimeException

cutSelected

int **cutSelected**()

Perform CUT operation under current selection
Throws:
RhapsodyRuntimeException

deleteSelected

`int` **`deleteSelected`**`()`

> Perform DELETE operation under current selection
> **Throws:**
> > RhapsodyRuntimeException

---

getInterfaceName

`java.lang.String` **`getInterfaceName`**`()`

> get property interfaceName
> **Throws:**
> > RhapsodyRuntimeException

---

pasteSelected

`int` **`pasteSelected`**`()`

> Perform PASTE operation under current selection
> **Throws:**
> > RhapsodyRuntimeException

---

com.telelogic.rhapsody.core
Interface IRPSendAction

All Superinterfaces:
        IRPAction, IRPModelElement

---

public interface **IRPSendAction**
extends IRPAction

The IRPSendAction interface represents Send Action elements in an activity or statechart. To add a
SendAction element, use addState to add a new state, and then call the method setStateType on the state your
created, using "EventState" as the argument, for example:

```
IRPState sendActionState = activity_1.getRootState().addState("send_action");
sendActionState.setStateType("EventState");
```

After creating the send action state, you get the send action element as follows:

```
IRPSendAction sendActionElement = sendActionState.getSendAction();
IRPEvent eventA = cameraPackage.addEvent("event_A");
sendActionElement.setEvent(eventA);
```

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

void	**addArgumentValue**(java.lang.String value, int position) Provides an argument value for an argument of the event associated with the Send Action element.
IRPCollection	**getArgVals**() Returns a collection of the argument values that were set for the event associated with the Send Action element.
IRPEvent	

Method Summary

	getEvent() Gets the event sent by the Send Action element.
IRPInterfaceItem	**getInvokedOperation**() Returns the IRPInterfaceItem element that is invoked by the Send Action element.
IRPModelElement	**getTarget**() Gets the event target of the Send Action element.
void	**setEvent**(IRPEvent event) Specifies the event sent by the Send Action element.
void	**setInvokedOperation**(IRPInterfaceItem invokedOperation) set property invokedOperation
void	**setTarget**(IRPModelElement target) Sets the specified model element to be the target of the Send Action element.

Methods inherited from interface com.telelogic.rhapsody.core.IRPAction

getBody, setBody

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addArgumentValue

```
void addArgumentValue(java.lang.String value,
                      int position)
```

Provides an argument value for an argument of the event associated with the Send Action element.
Parameters:
value - the value to use for the argument, expressed as a string
position - the position of the argument in the argument list (starts at 1)

---

getArgVals

IRPCollection **getArgVals**()

Returns a collection of the argument values that were set for the event associated with the Send Action element. The collection consists of strings representing the argument values.
Returns:
the argument values that were set for the event associated with the Send Action element

---

getEvent

IRPEvent **getEvent**()

Gets the event sent by the Send Action element.
Returns:
the event sent by the Send Action element

---

getInvokedOperation

IRPInterfaceItem **getInvokedOperation**()

Returns the IRPInterfaceItem element that is invoked by the Send Action element.
Returns:
the IRPInterfaceItem element that is invoked by the Send Action element

---

getTarget

IRPModelElement **getTarget**()

Gets the event target of the Send Action element.
Returns:
the target of the Send Action element

---

setEvent

void **setEvent**([IRPEvent](#) event)

Specifies the event sent by the Send Action element.
Parameters:
event - the event that should be sent by the Send Action element

setInvokedOperation

void **setInvokedOperation**([IRPInterfaceItem](#) invokedOperation)

set property invokedOperation
Throws:
[RhapsodyRuntimeException](#)

setTarget

void **setTarget**([IRPModelElement](#) target)

Sets the specified model element to be the target of the Send Action element.
Parameters:
target - the model element that should be used as the target of the Send Action element

---

com.telelogic.rhapsody.core
Interface IRPSequenceDiagram

All Superinterfaces:
> IRPDiagram, IRPModelElement, IRPUnit

All Known Subinterfaces:
> IRPTimingDiagram

---

public interface **IRPSequenceDiagram**
extends IRPDiagram

The IRPSequenceDiagram interface represents sequence diagrams in a Rational Rhapsody model.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPCollaboration	**getLogicalCollaboration**() Returns the IRPCollaboration object underlying the sequence diagram.
IRPCollection	**getRelatedUseCases**() For internal use only.

Methods inherited from interface com.telelogic.rhapsody.core.IRPDiagram
addFreeShapeByType, addImage, addNewEdgeByType, addNewEdgeForElement, addNewNodeByType, addNewNodeForElement, addTextBox, closeDiagram, completeRelations, getCorrespondingGraphicElements, getElementsInDiagram, getGraphicalElements, getLastVisualizationModifiedTime, getPicture, getPictureAs, getPictureAsDividedMetafiles, getPictureEx, getPicturesWithImageMap, isOpen, isShowDiagramFrame, openDiagram, openDiagramView, populateDiagram, removeGraphElements, setShowDiagramFrame

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit
copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime,

---

setTarget   570

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getLogicalCollaboration

IRPCollaboration **getLogicalCollaboration**()

> Returns the IRPCollaboration object underlying the sequence diagram.
> **Returns:**
> > the IRPCollaboration object underlying the sequence diagram

getRelatedUseCases

IRPCollection **getRelatedUseCases**()

 com.telelogic.rhapsody.core
Interface IRPState

All Superinterfaces:
>   IRPModelElement, IRPStateVertex

All Known Subinterfaces:
>   IRPAcceptEventAction, IRPAcceptTimeEvent, IRPCallOperation, IRPObjectNode

---

```
public interface IRPState
```
extends IRPStateVertex

The IRPState interface represents states in a statechart.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPState	**addActivityFinal**() Adds an ActivityFinal element to an Activity.
IRPConnector	**addConnector**(java.lang.String type) Adds a connector element of the specified type to the state.
IRPTransition	**addInternalTransition**(IRPInterfaceItem trigger) method addInternalTransition
IRPState	**addState**(java.lang.String name) Adds a new substate to this state.
IRPTransition	**addStaticReaction**(IRPInterfaceItem trigger) Adds an internal transition to the state.
IRPState	**addTerminationState**() Adds a termination state to a statechart.
IRPTransition	**createDefaultTransition**(IRPState from) Creates a default transition within the state.

Method Summary

[IRPStatechart](#)	**createNestedStatechart**() Creates a sub-statechart for the state.
void	**deleteConnector**([IRPConnector](#) connector) Deletes the specified connector element.
void	**deleteInternalTransition**([IRPTransition](#) pVal) method deleteInternalTransition
void	**deleteStaticReaction**([IRPTransition](#) pVal) Deletes the specified internal transition.
[IRPTransition](#)	**getDefaultTransition**() Returns the default transition within the state.
java.lang.String	**getEntryAction**() Returns the entry action that was defined for the state.
java.lang.String	**getExitAction**() Returns the exit action that was defined for the state.
java.lang.String	**getFullNameInStatechart**() Returns the full name of the state within the statechart, including information about its hierarchical position within the statechart.
[IRPState](#)	**getInheritsFrom**() Returns the corresponding state from the statechart of the class that this class is derived from.
[IRPCollection](#)	**getInternalTransitions**() Returns a collection of the state's internal transitions.
int	**getIsOverridden**() Checks whether there is still an inheritance relationship between this state and the corresponding state from the statechart of the class that this class is derived from.
int	**getIsReferenceActivity**() Checks whether this element is a call behavior element.
[IRPStatechart](#)	**getItsStatechart**() Returns the statechart that this state belongs to.
[IRPSwimlane](#)	**getItsSwimlane**() Returns the swimlane that the action is located in.
[IRPCollection](#)	**getLogicalStates**() Returns a collection of all the substates of the current state and all the first-level substates of those states, meaning down to the second level.
[IRPStatechart](#)	**getNestedStatechart**() Returns the state's sub-statechart.
[IRPModelElement](#)	**getReferenceToActivity**() For call behavior elements, returns the activity that is referenced.
[IRPSendAction](#)	**getSendAction**()

Method Summary

	Returns the Send Action element associated with the state.
java.lang.String	**getStateType**() Returns the type of the state, for example, an And state or a Termination state.
[IRPCollection](#)	**getStaticReactions**() Returns a collection of the state's internal transitions.
[IRPCollection](#)	**getSubStates**() Returns a collection of the substates contained in this state.
[IRPCollection](#)	**getSubStateVertices**() Returns a collection of all the first-level elements contained in this state - this includes both node elements and connector elements.
[IRPAction](#)	**getTheEntryAction**() method getTheEntryAction
[IRPAction](#)	**getTheExitAction**() method getTheExitAction
int	**isAnd**() Checks whether the state contains one or more And Lines.
int	**isCompound**() Checks whether the state is a compound state, meaning a state that contains one or more substates.
int	**isLeaf**() Checks whether the state is a leaf state, meaning a state that does not contain any substates.
int	**isRoot**() Checks whether the state is the root state of the statechart.
int	**isSendActionState**() Checks whether the state is a Send Action element.
void	**overrideInheritance**() Breaks the inheritance relationship between this state and the corresponding state from the statechart of the class that this class is derived from.
[IRPState](#)	**resetEntryActionInheritance**() Restores the inheritance relationship between this state and the corresponding state from the statechart of the class that this class is derived from, for the entry action.
[IRPState](#)	**resetExitActionInheritance**() Restores the inheritance relationship between this state and the corresponding state from the statechart of the class that this class is derived from, for the exit action.
void	**setEntryAction**(java.lang.String entryAction) Sets the entry action for the state.
void	**setExitAction**(java.lang.String exitAction) Sets the exit action for the state.

Method Summary

void	**setInternalTransition**(java.lang.String trigVal, java.lang.String guardVal, java.lang.String actionVal) method setInternalTransition
void	**setItsSwimlane**([IRPSwimlane](#) itsSwimlane) Specifies the swimlane that the action should be in
void	**setReferenceToActivity**([IRPModelElement](#) referenceToActivity) For call behavior elements, sets the activity that is referenced by the element.
void	**setStateType**(java.lang.String stateType) Specifies the type of the state
void	**setStaticReaction**(java.lang.String trigVal, java.lang.String guardVal, java.lang.String actionVal) Adds a new internal transition to the state.
void	**unoverrideInheritance**() Restores the inheritance relationship between this state and the corresponding state from the statechart of the class that this class is derived from.

Methods inherited from interface com.telelogic.rhapsody.core.**IRPStateVertex**

addFlow, addTransition, deleteTransition, getInTransitions, getOutTransitions, getParent, setParent

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addActivityFinal

[IRPState](#) **addActivityFinal**()

> Adds an ActivityFinal element to an Activity. This method should be called on the root state of the diagram, which you can get by calling IRPStatechart.getRootState().
> **Returns:**
> > the ActivityFinal element that was created

---

addConnector

[IRPConnector](#) **addConnector**(java.lang.String type)

> Adds a connector element of the specified type to the state.
> **Parameters:**
> > type - the type of connector that should be added - the valid values for this parameter are: Condition, Fork, History, Join, Termination, InPin, OutPin, InOutPin
> **Returns:**
> > the connector element that was created

---

addInternalTransition

[IRPTransition](#) **addInternalTransition**([IRPInterfaceItem](#) trigger)

> method addInternalTransition
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

addState

[IRPState](#) **addState**(java.lang.String name)

> Adds a new substate to this state. If you want to add a new top-level state to your statechart, you can call this method on the root state of the statechart, which you can get by calling IRPStatechart.getRootState().
> **Parameters:**
> > name - the name to use for the new state
> **Returns:**
> > the state created

---

addStaticReaction

[IRPTransition](#) **addStaticReaction**([IRPInterfaceItem](#) trigger)

> Adds an internal transition to the state.
> **Parameters:**
>> `trigger` - the trigger to use for the internal transition
> **Returns:**
>> the internal transition that was created

addTerminationState

[IRPState](#) **addTerminationState**()

> Adds a termination state to a statechart. This method should be called on the root state of the statechart, which you can get by calling IRPStatechart.getRootState().
> **Returns:**
>> the termination state that was created

createDefaultTransition

[IRPTransition](#) **createDefaultTransition**([IRPState](#) from)

> Creates a default transition within the state.
> **Parameters:**
>> `from` - the substate that the default transition should lead to
> **Returns:**
>> the default transition that was created

createNestedStatechart

[IRPStatechart](#) **createNestedStatechart**()

> Creates a sub-statechart for the state.
> **Returns:**
>> the sub-statechart created

deleteConnector

void **deleteConnector**([IRPConnector](#) connector)

> Deletes the specified connector element.
> **Parameters:**
>> `connector` - the connector element that should be deleted

deleteInternalTransition

void **deleteInternalTransition**([IRPTransition](#) pVal)

> method deleteInternalTransition
> **Throws:**
> > [RhapsodyRuntimeException](#)

deleteStaticReaction

void **deleteStaticReaction**([IRPTransition](#) pVal)

> Deletes the specified internal transition.
> **Parameters:**
> > pVal - the internal transition that should be deleted

getDefaultTransition

[IRPTransition](#) **getDefaultTransition**()

> Returns the default transition within the state.
> **Returns:**
> > the default transition within the state

getEntryAction

java.lang.String **getEntryAction**()

> Returns the entry action that was defined for the state.
> **Returns:**
> > the entry action that was defined for the state

getExitAction

java.lang.String **getExitAction**()

> Returns the exit action that was defined for the state.
> **Returns:**
> > the exit action that was defined for the state

getFullNameInStatechart

java.lang.String **getFullNameInStatechart**()

Returns the full name of the state within the statechart, including information about its hierarchical position within the statechart. For example, if your statechart includes a state called Listening within a top-level state called On, the full name would be ROOT.On.Listening.
Returns:
> the full name of the state within the statechart

getInheritsFrom

[IRPState](#) **getInheritsFrom**()

Returns the corresponding state from the statechart of the class that this class is derived from.
Returns:
> the corresponding state from the statechart of the class that this class is derived from

getInternalTransitions

[IRPCollection](#) **getInternalTransitions**()

Returns a collection of the state's internal transitions.
Returns:
> the state's internal transitions

```
IRPStatechart cameraStatechart = cameraClass.addStatechart();
IRPEvent trig_for_internal = cameraPackage.addEvent("trigger_internal");
IRPEvent trig_for_internal2 = cameraPackage.addEvent("trigger_internal2");
IRPState stateOne = cameraStatechart.getRootState().addState("state_one");
stateOne.addInternalTransition(trig_for_internal);
stateOne.addInternalTransition(trig_for_internal2);
// now, get and print out the state's internal transitions
IRPCollection allInternalTransitions = stateOne.getInternalTransitions();
IRPTransition currentTransition;
int numberOfInternalTransitions = allInternalTransitions.getCount();
for(int i = 1; i < numberOfInternalTransitions+1 ; i++) {
        currentTransition = (IRPTransition)allInternalTransitions.getItem(i);
        System.out.println(currentTransition.getDisplayName());
}
```

getIsOverridden

int **getIsOverridden**()

Checks whether there is still an inheritance relationship between this state and the corresponding state from the statechart of the class that this class is derived from.
Returns:
> 1 if the inheritance relationship is overridden, 0 if there is an inheritance relationship

getIsReferenceActivity

int **getIsReferenceActivity**()

Checks whether this element is a call behavior element. Note that the Rhapsody API does not provide a method to change an existing IRPState element to a call behavior element. The only way to create a call behavior element is to call the method IRPFlowchart.addCallBehavior or IRPFlowchart.addReferenceActivity.
Returns:
1 if the element is a call behavior element, 0 if it is not

getItsStatechart

IRPStatechart **getItsStatechart**()

Returns the statechart that this state belongs to.
Returns:
the statechart that this state belongs to

getItsSwimlane

IRPSwimlane **getItsSwimlane**()

Returns the swimlane that the action is located in.
Returns:
the swimlane that the action is located in

getLogicalStates

IRPCollection **getLogicalStates**()

Returns a collection of all the substates of the current state and all the first-level substates of those states, meaning down to the second level.
Returns:
a collection of all the substates of the current state and all the first-level substates of those states

getNestedStatechart

IRPStatechart **getNestedStatechart**()

Returns the state's sub-statechart.
Returns:
the state's sub-statechart

getReferenceToActivity

[IRPModelElement](#) **getReferenceToActivity**()

> For call behavior elements, returns the activity that is referenced.
> **Returns:**
> > the activity that is referenced

---

getSendAction

[IRPSendAction](#) **getSendAction**()

> Returns the Send Action element associated with the state. In the context of the API, a Send Action element is an object of type IRPState for which the state type was set to "EventState" using the setStateType method. In order to manipulate a Send Action element, for example, to set the event for the Send Action, you must first get the Second Action element using the method getSendAction.
> **Returns:**
> > the Send Action element associated with the state

```
 IRPState sendActionState = testfc.getRootState().addState("send_action");
 sendActionState.setStateType("EventState");
 IRPSendAction sendActionElement = sendActionState.getSendAction();
 IRPEvent eventA = cameraPackage.addEvent("event_A");
 sendActionElement.setEvent(eventA);
```

---

getStateType

java.lang.String **getStateType**()

> Returns the type of the state, for example, an And state or a Termination state. For the full list of state types, see the documentation for the operation setStateType.
> **Returns:**
> > the state's type

---

getStaticReactions

[IRPCollection](#) **getStaticReactions**()

> Returns a collection of the state's internal transitions.
> **Returns:**
> > the state's internal transitions

---

getSubStateVertices

[IRPCollection](#) **getSubStateVertices**()

getReferenceToActivity

Returns a collection of all the first-level elements contained in this state - this includes both node elements and connector elements. The method does not return elements nested within these first-level elements.

Returns:
> a collection of all the first-level elements contained in this state

getSubStates

[IRPCollection](#) **getSubStates**()

Returns a collection of the substates contained in this state. Note that this will not work if a state contains a sub-statechart. In such a case, you would have to use code that resembles the following:

```
IRPState parentState = (IRPState)currentProject.findNestedElementRecursive("busy", "Sta
IRPState topLevelStateInSubchart = (IRPState)(parentState.getNestedStatechart().getRoot
IRPCollection substates = topLevelStateInSubchart.getSubStates();
```

Returns:
> the substates contained in this state (collection of IRPState objects)

getTheEntryAction

[IRPAction](#) **getTheEntryAction**()

> method getTheEntryAction
> **Throws:**
>> [RhapsodyRuntimeException](#)

getTheExitAction

[IRPAction](#) **getTheExitAction**()

> method getTheExitAction
> **Throws:**
>> [RhapsodyRuntimeException](#)

isAnd

int **isAnd**()

> Checks whether the state contains one or more And Lines.
> **Returns:**
>> 1 if the state contains one or more And Lines, 0 otherwise.

isCompound

int **isCompound**()

Checks whether the state is a compound state, meaning a state that contains one or more substates.
Returns:
1 if the state is a compound state, 0 otherwise

---

isLeaf

int **isLeaf**()

Checks whether the state is a leaf state, meaning a state that does not contain any substates.
Returns:
1 if the state is a leaf state, 0 if the state contains one or more substates.

---

isRoot

int **isRoot**()

Checks whether the state is the root state of the statechart.
Returns:
1 if the state is the root state of the statechart, 0 otherwise

---

isSendActionState

int **isSendActionState**()

Checks whether the state is a Send Action element.
Returns:
1 if it is a Send Action element, 0 otherwise

---

overrideInheritance

void **overrideInheritance**()

Breaks the inheritance relationship between this state and the corresponding state from the statechart of the class that this class is derived from.

---

resetEntryActionInheritance

IRPState **resetEntryActionInheritance**()

Restores the inheritance relationship between this state and the corresponding state from the statechart of the class that this class is derived from, for the entry action.

Returns:
>   the state on which the method was called (sic)

---

resetExitActionInheritance

[IRPState](#) **resetExitActionInheritance**()

>   Restores the inheritance relationship between this state and the corresponding state from the statechart
>   of the class that this class is derived from, for the exit action.
>   **Returns:**
>   the state on which the method was called (sic)

---

setEntryAction

void **setEntryAction**(java.lang.String entryAction)

>   Sets the entry action for the state.
>   **Parameters:**
>   entryAction - the code to use for the state's entry action

---

setExitAction

void **setExitAction**(java.lang.String exitAction)

>   Sets the exit action for the state.
>   **Parameters:**
>   exitAction - the code to use for the state's exit action

---

setInternalTransition

void **setInternalTransition**(java.lang.String trigVal,
                              java.lang.String guardVal,
                              java.lang.String actionVal)

>   method setInternalTransition
>   **Throws:**
>   [RhapsodyRuntimeException](#)

---

setItsSwimlane

void **setItsSwimlane**([IRPSwimlane](#) itsSwimlane)

>   Specifies the swimlane that the action should be in
>   **Parameters:**
>   itsSwimlane - the swimlane that the action should be in

setReferenceToActivity

void **setReferenceToActivity**([IRPModelElement](#) referenceToActivity)

For call behavior elements, sets the activity that is referenced by the element.
Parameters:
referenceToActivity - the activity that should be referenced by the call behavior element

setStateType

void **setStateType**(java.lang.String stateType)

Specifies the type of the state
Parameters:
stateType - the type of the state. The valid strings for this parameter are: "And", "Or" (for a state that is not an "And" state), "LocalTermination" (for Termination State), "Block" (for Action Block), "Action", "SubActivity", "EventState" (for Send Action), and "FlowFinal"

setStaticReaction

void **setStaticReaction**(java.lang.String trigVal,
                          java.lang.String guardVal,
                          java.lang.String actionVal)

Adds a new internal transition to the state.
Parameters:
trigVal - the trigger to set for the internal transition
guardVal - the guard to set for the internal transition
actionVal - the action to set for the internal transition

unoverrideInheritance

void **unoverrideInheritance**()

Restores the inheritance relationship between this state and the corresponding state from the statechart of the class that this class is derived from. This method is used to restore the relationship that was severed with the method overrideInheritance().

---

com.telelogic.rhapsody.core
Interface IRPStatechart

All Superinterfaces:
> IRPClass, IRPClassifier, IRPModelElement, IRPUnit

All Known Subinterfaces:
> IRPFlowchart

---

public interface **IRPStatechart**
extends IRPClass

The IRPStatechart interface represents the statechart elements underlying a statechart. The statechart itself is represented by the IRPStatechartDiagram interface. You can create an IRPStatechart object with the method IRPClass.addStatechart().

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPGraphElement	**addFreeShapeByType**(java.lang.String metaType, IRPCollection xPoints, IRPCollection yPoints) Adds a free shape of the type specified, using the x coordinates and y coordinates provided.
IRPGraphElement	**addImage**(java.lang.String filename, int xPosition, int yPosition, int nWidth, int nHeight) Adds an image to the statechart, using the specified file, starting point, width, and height.
IRPGraphEdge	**addNewEdgeByType**(java.lang.String metaType, IRPGraphElement src, int xSrcPosition, int ySrcPosition, IRPGraphElement trg, int xTrgPosition, int yTrgPosition) Adds a connector element of the specified type to the statechart, using the source and target elements specified.
IRPGraphEdge	**addNewEdgeForElement**(IRPModelElement element, IRPGraphNode src, int xSrcPosition, int ySrcPosition, IRPGraphNode trg,

Method Summary

	`int xTrgPosition, int yTrgPosition)` Adds a connector graphical element to the statechart to represent the specified model element.
[IRPGraphNode](#)	**addNewNodeByType**(java.lang.String metaType, int xPosition, `int yPosition, int nWidth, int nHeight)` Adds a statechart element of the specified type to the statechart, using the position and dimensions specified.
[IRPGraphNode](#)	**addNewNodeForElement**([IRPModelElement](#) element, int xPosition, `int yPosition, int nWidth, int nHeight)` Adds a graphical element to the statechart to represent the specified model element.
[IRPGraphElement](#)	**addTextBox**(java.lang.String text, int xPosition, int yPosition, `int nWidth, int nHeight)` Adds a text box using the specified text, starting point, width, and height.
void	**closeDiagram**() Closes the statechart.
void	**createGraphics**() Creates the graphical representation of the elements in the statechart.
void	**deleteState**([IRPState](#) state) Deletes the specified state from the statechart.
int	**findTrigger**([IRPInterfaceItem](#) Item) Checks whether the specified IRPInterfaceItem element serves as the trigger of a transition in the statechart.
[IRPCollection](#)	**getAllTriggers**() Returns a collection of all the triggers in the statechart
[IRPCollection](#)	**getElementsInDiagram**() Returns a collection of all of the elements in the statechart.
[IRPCollection](#)	**getGraphicalElements**() Returns a collection of all the graphical elements in the statechart.
[IRPStatechart](#)	**getInheritsFrom**() Returns the statechart of the base class of this class.
int	**getIsMainBehavior**() Checks whether the statechart is the main behavior for the class.
int	**getIsOverridden**() Checks whether the inheritance relationship between this statechart and the statechart of the base class was overridden.
[IRPClassifier](#)	**getItsClass**() Returns the class that the statechart is associated with.
void	**getPicture**(java.lang.String filename) Saves the statechart as an emf format file, using the path and filename provided as a parameter.

Method Summary

[IRPCollection](#)	**getPictureAs**(java.lang.String firstFileName, java.lang.String imageFormat, int getImageMaps, [IRPCollection](#) diagrammap) Saves the statechart in the specified graphic format, breaking the diagram into a number of files if necessary.
[IRPCollection](#)	**getPictureAsDividedMetafiles**(java.lang.String firstFileName) Saves the statechart as an emf format file, breaking the diagram into a number of such files if necessary.
[IRPCollection](#)	**getPicturesWithImageMap**(java.lang.String firstFileName, [IRPCollection](#) diagrammap) Saves the statechart as an emf format file, breaking the diagram into a number of files if necessary.
[IRPState](#)	**getRootState**() Returns the root state of the statechart.
[IRPStatechartDiagram](#)	**getStatechartDiagram**() Returns the IRPStatechartDiagram object associated with the statechart.
[IRPAXViewCtrl](#)	**openDiagramView**() Used internally by Rational Rhapsody to display diagrams within Eclipse (when using the Rhapsody-Eclipse platform integration).
void	**overrideInheritance**() Breaks the inheritance relationship between this statechart and the statechart of the base class.
void	**populateDiagram**([IRPCollection](#) elementsToPopulate, [IRPCollection](#) relationsTypes, java.lang.String createContent) Populates the statechart with the elements and types of relations specified.
void	**setAsMainBehavior**() Specifies that this statechart should be the main behavior for the class.
void	**setShowDiagramFrame**(int bShow) Shows/hides the diagram frame.
void	**unoverrideInheritance**() Restores the inheritance relationship between this statechart and the statechart of the base class.

Methods inherited from interface com.telelogic.rhapsody.core.[IRPClass](#)

[addClass](#), [addConstructor](#), [addDestructor](#), [addEventReception](#), [addLink](#), [addLinkToPartViaPort](#), [addReception](#), [addSuperclass](#), [addTriggeredOperation](#), [addType](#), [deleteClass](#), [deleteConstructor](#), [deleteDestructor](#), [deleteEventReception](#), [deleteReception](#), [deleteSuperclass](#), [deleteType](#), [getIsAbstract](#), [getIsActive](#), [getIsBehaviorOverriden](#), [getIsComposite](#), [getIsFinal](#), [getIsReactive](#), [setIsAbstract](#), [setIsActive](#), [setIsBehaviorOverriden](#), [setIsFinal](#)

Methods inherited from interface com.telelogic.rhapsody.core.[IRPClassifier](#)

Methods inherited from interface com.telelogic.rhapsody.core.IRPClassifier

addActivityDiagram, addAttribute, addFlowItems, addFlows, addGeneralization, addOperation, addRelation, addRelationTo, addStatechart, addUnidirectionalRelation, addUnidirectionalRelationTo, deleteAttribute, deleteFlowItems, deleteFlows, deleteGeneralization, deleteOperation, deleteRelation, findAttribute, findBaseClassifier, findDerivedClassifier, findGeneralization, findInterfaceItem, findNestedClassifier, findNestedClassifierRecursive, findRelation, findTrigger, getActivityDiagram, getAttributes, getAttributesIncludingBases, getBaseClassifiers, getBehavioralDiagrams, getDerivedClassifiers, getFlowItems, getFlows, getGeneralizations, getInterfaceItems, getInterfaceItemsIncludingBases, getLinks, getNestedClassifiers, getOperations, getPorts, getRelations, getRelationsIncludingBases, getSequenceDiagrams, getSourceArtifacts, getStatechart

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addFreeShapeByType

```
IRPGraphElement addFreeShapeByType(java.lang.String metaType,
                                   IRPCollection xPoints,
                                   IRPCollection yPoints)
```

Adds a free shape of the type specified, using the x coordinates and y coordinates provided.

Parameters:

metaType - the type of shape to add. The possible values for this parameter are: "Polyline", "Polygon", "Rectangle", "Polycurve", "Closed Curve", "Ellipse".

xPoints - collection of integers representing the x coordinates for the shape

yPoints - collection of integers representing the y coordinates for the shape

Returns:

the new shape that was created

addImage

```
IRPGraphElement addImage(java.lang.String filename,
                         int xPosition,
                         int yPosition,
                         int nWidth,
                         int nHeight)
```

Adds an image to the statechart, using the specified file, starting point, width, and height.

Parameters:

filename - the full path to the image

xPosition - the x coordinate for the top left corner of the image, in pixels

yPosition - the y coordinate for the top left corner of the image, in pixels

nWidth - the width of the image, in pixels

nHeight - the height of the image, in pixels

Returns:

the new image element that was created

addNewEdgeByType

```
IRPGraphEdge addNewEdgeByType(java.lang.String metaType,
                              IRPGraphElement src,
                              int xSrcPosition,
                              int ySrcPosition,
                              IRPGraphElement trg,
                              int xTrgPosition,
                              int yTrgPosition)
```

Adds a connector element of the specified type to the statechart, using the source and target elements specified. Note that this method can only be used for connector elements that only have graphical representations and are not actual elements in the model. "Ordinary" connector elements are added to a statechart by carrying out two steps: 1) adding the new element to your model 2) adding a graphical representation of the element to the statechart using the method IRPStatechart.addNewEdgeForElement.

Parameters:

metaType - the type of connector element to add to the statechart. The strings that can be used for this parameter are: "anchor", "compRealization", "Containment Arrow", and "communication path".

src - the graphical element that is the source for the connector

xSrcPosition - the distance, in pixels, from the the left edge of the diagram to a point within the source graphical element

ySrcPosition - the distance, in pixels, from the the top edge of the diagram to a point within the source graphical element

trg - the graphical element that is the target for the connector

xTrgPosition - the distance, in pixels, from the the left edge of the diagram to a point within the target graphical element

yTrgPosition - the distance, in pixels, from the the top edge of the diagram to a point within the target graphical element

Returns:

the graphical element that was added to the statechart

See Also:

addNewEdgeForElement(com.telelogic.rhapsody.core.IRPModelElement, com.telelogic.rhapsody.core.IRPGraphNode, int, int, com.telelogic.rhapsody.core.IRPGraphNode, int, int)

---

addNewEdgeForElement

IRPGraphEdge **addNewEdgeForElement**(IRPModelElement element,
                                    IRPGraphNode src,
                                    int xSrcPosition,
                                    int ySrcPosition,
                                    IRPGraphNode trg,
                                    int xTrgPosition,
                                    int yTrgPosition)

Adds a connector graphical element to the statechart to represent the specified model element.

Parameters:

element - the model element to add to the statechart.

src - the graphical element that is the source for the connector

xSrcPosition - the distance, in pixels, from the the left edge of the diagram to a point within the source graphical element

ySrcPosition - the distance, in pixels, from the the top edge of the diagram to a point within the source graphical element

trg - the graphical element that is the target for the connector

xTrgPosition - the distance, in pixels, from the the left edge of the diagram to a point within the target graphical element

yTrgPosition - the distance, in pixels, from the the top edge of the diagram to a point within the target graphical element

Returns:

the connector graphical element that was added to the statechart

---

addNewNodeByType

IRPGraphNode **addNewNodeByType**(java.lang.String metaType,
                                 int xPosition,

addNewEdgeByType   592

```
                          int yPosition,
                          int nWidth,
                          int nHeight)
```

Adds a statechart element of the specified type to the statechart, using the position and dimensions specified. Note that this method can only be used for statechart elements that only have graphical representations and are not actual elements in the model. "Ordinary" model elements are added to a statechart by carrying out two steps: 1) adding the new element to your model 2) adding a graphical representation of the element to the statechart using the method IRPStatechart.addNewNodeForElement.

Parameters:

> `metaType` - the type of element to add to the diagram. The strings that can be used for this parameter are: "OrState"(for And Line), "Note"; panel diagram elements: "Knob", "Gauge", "Meter", "LevelIndicator", "MatrixDisplay", "DigitalDisplay", "Led", "OnOffSwitch", "PushButton", "ButtonArray", "TextBox", "Slider"; free shapes: "Polyline", "Ploygon", "Rectangle", "Polycurve", "Closed Curve","Ellipse","Image".
>
> `xPosition` - the position of the left edge of the graphical object, in pixels, relative to the left edge of the diagram
>
> `yPosition` - the position of the top edge of the graphical object, in pixels, relative to the top edge of the diagram
>
> `nWidth` - the width of the graphical object
>
> `nHeight` - the height of the graphical object

Returns:

> the graphical element that was added to the statechart

See Also:

> addNewNodeForElement(com.telelogic.rhapsody.core.IRPModelElement, int, int, int, int)

---

addNewNodeForElement

IRPGraphNode **addNewNodeForElement**(IRPModelElement element,
```
                                      int xPosition,
                                      int yPosition,
                                      int nWidth,
                                      int nHeight)
```

Adds a graphical element to the statechart to represent the specified model element. For connector elements, use the method addNewEdgeForElement.

Parameters:

> `element` - the model element to add to the statechart.
>
> `xPosition` - the position of the left edge of the graphical object, in pixels, relative to the left edge of the diagram
>
> `yPosition` - the position of the top edge of the graphical object, in pixels, relative to the top edge of the diagram
>
> `nWidth` - the width of the graphical object
>
> `nHeight` - the height of the graphical object

Returns:

> the graphical element that was added to the statechart

---

addTextBox

[IRPGraphElement](#) **addTextBox**(java.lang.String text,
                             int xPosition,
                             int yPosition,
                             int nWidth,
                             int nHeight)

Adds a text box using the specified text, starting point, width, and height.

Parameters:
text - the text that should be displayed

xPosition - the x coordinate for the top left corner of the box, in pixels

yPosition - the y coordinate for the top left corner of the box, in pixels

nWidth - the width of the text box, in pixels

nHeight - the height of the text box, in pixels

Returns:
the new text box that was created

openDiagramView

[IRPAXViewCtrl](#) **openDiagramView**()

Used internally by Rational Rhapsody to display diagrams within Eclipse (when using the Rhapsody-Eclipse platform integration).

closeDiagram

void **closeDiagram**()

Closes the statechart.

createGraphics

void **createGraphics**()

Creates the graphical representation of the elements in the statechart. When you create a statechart with the API, the graphical representation is not created by default. This means that the first time you open the statechart in Rational Rhapsody, you will be asked if the graphics should be created. You can create the graphical representation directly by calling createGraphics().

deleteState

void **deleteState**([IRPState](#) state)

Deletes the specified state from the statechart.

Parameters:
state - the state to delete

addTextBox 594

findTrigger

int **findTrigger**(<u>IRPInterfaceItem</u> Item)

> Checks whether the specified IRPInterfaceItem element serves as the trigger of a transition in the statechart.
> **Parameters:**
> > Item - the IRPInterfaceItem element to check
> **Returns:**
> > 1 if the specified element serves as the trigger of a transition in the statechart, 0 otherwise

getAllTriggers

<u>IRPCollection</u> **getAllTriggers**()

> Returns a collection of all the triggers in the statechart
> **Returns:**
> > all of the triggers in the statechart

getElementsInDiagram

<u>IRPCollection</u> **getElementsInDiagram**()

> Returns a collection of all of the elements in the statechart.
> **Returns:**
> > all of the elements in the statechart

getGraphicalElements

<u>IRPCollection</u> **getGraphicalElements**()

> Returns a collection of all the graphical elements in the statechart.
> **Returns:**
> > collection of IRPGraphElement objects, representing all the graphical elements in the statechart.

getInheritsFrom

<u>IRPStatechart</u> **getInheritsFrom**()

> Returns the statechart of the base class of this class.
> **Returns:**
> > the statechart of the base class of this class

getIsMainBehavior

int **getIsMainBehavior**()

Checks whether the statechart is the main behavior for the class. Rational Rhapsody allows you to define multiple statecharts and activities. One of these is defined as the "main" behavior, which is executed and can then reference other statecharts and activities.

Returns:

1 if the statechart is the main behavior, 0 otherwise

getIsOverridden

int **getIsOverridden**()

Checks whether the inheritance relationship between this statechart and the statechart of the base class was overridden.

Returns:

1 if the inheritance relationship between the statecharts was overridden, 0 if the relationship still exists.

getItsClass

IRPClassifier **getItsClass**()

Returns the class that the statechart is associated with.

Returns:

the class that the statechart is associated with

getPicture

void **getPicture**(java.lang.String filename)

Saves the statechart as an emf format file, using the path and filename provided as a parameter.

Parameters:

filename - the full path to use for saving the file

getPictureAs

IRPCollection **getPictureAs**(java.lang.String firstFileName,
                         java.lang.String imageFormat,
                         int getImageMaps,
                         IRPCollection diagrammap)

Saves the statechart in the specified graphic format, breaking the diagram into a number of files if necessary. The need to break the diagram into a number of files is based on the value of the property General:Graphics:ExportedDiagramScale. If the property is set to a value other than FitToOnePage,

more than one file will be created. In addition, this method can be used to retrieve diagram element information that can be used to create an HTML image map.

Parameters:

firstFileName - the name to use for the file created. If more than one file is created, the filenames used will be based on the following convention: firstFileNameZ_X_Y, where Z is the number of the created file, X is the number of the page along the X vector, and Y is the number of the page along the Y vector.

imageFormat - the graphic format in which the diagram should be saved. This can be one of the following: EMF, BMP, JPEG, JPG, TIFF.

getImageMaps - use this argument to indicate whether the method should also provide a collection of IRPImageMap objects that can be used to construct an HTML image map for the diagram. (Use 1 if you want this information, else use 0.)

diagrammap - The collection to use to store the IRPImageMap objects containing the required information for constructing an HTML image map

Returns:

collection that contains the names of the files that were created

---

getPictureAsDividedMetafiles

[IRPCollection](#) **getPictureAsDividedMetafiles**(java.lang.String firstFileName)

Saves the statechart as an emf format file, breaking the diagram into a number of such files if necessary. The need to break the diagram into a number of files is based on the value of the property General:Graphics:ExportedDiagramScale. If the property is set to a value other than FitToOnePage, more than one file will be created.

Parameters:

firstFileName - the name to use for the first file created. If more than one file is created, the filenames used will be based on the following convention: firstFileNameZ_X_Y, where Z is the number of the created file, X is the number of the page along the X vector, and Y is the number of the page along the Y vector.

Returns:

collection that contains the names of the files that were created

---

getPicturesWithImageMap

[IRPCollection](#) **getPicturesWithImageMap**(java.lang.String firstFileName,
                                  [IRPCollection](#) diagrammap)

Saves the statechart as an emf format file, breaking the diagram into a number of files if necessary. The need to break the diagram into a number of files is based on the value of the property General:Graphics:ExportedDiagramScale. If the property is set to a value other than FitToOnePage, more than one file will be created. In addition, this method retrieves diagram element information that can be used to create an HTML image map.

Parameters:

firstFileName - the name to use for the file created. If more than one file is created, the filenames used will be based on the following convention: firstFileNameZ_X_Y, where Z is the number of the created file, X is the number of the page along the X vector, and Y is the number of the page along the Y vector.

diagrammap - The collection to use to store the IRPImageMap objects containing the required information for constructing an HTML image map
Returns:
collection that contains the names of the files that were created

getRootState

IRPState **getRootState**()

Returns the root state of the statechart. To create a top-level state in a statechart, you add it to the root state.
Returns:
the root state of the statechart

getStatechartDiagram

IRPStatechartDiagram **getStatechartDiagram**()

Returns the IRPStatechartDiagram object associated with the statechart.
Returns:
the IRPStatechartDiagram object associated with the statechart

overrideInheritance

void **overrideInheritance**()

Breaks the inheritance relationship between this statechart and the statechart of the base class.

populateDiagram

void **populateDiagram**(IRPCollection elementsToPopulate,
                         IRPCollection relationsTypes,
                         java.lang.String createContent)

Populates the statechart with the elements and types of relations specified.
Parameters:
elementsToPopulate - the elements (nodes) to add to the diagram
relationsTypes - the types of relations that should be drawn on the diagram. You can use the string AllRelations to display all types, or use any combination of the following strings: Composition, Association, Link, Dependency, Inheritance, Anchor, InformationFlow
createContent - the elements that should be included in addition to those specified. This argument can take any of the following strings: among, from, to, fromto. If you use "among", only the elements you specified will be included. If you use one of the other strings, the diagram will also include elements that the selected elements are related to

setAsMainBehavior

void **setAsMainBehavior**()

> Specifies that this statechart should be the main behavior for the class. Rational Rhapsody allows you to define multiple statecharts and activities. One of these is defined as the "main" behavior, which is executed and can then reference other statecharts and activities.

setShowDiagramFrame

void **setShowDiagramFrame**(int bShow)

> Shows/hides the diagram frame.
> **Parameters:**
> > bShow - use 1 to show the diagram frame, 0 to hide the frame.

unoverrideInheritance

void **unoverrideInheritance**()

> Restores the inheritance relationship between this statechart and the statechart of the base class.

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES**   **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPStatechartDiagram

All Superinterfaces:
> IRPDiagram, IRPModelElement, IRPUnit

All Known Subinterfaces:
> IRPActivityDiagram

---

public interface **IRPStatechartDiagram**
extends IRPDiagram

The IRPStatechartDiagram interface represents statecharts in a Rational Rhapsody model.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPCollection	**addAndLine**(IRPGraphNode sourceState, int xStartPosition, int yStartPosition, int xEndPosition, int yEndPosition) Adds an And Line to the specified state.
void	**createGraphics**() Creates the graphical representation of the elements in the statechart.
IRPStatechart	**getStatechart**() Returns the IRPStatechart object underlying the statechart.

Methods inherited from interface com.telelogic.rhapsody.core.IRPDiagram
addFreeShapeByType, addImage, addNewEdgeByType, addNewEdgeForElement, addNewNodeByType, addNewNodeForElement, addTextBox, closeDiagram, completeRelations, getCorrespondingGraphicElements, getElementsInDiagram, getGraphicalElements, getLastVisualizationModifiedTime, getPicture, getPictureAs, getPictureAsDividedMetafiles, getPictureEx, getPicturesWithImageMap, isOpen, isShowDiagramFrame, openDiagram, openDiagramView, populateDiagram, removeGraphElements, setShowDiagramFrame

---

unoverrideInheritance   600

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addAndLine

```
IRPCollection addAndLine(IRPGraphNode sourceState,
                    int xStartPosition,
                    int yStartPosition,
                    int xEndPosition,
                    int yEndPosition)
```

Adds an And Line to the specified state.

Parameters:

sourceState - the graphical element representing the state to which the And Line should be added

com.telelogic.rhapsody.coreInterface IRPStatechartDiagram

xStartPosition - the x position at which the And Line should begin

yStartPosition - the y position at which the And Line should begin

xEndPosition - the x position at which the And Line should end

yEndPosition - the y position at which the And Line should end

Returns:

a collection of the new orthogonal states created

```
IRPApplication app = RhapsodyAppServer.getActiveRhapsodyApplication();
        IRPProject prj = app.activeProject();
        IRPPackage vehiclePackage = prj.addPackage("Vehicles");
        IRPClass carClass = vehiclePackage.addClass("Car");
        IRPStatechart carStatechart = carClass.addStatechart();
        IRPState rootState = carStatechart.getRootState();
        IRPState runningState = rootState.addState("Running");
        IRPStatechartDiagram scDiagram = carStatechart.getStatechartDiagram();
        IRPGraphNode runningStateNode = scDiagram.addNewNodeForElement(runningState, 100, 100, 400,
        IRPCollection stateNodesCreated = scDiagram.addAndLine(runningStateNode, 300, 100, 300, 500
        IRPGraphNode newStateNodeCreated = null;
        for (int stateNodeCounter = 1; stateNodeCounter < stateNodesCreated.getCount()+1;stateNodeC
                newStateNodeCreated = (IRPGraphNode)stateNodesCreated.getItem(stateNodeCounter);
                System.out.println(newStateNodeCreated.getModelObject().getName());
        }
```

createGraphics

void **createGraphics**()

Creates the graphical representation of the elements in the statechart. When you create a statechart with the API, the graphical representation is not created by default. This means that the first time you open the statechart in Rational Rhapsody, you will be asked if the graphics should be created. You can create the graphical representation directly by calling createGraphics().

getStatechart

IRPStatechart **getStatechart**()

Returns the IRPStatechart object underlying the statechart.
Returns:
the IRPStatechart object underlying the statechart

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

PREV CLASS   **NEXT CLASS**   **FRAMES**   **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPStateVertex

All Superinterfaces:
> IRPModelElement

All Known Subinterfaces:
> IRPAcceptEventAction, IRPAcceptTimeEvent, IRPCallOperation, IRPConnector, IRPObjectNode,
> IRPPin, IRPState

---

public interface **IRPStateVertex**
extends IRPModelElement

The IRPStateVertex interface represents the characteristics that are shared by various statechart elements such as states, join/fork connectors, and condition connectors.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPTransition	**addFlow**(java.lang.String type, IRPStateVertex to) Adds a control flow or object flow from this element to the specified element.
IRPTransition	**addTransition**(IRPStateVertex to) Adds a transition from this element to the specified element.
void	**deleteTransition**(IRPTransition transition) Deletes the specified transition.
IRPCollection	**getInTransitions**() Returns all of the transitions that enter the element.
IRPCollection	**getOutTransitions**() Returns all of the transitions that exit the element.
IRPState	**getParent**() Returns the element's parent.
void	

getStatechart   603

Method Summary

	setParent([IRPState](#) parent) Sets the parent state of the element.

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addFlow

[IRPTransition](#) **addFlow**(java.lang.String type,
                    [IRPStateVertex](#) to)

Adds a control flow or object flow from this element to the specified element.

Parameters:

type - the type of flow to create - the valid strings that can be used are ControlFlow and ObjectFlow

to - the target element for the new flow

Returns:

the flow created

addTransition

IRPTransition **addTransition**(IRPStateVertex to)

> Adds a transition from this element to the specified element.
> **Parameters:**
> > to - the target element for the new transition
> **Returns:**
> > the transition created

deleteTransition

void **deleteTransition**(IRPTransition transition)

> Deletes the specified transition.
> **Parameters:**
> > transition - the transition to delete

getInTransitions

IRPCollection **getInTransitions**()

> Returns all of the transitions that enter the element. Note that if there are any internal transitions defined, they will also be included in the collection that is returned. If you want to identify which transitions are internal, you can use the method IRPTransition.isStaticReaction().
> **Returns:**
> > all the transitions that enter the element (collection of IRPTransition elements).

getOutTransitions

IRPCollection **getOutTransitions**()

> Returns all of the transitions that exit the element. Note that if there are any internal transitions defined, they will also be included in the collection that is returned. If you want to identify which transitions are internal, you can use the method IRPTransition.isStaticReaction().
> **Returns:**
> > all the transitions that exit the element (collection of IRPTransition elements).

getParent

IRPState **getParent**()

> Returns the element's parent. If the element is not contained in a specific state, the root state of the diagram is returned.
> **Returns:**
> > the element's parent

setParent

```
void setParent(IRPState parent)
```

Sets the parent state of the element.

Parameters:

parent - the state that should serve as the parent of the element

com.telelogic.rhapsody.core
Interface IRPStereotype

All Superinterfaces:
> IRPClassifier, IRPModelElement, IRPUnit

---

public interface **IRPStereotype**
extends IRPClassifier

The IRPStereotype interface represents sterotypes in Rational Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

void	**addMetaClass**(java.lang.String metaClass) Adds a metaclass to the list of metaclasses that the stereotype can be applied to.
java.lang.String	**getIcon**() Gets the full path for the image file that is associated with this stereotype.
int	**getIsNewTerm**() Checks whether the stereotype is a "new term" stereotype.
java.lang.String	**getOfMetaClass**() Gets the names of the metaclasses that the stereotype can be applied to.
void	**removeMetaClass**(java.lang.String metaClass) Removes a metaclass from the list of metaclasses that the stereotype can be applied to.
void	**setIsNewTerm**(int isNewTerm) Used to change a stereotype to a "new term" stereotype, or change a "new term" stereotype to an ordinary stereotype.

Methods inherited from interface com.telelogic.rhapsody.core.IRPClassifier
addActivityDiagram, addAttribute, addFlowItems, addFlows, addGeneralization, addOperation, addRelation, addRelationTo, addStatechart, addUnidirectionalRelation,

setParent   607

Methods inherited from interface com.telelogic.rhapsody.core.**IRPClassifier**
addUnidirectionalRelationTo, deleteAttribute, deleteFlowItems, deleteFlows, deleteGeneralization, deleteOperation, deleteRelation, findAttribute, findBaseClassifier, findDerivedClassifier, findGeneralization, findInterfaceItem, findNestedClassifier, findNestedClassifierRecursive, findRelation, findTrigger, getActivityDiagram, getAttributes, getAttributesIncludingBases, getBaseClassifiers, getBehavioralDiagrams, getDerivedClassifiers, getFlowItems, getFlows, getGeneralizations, getInterfaceItems, getInterfaceItemsIncludingBases, getLinks, getNestedClassifiers, getOperations, getPorts, getRelations, getRelationsIncludingBases, getSequenceDiagrams, getSourceArtifacts, getStatechart

Methods inherited from interface com.telelogic.rhapsody.core.**IRPUnit**
copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addMetaClass

void **addMetaClass**(java.lang.String metaClass)

> Adds a metaclass to the list of metaclasses that the stereotype can be applied to.
> **Parameters:**
>> metaClass - the metaclass to add. Note that this string parameter can only contain the name of one metaclass. Adding multiple metaclasses requires multiple calls of this method.

getIcon

java.lang.String **getIcon**()

> Gets the full path for the image file that is associated with this stereotype.
> **Returns:**
>> the full path for the image file that is associated with this stereotype

getIsNewTerm

int **getIsNewTerm**()

> Checks whether the stereotype is a "new term" stereotype. For more information about "new terms", see the help for customizing Rhapsody.
> **Returns:**
>> indication of whether the stereotype is a "new term". 1 means that the stereotype is a "new term", 0 means that the stereotype is not a "new term".

getOfMetaClass

java.lang.String **getOfMetaClass**()

> Gets the names of the metaclasses that the stereotype can be applied to.
> **Returns:**
>> the names of the metaclasses that the stereotype can be applied to. If there is more than one such metaclass, the string returned will consist of a comma-separated list of the names.

removeMetaClass

void **removeMetaClass**(java.lang.String metaClass)

> Removes a metaclass from the list of metaclasses that the stereotype can be applied to.
> **Parameters:**
>> metaClass - the metaclass to remove. Note that this string parameter can only contain the name of one metaclass. Removing multiple metaclasses requires multiple calls of this method.

setIsNewTerm

```
void setIsNewTerm(int isNewTerm)
```

Used to change a stereotype to a "new term" stereotype, or change a "new term" stereotype to an ordinary stereotype. For more information about "new terms", see the help for customizing Rhapsody.

Parameters:

isNewTerm - Use 1 to change the stereotype to a "new term" stereotype. Use 0 to change a "new term" stereotype to an ordinary stereotype.

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPStructureDiagram

All Superinterfaces:
> IRPDiagram, IRPModelElement, IRPUnit

---

public interface **IRPStructureDiagram**
extends IRPDiagram

The IRPStructureDiagram interface represents structure diagrams in a Rational Rhapsody model.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement

IRPModelElement.OSLCLink

Method Summary

Methods inherited from interface com.telelogic.rhapsody.core.IRPDiagram

addFreeShapeByType, addImage, addNewEdgeByType, addNewEdgeForElement, addNewNodeByType, addNewNodeForElement, addTextBox, closeDiagram, completeRelations, getCorrespondingGraphicElements, getElementsInDiagram, getGraphicalElements, getLastVisualizationModifiedTime, getPicture, getPictureAs, getPictureAsDividedMetafiles, getPictureEx, getPicturesWithImageMap, isOpen, isShowDiagramFrame, openDiagram, openDiagramView, populateDiagram, removeGraphElements, setShowDiagramFrame

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype,

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core Interface IRPStructureDiagram   612

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

com.telelogic.rhapsody.core
Interface IRPSwimlane

All Superinterfaces:
> IRPModelElement

---

public interface **IRPSwimlane**
extends IRPModelElement

The IRPSwimlane interface represents swimlanes in an activity diagram.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPSwimlane	**addSwimlane**(java.lang.String name) method addSwimlane
IRPCollection	**getContents**() get property contents
IRPModelElement	**getRepresents**() get property represents
IRPCollection	**getSwimlanes**() get property swimlanes
void	**setRepresents**(IRPModelElement represents) set property represents

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies,

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addSwimlane

IRPSwimlane **addSwimlane**(java.lang.String name)

> method addSwimlane
> **Throws:**
> > RhapsodyRuntimeException

getContents

IRPCollection **getContents**()

> get property contents
> **Throws:**
> > RhapsodyRuntimeException

getRepresents

IRPModelElement **getRepresents**()

> get property represents
> **Throws:**
> > RhapsodyRuntimeException

getSwimlanes

<u>IRPCollection</u> **getSwimlanes**()

> get property swimlanes
> **Throws:**
> > <u>RhapsodyRuntimeException</u>

setRepresents

void **setRepresents**(<u>IRPModelElement</u> represents)

> set property represents
> **Throws:**
> > <u>RhapsodyRuntimeException</u>

com.telelogic.rhapsody.core
Interface IRPSysMLPort

All Superinterfaces:
> IRPInstance, IRPModelElement, IRPRelation, IRPUnit

---

public interface **IRPSysMLPort**
extends IRPInstance

The IRPSysMLPort interface represents flowport elements in Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPLink	**addLink**(IRPInstance fromPart, IRPInstance toPart, IRPRelation assoc, IRPSysMLPort toPort, IRPPackage newOwner) This method is used to create a link between flowports on two parts.
int	**getIsReversed**() Checks whether the flowport was specified as conjugated.
java.lang.String	**getPortDirection**() Returns the direction that was specified for the flowport.
IRPClassifier	**getType**() Returns the type that was specified for the flowport.
void	**setIsReversed**(int isReversed) Specifies whether the flowport should be conjugated
void	**setPortDirection**(java.lang.String portDirection) Sets the direction of the flowport.
void	**setType**(IRPClassifier type) Sets the type for the flowport.

Methods inherited from interface com.telelogic.rhapsody.core.IRPInstance

---

setRepresents   616

Methods inherited from interface com.telelogic.rhapsody.core.IRPInstance
addRelationToTheWhole, getAttributeValue, getInLinks, getInstantiatedBy, getListOfInitializerArguments, getOutLinks, setAttributeValue, setExplicit, setImplicit, setInitializerArgumentValue, setInstantiatedBy

Methods inherited from interface com.telelogic.rhapsody.core.IRPRelation
addQualifier, getAssociationClass, getInverse, getIsNavigable, getIsSymmetric, getMultiplicity, getObjectAsObjectType, getOfClass, getOtherClass, getQualifier, getQualifiers, getQualifierType, getRelationLabel, getRelationLinkName, getRelationRoleName, getRelationType, getVisibility, isTypelessObject, makeUnidirect, removeQualifier, setInverse, setIsNavigable, setMultiplicity, setOfClass, setOtherClass, setQualifier, setQualifierType, setRelationLabel, setRelationLinkName, setRelationRoleName, setRelationType

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit
copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addLink

```
IRPLink addLink(IRPInstance fromPart,
                IRPInstance toPart,
                IRPRelation assoc,
                IRPSysMLPort toPort,
                IRPPackage newOwner)
```

This method is used to create a link between flowports on two parts.

Parameters:

fromPart - the "from" part for the link

toPart - the "to" part for the link

assoc - use Null for this argument (it is not relevant for links between flowports)

toPort - the "to" port for the link

newOwner - the package that should be the owner of the link created

Returns:

the link that was created

Throws:

RhapsodyRuntimeException

getIsReversed

```
int getIsReversed()
```

Checks whether the flowport was specified as conjugated.

Returns:

1 if the flowport was specified as conjugated, 0 otherwise

Throws:

RhapsodyRuntimeException

getPortDirection

```
java.lang.String getPortDirection()
```

Returns the direction that was specified for the flowport.

Returns:

the direction that was specified for the flowport - will be one of the following values: "In", "Out", "InOut"

Throws:

RhapsodyRuntimeException

getType

```
IRPClassifier getType()
```

Returns the type that was specified for the flowport.
Returns:
>   the type that was specified for the flowport
Throws:
>   RhapsodyRuntimeException

---

setIsReversed

void **setIsReversed**(int isReversed)

Specifies whether the flowport should be conjugated
Parameters:
>   isReversed - use 1 to specify that the flowport should be conjugated, 0 otherwise
Throws:
>   RhapsodyRuntimeException

---

setPortDirection

void **setPortDirection**(java.lang.String portDirection)

Sets the direction of the flowport.
Parameters:
>   portDirection - the direction to use for the flowport. The valid values are "In", "Out", and
>   "InOut".
Throws:
>   RhapsodyRuntimeException

---

setType

void **setType**(IRPClassifier type)

Sets the type for the flowport.
Parameters:
>   type - the type to use for the flowport
Throws:
>   RhapsodyRuntimeException

---

com.telelogic.rhapsody.core
Class IRPTableLayout.Column.AnnotationAttribute

```
java.lang.Object
  └ com.telelogic.rhapsody.core.IRPTableLayout.Column.AnnotationAttribute
```

Enclosing class:
> IRPTableLayout.Column

---

public static final class **IRPTableLayout.Column.AnnotationAttribute**
extends java.lang.Object

Contains values to be used for Property parameter of addColumn method, when AnnotationAttribute is selected for the Type parameter of addColumn method.

Field Summary

static java.lang.String	**ID** Value to be used for Property parameter of addColumn method, when AnnotationAttribute is selected for the Type parameter of addColumn method.
static java.lang.String	**SPECIFICATION** Value to be used for Property parameter of addColumn method, when AnnotationAttribute is selected for the Type parameter of addColumn method.

Constructor Summary

IRPTableLayout.Column.AnnotationAttribute()

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

setType   620

Field Detail

ID

```
public static final java.lang.String ID
```

> Value to be used for Property parameter of addColumn method, when AnnotationAttribute is selected for the Type parameter of addColumn method.

> **See Also:**
> > [Constant Field Values](#)

SPECIFICATION

```
public static final java.lang.String SPECIFICATION
```

> Value to be used for Property parameter of addColumn method, when AnnotationAttribute is selected for the Type parameter of addColumn method.

> **See Also:**
> > [Constant Field Values](#)

Constructor Detail

IRPTableLayout.Column.AnnotationAttribute

```
public IRPTableLayout.Column.AnnotationAttribute()
```

[Package](#)	**Class**	**[Use](#)**	**[Tree](#)**	**[Serialized](#)**	**[Deprecated](#)**	**[Index](#)**	**[Help](#)**

[PREV CLASS](#)   [NEXT CLASS](#)   **[FRAMES](#)   [NO FRAMES](#)   [All Classes](#)**

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)   DETAIL: [FIELD](#) | [CONSTR](#) | METHOD

com.telelogic.rhapsody.coreClass IRPTableLayout.Column.AnnotationAttribute   621

com.telelogic.rhapsody.core
Class IRPTableLayout.Column.DependsOn

```
java.lang.Object
  └─ com.telelogic.rhapsody.core.IRPTableLayout.Column.DependsOn
```

Enclosing class:
> IRPTableLayout.Column

---

public static final class **IRPTableLayout.Column.DependsOn**
extends java.lang.Object

Contains the pre-defined values to be used for Property parameter of addColumn method, when DependsOn is selected for the Type parameter of addColumn method. Other legal values for this field are names of Streotypes applicable to Dependency.

Field Summary

static java.lang.String	**DEPENDENCY** Value to be used for Property parameter of addColumn method, when FlowAttribute is selected for the Type parameter of addColumn method.

Constructor Summary

IRPTableLayout.Column.DependsOn()

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

DEPENDENCY

```
public static final java.lang.String DEPENDENCY
```

Value to be used for Property parameter of addColumn method, when FlowAttribute is selected for the Type parameter of addColumn method.

See Also:
Constant Field Values

Constructor Detail

IRPTableLayout.Column.DependsOn

```
public IRPTableLayout.Column.DependsOn()
```

com.telelogic.rhapsody.core
Class IRPTableLayout.Column.FlowAttribute

```
java.lang.Object
  └─com.telelogic.rhapsody.core.IRPTableLayout.Column.FlowAttribute
```

Enclosing class:
> IRPTableLayout.Column

---

public static final class **IRPTableLayout.Column.FlowAttribute**
extends java.lang.Object

Contains values to be used for Property parameter of addColumn method, when FlowAttribute is selected for the Type parameter of addColumn method.

Field Summary

static java.lang.String	**ITEM FLOWS** Value to be used for Property parameter of addColumn method, when FlowAttribute is selected for the Type parameter of addColumn method.

Constructor Summary

IRPTableLayout.Column.FlowAttribute()

Method Summary

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Field Detail

ITEM_FLOWS

`public static final java.lang.String` **`ITEM_FLOWS`**

> Value to be used for Property parameter of addColumn method, when FlowAttribute is selected for the Type parameter of addColumn method.

> **See Also:**
> [Constant Field Values](#)

Constructor Detail

IRPTableLayout.Column.FlowAttribute

`public` **`IRPTableLayout.Column.FlowAttribute`**`()`

---

com.telelogic.rhapsody.core
Class IRPTableLayout.Column.GeneralAttribute

```
java.lang.Object
  └─ com.telelogic.rhapsody.core.IRPTableLayout.Column.GeneralAttribute
```

Enclosing class:
> IRPTableLayout.Column

---

public static final class **IRPTableLayout.Column.GeneralAttribute**
extends java.lang.Object

Contains values to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.

Field Summary

static java.lang.String	**CLASSIFIER** Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.
static java.lang.String	**DESCRIPTION** Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.
static java.lang.String	**ELEMENT_TYPE** Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.
static java.lang.String	**FULL_PATH_NAME**
static java.lang.String	**LABEL** Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.
static java.lang.String	**NAME** Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.
static java.lang.String	**OWNER** Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.

Field Summary

static java.lang.String	**STEREOTYPES** Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.
static java.lang.String	**VALUE**

Constructor Summary

IRPTableLayout.Column.GeneralAttribute()

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

NAME

public static final java.lang.String **NAME**

> Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.

> **See Also:**
> Constant Field Values

LABEL

public static final java.lang.String **LABEL**

> Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.

> **See Also:**
> Constant Field Values

OWNER

`public static final java.lang.String **OWNER**`

Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.

See Also:
[Constant Field Values](#)

DESCRIPTION

`public static final java.lang.String **DESCRIPTION**`

Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.

See Also:
[Constant Field Values](#)

ELEMENT_TYPE

`public static final java.lang.String **ELEMENT_TYPE**`

Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.

See Also:
[Constant Field Values](#)

STEREOTYPES

`public static final java.lang.String **STEREOTYPES**`

Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.

See Also:
[Constant Field Values](#)

CLASSIFIER

`public static final java.lang.String **CLASSIFIER**`

Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.

See Also:
[Constant Field Values](#)

FULL_PATH_NAME

```
public static final java.lang.String FULL_PATH_NAME
```

See Also:
[Constant Field Values](#)

VALUE

```
public static final java.lang.String VALUE
```

See Also:
[Constant Field Values](#)

Constructor Detail

IRPTableLayout.Column.GeneralAttribute

```
public IRPTableLayout.Column.GeneralAttribute()
```

com.telelogic.rhapsody.core
Class IRPTableLayout.Column

```
java.lang.Object
  └─com.telelogic.rhapsody.core.IRPTableLayout.Column
```

Enclosing interface:
> IRPTableLayout

---

public static final class **IRPTableLayout.Column**
extends java.lang.Object

This class holds constant values to be used with addColumn method.

Nested Class Summary	
static class	**IRPTableLayout.Column.AnnotationAttribute** Contains values to be used for Property parameter of addColumn method, when AnnotationAttribute is selected for the Type parameter of addColumn method.
static class	**IRPTableLayout.Column.DependsOn** Contains the pre-defined values to be used for Property parameter of addColumn method, when DependsOn is selected for the Type parameter of addColumn method.
static class	**IRPTableLayout.Column.FlowAttribute** Contains values to be used for Property parameter of addColumn method, when FlowAttribute is selected for the Type parameter of addColumn method.
static class	**IRPTableLayout.Column.GeneralAttribute** Contains values to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.
static class	**IRPTableLayout.Column.ImplementationCellType**
static class	**IRPTableLayout.Column.RelationAttributeFrom** Contains values to be used for Property parameter of addColumn method, when RelationAttributeFrom is selected for the Type parameter of addColumn method.
static class	**IRPTableLayout.Column.RelationAttributeTo** Contains values to be used for Property parameter of addColumn method, when RelationAttributeTo is selected for the Type parameter of addColumn method.
static class	

Nested Class Summary

	IRPTableLayout.Column.RequirementAttribute Contains values to be used for Property parameter of addColumn method, when RequirementAttribute is selected for the Type parameter of addColumn method.
static class	**IRPTableLayout.Column.UserDefinedMethod** Contains values to be used for Property parameter of addColumn method, when USER_DEFINED_METHOD is selected for the Type parameter of addColumn method.

Field Summary

static java.lang.String	**ANNOTATION_ATTRIBUTE** Value used for Type parameter of addColumn method
static java.lang.String	**DEPENDS_ON** Value used for Type parameter of addColumn method.
static java.lang.String	**FLOW_ATTRIBUTE** Value used for Type parameter of addColumn method
static java.lang.String	**GENERAL_ATTRIBUTE** Value used for Type parameter of addColumn method
static java.lang.String	**RELAION_ATTRIBUTE_FROM** **Deprecated.**
static java.lang.String	**RELAION_ATTRIBUTE_TO** **Deprecated.**
static java.lang.String	**RELATION_ATTRIBUTE_FROM** Value used for Type parameter of addColumn method
static java.lang.String	**RELATION_ATTRIBUTE_TO** Value used for Type parameter of addColumn method
static java.lang.String	**REQUIREMENT_ATTRIBUTE** Value used for Type parameter of addColumn method
static java.lang.String	**TAG** Value used for Type parameter of addColumn method
static java.lang.String	**TAG_EDIT** Value used for Type parameter of addColumn method
static java.lang.String	**TAG_EDIT_STRICT** Value used for Type parameter of addColumn method
static java.lang.String	**USER_DEFINED_METHOD** Value used for Type parameter of addColumn method.

Constructor Summary

IRPTableLayout.Column()	

Method Summary

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

Field Detail

GENERAL_ATTRIBUTE

```
public static final java.lang.String GENERAL_ATTRIBUTE
```

Value used for Type parameter of addColumn method

See Also:
Constant Field Values

RELAION_ATTRIBUTE_FROM

```
public static final java.lang.String RELAION_ATTRIBUTE_FROM
```

Deprecated.
Value used for Type parameter of addColumn method

See Also:
Constant Field Values

RELAION_ATTRIBUTE_TO

```
public static final java.lang.String RELAION_ATTRIBUTE_TO
```

Deprecated.
Value used for Type parameter of addColumn method

See Also:
Constant Field Values

REQUIREMENT_ATTRIBUTE

```
public static final java.lang.String REQUIREMENT_ATTRIBUTE
```

Value used for Type parameter of addColumn method

ANNOTATION_ATTRIBUTE

public static final java.lang.String **ANNOTATION_ATTRIBUTE**

Value used for Type parameter of addColumn method

FLOW_ATTRIBUTE

public static final java.lang.String **FLOW_ATTRIBUTE**

Value used for Type parameter of addColumn method

TAG

public static final java.lang.String **TAG**

Value used for Type parameter of addColumn method

TAG_EDIT

public static final java.lang.String **TAG_EDIT**

Value used for Type parameter of addColumn method

TAG_EDIT_STRICT

public static final java.lang.String **TAG_EDIT_STRICT**

Value used for Type parameter of addColumn method

See Also:
[Constant Field Values](#)

DEPENDS_ON

`public static final java.lang.String` **`DEPENDS_ON`**

Value used for Type parameter of addColumn method. When using this value - for the "Type" parameter, the "Property" parameter can be set by one of the values defined in IRPTableLayout.Column.DependsOn, or by the name of Stereptype applicable to Dependency.

See Also:
[Constant Field Values](#)

USER_DEFINED_METHOD

`public static final java.lang.String` **`USER_DEFINED_METHOD`**

Value used for Type parameter of addColumn method. When this value is used - the value for the property parameter can be set to the plugin method to be executed, or to the contant defined in IRPTableLayout.Column.UserDefinedMethod.

See Also:
[Constant Field Values](#)

RELATION_ATTRIBUTE_FROM

`public static final java.lang.String` **`RELATION_ATTRIBUTE_FROM`**

Value used for Type parameter of addColumn method

See Also:
[Constant Field Values](#)

RELATION_ATTRIBUTE_TO

`public static final java.lang.String` **`RELATION_ATTRIBUTE_TO`**

Value used for Type parameter of addColumn method

See Also:
[Constant Field Values](#)

Constructor Detail

com.telelogic.rhapsody.core

IRPTableLayout.Column

```
public IRPTableLayout.Column()
```

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Class IRPTableLayout.Column.ImplementationCellType

```
java.lang.Object
  └─ com.telelogic.rhapsody.core.IRPTableLayout.Column.ImplementationCellType
```

Enclosing class:
> IRPTableLayout.Column

---

public static final class **IRPTableLayout.Column.ImplementationCellType**
extends java.lang.Object

Field Summary

static java.lang.String	**LIST OF MODEL ELEMENTS** Value to be used for cellType parameter of SetColumnImplementationCellType method.
static java.lang.String	**MODEL ELEMENT** Value to be used for cellType parameter of SetColumnImplementationCellType method.
static java.lang.String	**STRING** Value to be used for cellType parameter of SetColumnImplementationCellType method.

Constructor Summary

IRPTableLayout.Column.ImplementationCellType()

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

STRING

`public static final java.lang.String **STRING**`

>   Value to be used for cellType parameter of SetColumnImplementationCellType method.

>   **See Also:**
>>   [Constant Field Values](#)

MODEL_ELEMENT

`public static final java.lang.String **MODEL_ELEMENT**`

>   Value to be used for cellType parameter of SetColumnImplementationCellType method.

>   **See Also:**
>>   [Constant Field Values](#)

LIST_OF_MODEL_ELEMENTS

`public static final java.lang.String **LIST_OF_MODEL_ELEMENTS**`

>   Value to be used for cellType parameter of SetColumnImplementationCellType method.

>   **See Also:**
>>   [Constant Field Values](#)

Constructor Detail

IRPTableLayout.Column.ImplementationCellType

`public **IRPTableLayout.Column.ImplementationCellType**()`

---

com.telelogic.rhapsody.core
Class IRPTableLayout.Column.RelationAttributeFrom

```
java.lang.Object
  └─com.telelogic.rhapsody.core.IRPTableLayout.Column.RelationAttributeFrom
```

Enclosing class:
  IRPTableLayout.Column

---

public static final class **IRPTableLayout.Column.RelationAttributeFrom**
extends java.lang.Object

Contains values to be used for Property parameter of addColumn method, when RelationAttributeFrom is selected for the Type parameter of addColumn method.

Field Summary

static java.lang.String	**FROM_ELEMENT** Value to be used for Property parameter of addColumn method, when RelationAttributeFrom is selected for the Type parameter of addColumn method.
static java.lang.String	**PORT_PROVIDED_INERFACE** Value to be used for Property parameter of addColumn method, when RelationAttributeFrom is selected for the Type parameter of addColumn method.
static java.lang.String	**PORT_REQUIRED_INERFACE** Value to be used for Property parameter of addColumn method, when RelationAttributeFrom is selected for the Type parameter of addColumn method.
static java.lang.String	**PROVIDED_INERFACE_OPERATIONS** Value to be used for Property parameter of addColumn method, when RelationAttributeFrom is selected for the Type parameter of addColumn method.
static java.lang.String	**REQUIRED_INERFACE_OPERATIONS** Value to be used for Property parameter of addColumn method, when RelationAttributeFrom is selected for the Type parameter of addColumn method.
static java.lang.String	**VIA_PORT** Value to be used for Property parameter of addColumn method, when RelationAttributeFrom is selected for the Type parameter of addColumn method.

Constructor Summary

Constructor Summary

[IRPTableLayout.Column.RelationAttributeFrom](#)()

Method Summary

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

VIA_PORT

public static final java.lang.String **VIA_PORT**

> Value to be used for Property parameter of addColumn method, when RelationAttributeFrom is selected for the Type parameter of addColumn method.

> **See Also:**
> [Constant Field Values](#)

PORT_PROVIDED_INERFACE

public static final java.lang.String **PORT_PROVIDED_INERFACE**

> Value to be used for Property parameter of addColumn method, when RelationAttributeFrom is selected for the Type parameter of addColumn method.

> **See Also:**
> [Constant Field Values](#)

FROM_ELEMENT

public static final java.lang.String **FROM_ELEMENT**

> Value to be used for Property parameter of addColumn method, when RelationAttributeFrom is selected for the Type parameter of addColumn method.

> **See Also:**
> [Constant Field Values](#)

PORT_REQUIRED_INERFACE

`public static final java.lang.String` **`PORT_REQUIRED_INERFACE`**

Value to be used for Property parameter of addColumn method, when RelationAttributeFrom is selected for the Type parameter of addColumn method.

See Also:
[Constant Field Values](#)

PROVIDED_INERFACE_OPERATIONS

`public static final java.lang.String` **`PROVIDED_INERFACE_OPERATIONS`**

Value to be used for Property parameter of addColumn method, when RelationAttributeFrom is selected for the Type parameter of addColumn method.

See Also:
[Constant Field Values](#)

REQUIRED_INERFACE_OPERATIONS

`public static final java.lang.String` **`REQUIRED_INERFACE_OPERATIONS`**

Value to be used for Property parameter of addColumn method, when RelationAttributeFrom is selected for the Type parameter of addColumn method.

See Also:
[Constant Field Values](#)

Constructor Detail

IRPTableLayout.Column.RelationAttributeFrom

`public` **`IRPTableLayout.Column.RelationAttributeFrom`**`()`

---

com.telelogic.rhapsody.core
Class IRPTableLayout.Column.RelationAttributeTo

```
java.lang.Object
  └─ com.telelogic.rhapsody.core.IRPTableLayout.Column.RelationAttributeTo
```

Enclosing class:
> IRPTableLayout.Column

public static final class **IRPTableLayout.Column.RelationAttributeTo**
extends java.lang.Object

Contains values to be used for Property parameter of addColumn method, when RelationAttributeTo is selected for the Type parameter of addColumn method.

Field Summary

static java.lang.String	**PORT_PROVIDED_INERFACE** Value to be used for Property parameter of addColumn method, when RelationAttributeTo is selected for the Type parameter of addColumn method.
static java.lang.String	**PORT_REQUIRED_INERFACE** Value to be used for Property parameter of addColumn method, when RelationAttributeTo is selected for the Type parameter of addColumn method.
static java.lang.String	**PROVIDED_INERFACE_OPERATIONS** Value to be used for Property parameter of addColumn method, when RelationAttributeTo is selected for the Type parameter of addColumn method.
static java.lang.String	**REQUIRED_INERFACE_OPERATIONS** Value to be used for Property parameter of addColumn method, when RelationAttributeTo is selected for the Type parameter of addColumn method.
static java.lang.String	**TO_ELEMENT** Value to be used for Property parameter of addColumn method, when RelationAttributeTo is selected for the Type parameter of addColumn method.
static java.lang.String	**VIA_PORT** Value to be used for Property parameter of addColumn method, when RelationAttributeTo is selected for the Type parameter of addColumn method.

Constructor Summary

Constructor Summary

IRPTableLayout.Column.RelationAttributeTo()

Method Summary

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

VIA_PORT

public static final java.lang.String **VIA_PORT**

> Value to be used for Property parameter of addColumn method, when RelationAttributeTo is selected for the Type parameter of addColumn method.
>
> **See Also:**
> Constant Field Values

PORT_PROVIDED_INERFACE

public static final java.lang.String **PORT_PROVIDED_INERFACE**

> Value to be used for Property parameter of addColumn method, when RelationAttributeTo is selected for the Type parameter of addColumn method.
>
> **See Also:**
> Constant Field Values

TO_ELEMENT

public static final java.lang.String **TO_ELEMENT**

> Value to be used for Property parameter of addColumn method, when RelationAttributeTo is selected for the Type parameter of addColumn method.
>
> **See Also:**
> Constant Field Values

PORT_REQUIRED_INERFACE

`public static final java.lang.String **PORT_REQUIRED_INERFACE**`

> Value to be used for Property parameter of addColumn method, when RelationAttributeTo is selected for the Type parameter of addColumn method.
>
> **See Also:**
> > [Constant Field Values](#)

PROVIDED_INERFACE_OPERATIONS

`public static final java.lang.String **PROVIDED_INERFACE_OPERATIONS**`

> Value to be used for Property parameter of addColumn method, when RelationAttributeTo is selected for the Type parameter of addColumn method.
>
> **See Also:**
> > [Constant Field Values](#)

REQUIRED_INERFACE_OPERATIONS

`public static final java.lang.String **REQUIRED_INERFACE_OPERATIONS**`

> Value to be used for Property parameter of addColumn method, when RelationAttributeTo is selected for the Type parameter of addColumn method.
>
> **See Also:**
> > [Constant Field Values](#)

Constructor Detail

IRPTableLayout.Column.RelationAttributeTo

`public **IRPTableLayout.Column.RelationAttributeTo**()`

com.telelogic.rhapsody.core
Class IRPTableLayout.Column.RequirementAttribute

```
java.lang.Object
  └─com.telelogic.rhapsody.core.IRPTableLayout.Column.RequirementAttribute
```

Enclosing class:
> IRPTableLayout.Column

---

public static final class **IRPTableLayout.Column.RequirementAttribute**
extends java.lang.Object

Contains values to be used for Property parameter of addColumn method, when RequirementAttribute is selected for the Type parameter of addColumn method.

Field Summary

static java.lang.String	**ID** Value to be used for Property parameter of addColumn method, when RequirementAttribute is selected for the Type parameter of addColumn method.
static java.lang.String	**LINK_FROM**
static java.lang.String	**LINK_FROM_FULLNAME**
static java.lang.String	**LINK_SUSPECT**
static java.lang.String	**LINK_TYPE**
static java.lang.String	**SPECIFICATION** Value to be used for Property parameter of addColumn method, when RequirementAttribute is selected for the Type parameter of addColumn method.

Constructor Summary

IRPTableLayout.Column.RequirementAttribute()	

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

ID

public static final java.lang.String **ID**

> Value to be used for Property parameter of addColumn method, when RequirementAttribute is selected for the Type parameter of addColumn method.
>
> **See Also:**
> Constant Field Values

SPECIFICATION

public static final java.lang.String **SPECIFICATION**

> Value to be used for Property parameter of addColumn method, when RequirementAttribute is selected for the Type parameter of addColumn method.
>
> **See Also:**
> Constant Field Values

LINK_TYPE

public static final java.lang.String **LINK_TYPE**

> **See Also:**
> Constant Field Values

LINK_FROM

public static final java.lang.String **LINK_FROM**

> **See Also:**
> Constant Field Values

LINK_FROM_FULLNAME

public static final java.lang.String **LINK_FROM_FULLNAME**

> **See Also:**
> [Constant Field Values](#)

LINK_SUSPECT

public static final java.lang.String **LINK_SUSPECT**

> **See Also:**
> [Constant Field Values](#)

Constructor Detail

IRPTableLayout.Column.RequirementAttribute

public **IRPTableLayout.Column.RequirementAttribute**()

[Package](#) Class [Use](#) [Tree](#) [Serialized](#) [Deprecated](#) [Index](#) [Help](#)
[PREV CLASS](#) [NEXT CLASS](#)   **[FRAMES](#) [NO FRAMES](#) [All Classes](#)**
SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)   DETAIL: [FIELD](#) | [CONSTR](#) | METHOD

com.telelogic.rhapsody.core
Class IRPTableLayout.Column.UserDefinedMethod

```
java.lang.Object
  └─com.telelogic.rhapsody.core.IRPTableLayout.Column.UserDefinedMethod
```

Enclosing class:
> IRPTableLayout.Column

---

public static final class **IRPTableLayout.Column.UserDefinedMethod**
extends java.lang.Object

Contains values to be used for Property parameter of addColumn method, when
USER_DEFINED_METHOD is selected for the Type parameter of addColumn method.

Field Summary

static java.lang.String	**Implementation** Use this value to declare that a dynamic java code was set to be executed for this column.

Constructor Summary

IRPTableLayout.Column.UserDefinedMethod()

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

Implementation

`public static final java.lang.String` **`Implementation`**

Use this value to declare that a dynamic java code was set to be executed for this column.

See Also:
[Constant Field Values](#)

Constructor Detail

IRPTableLayout.Column.UserDefinedMethod

`public` **`IRPTableLayout.Column.UserDefinedMethod`**`()`

com.telelogic.rhapsody.core
Interface IRPTableLayout

All Superinterfaces:
> IRPModelElement, IRPUnit

```
public interface IRPTableLayout
extends IRPUnit
```

Nested Class Summary

static class	**IRPTableLayout.Column** This class holds constant values to be used with addColumn method.
static class	**IRPTableLayout.QueryOrElementsList** This class contains constant values for use with the methods setFromElementTypesUseQueryOrElementsList and setToElementTypesUseQueryOrElementsList.

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

void	**addColumn**(java.lang.String type, java.lang.String Property, java.lang.String ColumnName) Adds a new column to the table layout.
int	**addColumnEx**(java.lang.String type, java.lang.String Property, java.lang.String ColumnName, java.lang.String Context) Adds a new column to the table layout.
int	**getCollapseFirstColumn**() Checks whether the first column of the layout includes controls for collapsing and expanding rows that have the same value in the first column.
java.lang.String	**getColumnContext**(int Index) Returns the context pattern label that was specified for the column.
int	**getColumnCount**() Returns the number of columns in the table layout.
int	

Method Summary

	getColumnDefaultWidth(int Index) Returns the default width that was defined for the specified column.
int	**getColumnImplementationAllowNew**(int Index) Checks whether the user-defined picker for the specified column includes the New option in its list.
int	**getColumnImplementationAllowSelect**(int Index) Checks whether the user-defined picker for the specified column includes the Select option in its list.
java.lang.String	**getColumnImplementationCellType**(int Index) Returns the type of information that is displayed in the column's cells - string, model element, or list of model elements.
java.lang.String	**getColumnImplementationDisplayProperty**(int Index) Returns the type of element information that is displayed when the cell value type is set to model element or list of model elements.
java.lang.String	**getColumnImplementationGetterCode**(int Index) Returns the Java code for the getter for the cells in the specified column.
java.lang.String	**getColumnImplementationImports**(int Index) For columns that use customized cell behavior, this method returns the list of imports specified for the column.
java.lang.String	**getColumnImplementationPickerCode**(int Index) Returns the Java code for the picker for the cells in the specified column.
java.lang.String	**getColumnImplementationSetterCode**(int Index) Returns the Java code for the setter for the cells in the specified column.
java.lang.String	**getColumnName**(int Index) Returns the name of the specified column.
java.lang.String	**getColumnProperty**(int Index) Returns the Property of the specified column.
[IRPCollection](#)	**getColumns**()
java.lang.String	**getColumnType**(int Index) Returns the type of the specified table column.
[IRPCollection](#)	**getElementTypes**() Returns a collection of the element types that were specified to be displayed in the table.
[IRPCollection](#)	**getFromElementTypes**() For "relation tables", returns a collection of the element types specified as the "from" element types.
[IRPTableLayout](#)	**getFromElementTypesQueryToUse**() For "relation tables", returns the query that was specified to determine the "from" element types.

Method Summary

int	**getFromElementTypesUseQueryOrElementsList**() For "relation tables", checks whether a query or collection of element types was used to specify the "from" element types.
int	**getRelationTable**() Checks whether the table was defined as a "relation table".
IRPCollection	**getResultList**(IRPModelElement scope) method GetResultList
IRPCollection	**getToElementTypes**() For "relation tables", returns a collection of the element types specified as the "to" element types.
IRPTableLayout	**getToElementTypesQueryToUse**() For "relation tables", returns the query that was specified to determine the "to" element types.
int	**getToElementTypesUseQueryOrElementsList**() For "relation tables", checks whether a query or collection of element types was used to specify the "to" element types.
void	**removeColumn**(int Index) Removes the specified column from the table layout.
void	**setCollapseFirstColumn**(int collapse) Specifies whether or not the fist column should include controls for collapsing and expanding rows that have the same value in the first column.
void	**setColumnContext**(int Index, java.lang.String Context) If you have defined a context pattern, this method can be used to specify a label from the context pattern, for the specified column.
void	**setColumnDefaultWidth**(int Index, int width) Sets the default width of the specified column.
void	**setColumnImplementationAllowNew**(int Index, int value) For columns that use customized cell behavior, this method can be used to include the New option in the list provided by the picker.
void	**setColumnImplementationAllowSelect**(int Index, int value) For columns that use customized cell behavior, this method can be used to include the Select option in the list provided by the picker.
void	**setColumnImplementationCellType**(int Index, java.lang.String cellType) For columns that use customized cell behavior, this method is used to specify the type of information that will be displayed in the column's cells - string, model element, or list of model elements.
void	**setColumnImplementationDisplayProperty**(int Index, java.lang.String propertyToDisplay) For columns that use customized cell behavior, this method is used to specify the type of element information that should be displayed when the cell value type is set to model element or list of model elements, for example, the name or value of the element.

Method Summary

void	**setColumnImplementationGetterCode**(int Index, java.lang.String code) For columns that use customized cell behavior, this method is used to specify the Java code for the getter for the cells in the column.
void	**setColumnImplementationImports**(int Index, java.lang.String imports) For columns that use customized cell behavior, this method can be used to specify classes required by your code.
void	**setColumnImplementationPickerCode**(int Index, java.lang.String code) For columns that use customized cell behavior, this method is used to specify the Java code for the picker for the cells in the column.
void	**setColumnImplementationSetterCode**(int Index, java.lang.String code) For columns that use customized cell behavior, this method is used to specify the Java code for the setter for the cells in the column.
void	**setColumnName**(int Index, java.lang.String name) Sets the name of the specified column.
void	**setColumnProperty**(int Index, java.lang.String Property) Sets the Property of the specified column.
void	**setColumnType**(int Index, java.lang.String type) Sets the type of the specified table column.
void	**setElementTypes**(IRPCollection elements) Specifies the list of element types that should be displayed in the table.
void	**setFromElementTypes**(IRPCollection elements) For "relation tables", specifies the list of element types to use as the "from" element types.
void	**setFromElementTypesQueryToUse**(IRPTableLayout query) For "relation tables", specifies the query to use to determine the "from" element types for the table layout.
void	**setFromElementTypesUseQueryOrElementsList**(int queryOrElementsList) For "relation tables", specifies whether a query or collection of element types should be used to determine the "from" element types for the table layout.
void	**setRelationTable**(int relation) Specifies whether the table should be defined as a "relation table".
void	**setToElementTypes**(IRPCollection elements) For "relation tables", specifies the list of element types to use as the "to" element types for the table layout.
void	**setToElementTypesQueryToUse**(IRPTableLayout query) For "relation tables", specifies the query to use to determine the "to" element types for the table layout.
void	**setToElementTypesUseQueryOrElementsList**(int queryOrElementsList) For "relation tables", specifies whether a query or collection of element types should be used to determine the "to" element types for the table layout.

Methods inherited from interface com.telelogic.rhapsody.core.**IRPUnit**
copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addColumn

```
void addColumn(java.lang.String type,
               java.lang.String Property,
               java.lang.String ColumnName)
```

> Adds a new column to the table layout.
> **Parameters:**
>> type - the column Type (equivalent to Type field in the UI) - the valid values for this parameter are the constants that are defined in the class IRPTableLayout.Column, for example, IRPTableLayout.Column.ANNOTATION_ATTRIBUTE.

addColumn 653

Property - the column Property (equivalent to Property field in the UI) - the valid values for this parameter are the constants defined in the classes nested beneath the class IRPTableLayout.Column. The nested class to use depends upon the value used for the "type" parameter.

For example, if the value of the "type" parameter is IRPTableLayout.Column.ANNOTATION_ATTRIBUTE, the valid values for the "Property" parameter are the constants defined in the nested class IRPTableLayout.Column.AnnotationAttribute, such as IRPTableLayout.Column.AnnotationAttribute.ID and IRPTableLayout.Column.AnnotationAttribute.SPECIFICATION.

However, if you specified TAG, TAG_EDIT, TAG_EDIT_STRICT, or USER_DEFINED_METHOD as the "type" parameter, you must provide a string that reflects the full path of the relevant tag or the name of the relevant plugin method.

For TAG, TAG_EDIT, and TAG_EDIT_STRICT, use the string that is returned when you call the method getFullPathName for the tag.

If you specified USER_DEFINED_METHOD as the type, the Property parameter should take the form pluginName.method. (If you do not specify the plugin name, all plugins will be searched until a method with the specified name is found).

ColumnName - the text to use as the heading for the column

addColumnEx

```
int addColumnEx(java.lang.String type,
                java.lang.String Property,
                java.lang.String ColumnName,
                java.lang.String Context)
```

Adds a new column to the table layout. Differs from the addColumn method in that it allows you to also specify a label from a context pattern and it returns the index of the new column added.

Parameters:

type - the type to use for the column (one of the constants defined in the class IRPTableLayout.Column, for example, IRPTableLayout.Column.GENERAL_ATTRIBUTE)

Property - the Property to use for the specified column. The values that can be used for this parameter are the constants defined in the classes nested under IRPTableLayout.Column, for example, IRPTableLayout.Column.GeneralAttribute.NAME. Note that the Property must match the column type. For example, if the type of the column was set to IRPTableLayout.Column.ANNOTATION_ATTRIBUTE, the available values for the Property of the column are the constants defined in the class IRPTableLayout.Column.AnnotationAttribute, such as IRPTableLayout.Column.AnnotationAttribute.ID and IRPTableLayout.Column.AnnotationAttribute.SPECIFICATION.

ColumnName - the text to use as the heading for the column

Context - a label from the context pattern that was defined. If you do not want to specify a context pattern label, use an empty string for this parameter.

Returns:

the index of the new column that was created (index of first column is 0)

Throws:
> RhapsodyRuntimeException

---

getCollapseFirstColumn

int **getCollapseFirstColumn**()

> Checks whether the first column of the layout includes controls for collapsing and expanding rows that have the same value in the first column.
> **Returns:**
> > 1 if the first column includes collapse/expand controls, 0 otherwise

---

getColumnContext

java.lang.String **getColumnContext**(int Index)

> Returns the context pattern label that was specified for the column.
> **Parameters:**
> > Index - the index of the column (index of first column is 0)
> **Returns:**
> > the context pattern label that was specified for the column
> **Throws:**
> > RhapsodyRuntimeException

---

getColumnDefaultWidth

int **getColumnDefaultWidth**(int Index)

> Returns the default width that was defined for the specified column.
> **Parameters:**
> > Index - the index of the column whose default width should be returned (index of first column is 0)
> **Returns:**
> > the default width defined for the specified column (in pixels)
> **Throws:**
> > RhapsodyRuntimeException

---

getColumnImplementationAllowNew

int **getColumnImplementationAllowNew**(int Index)

> Checks whether the user-defined picker for the specified column includes the New option in its list.
> **Parameters:**
> > Index - the index of the column (index of first column is 0)
> **Returns:**
> > 1 if the picker includes the New option, 0 if it does not

---

Throws:

        RhapsodyRuntimeException

---

getColumnImplementationAllowSelect

int **getColumnImplementationAllowSelect**(int Index)

Checks whether the user-defined picker for the specified column includes the Select option in its list.

Parameters:

        Index - the index of the column (index of first column is 0)

Returns:

        1 if the picker includes the Select option, 0 if it does not

Throws:

        RhapsodyRuntimeException

---

getColumnImplementationCellType

java.lang.String **getColumnImplementationCellType**(int Index)

Returns the type of information that is displayed in the column's cells - string, model element, or list of model elements. The value returned will be one of the constants defined in the class IRPTableLayout.Column.ImplementationCellType.

Parameters:

        Index - the index of the column (index of first column is 0)

Returns:

        the type of information that is displayed in the column's cells (one of the constants defined in the class IRPTableLayout.Column.ImplementationCellType, for example, IRPTableLayout.Column.ImplementationCellType.MODEL_ELEMENT)

Throws:

        RhapsodyRuntimeException

---

getColumnImplementationDisplayProperty

java.lang.String **getColumnImplementationDisplayProperty**(int Index)

Returns the type of element information that is displayed when the cell value type is set to model element or list of model elements. The value returned will be one of the constants defined in the class IRPTableLayout.Column.GeneralAttribute.

Parameters:

        Index - the index of the column (index of first column is 0)

Returns:

        the type of element information that is displayed when the cell value type is set to model element or list of model elements (one of the constants defined in the class IRPTableLayout.Column.GeneralAttribute, for example, IRPTableLayout.Column.GeneralAttribute.NAME)

Throws:

        RhapsodyRuntimeException

---

getColumnImplementationGetterCode

`java.lang.String` **`getColumnImplementationGetterCode`**`(int Index)`

Returns the Java code for the getter for the cells in the specified column.

Parameters:

`Index` - the index of the column (index of first column is 0)

Returns:

the Java code for the getter for the cells in the column

Throws:

RhapsodyRuntimeException

getColumnImplementationImports

`java.lang.String` **`getColumnImplementationImports`**`(int Index)`

For columns that use customized cell behavior, this method returns the list of imports specified for the column.

Parameters:

`Index` - the index of the column (index of first column is 0)

Returns:

comma-separated list of the imports specified for the column

Throws:

RhapsodyRuntimeException

getColumnImplementationPickerCode

`java.lang.String` **`getColumnImplementationPickerCode`**`(int Index)`

Returns the Java code for the picker for the cells in the specified column.

Parameters:

`Index` - the index of the column (index of first column is 0)

Returns:

the Java code for the picker for the cells in the column

Throws:

RhapsodyRuntimeException

getColumnImplementationSetterCode

`java.lang.String` **`getColumnImplementationSetterCode`**`(int Index)`

Returns the Java code for the setter for the cells in the specified column.

Parameters:

`Index` - the index of the column (index of first column is 0)

Returns:

the Java code for the setter for the cells in the column

Throws:

RhapsodyRuntimeException

getColumnName

`java.lang.String` **`getColumnName`**`(int Index)`

Returns the name of the specified column.

Parameters:

`Index` - the index of the column whose name should be returned (index of first column is 0)

Returns:

the name of the specified column

Throws:

RhapsodyRuntimeException

getColumnProperty

`java.lang.String` **`getColumnProperty`**`(int Index)`

Returns the Property of the specified column. Corresponds to the Property field on the Columns tab for table layouts. The value returned will be one of the constants defined in the classes nested under IRPTableLayout.Column, for example, IRPTableLayout.Column.GeneralAttribute.NAME.

Parameters:

`Index` - the index of the column (index of first column is 0)

Returns:

the Property of the specified column. Value returned will be one of the constants defined in the classes nested under IRPTableLayout.Column, for example, IRPTableLayout.Column.GeneralAttribute.NAME

Throws:

RhapsodyRuntimeException

getColumnType

`java.lang.String` **`getColumnType`**`(int Index)`

Returns the type of the specified table column. The value returned will be one of the constants defined in the class IRPTableLayout.Column.

Parameters:

`Index` - the index of the column (index of first column is 0)

Returns:

the type of the table column (one of the constants defined in the class IRPTableLayout.Column, for example, IRPTableLayout.Column.ANNOTATION_ATTRIBUTE)

Throws:

RhapsodyRuntimeException

getColumns

IRPCollection **getColumns**()
> **Returns:**
>> collection of columns
>
> **Throws:**
>> RhapsodyRuntimeException

---

getElementTypes

IRPCollection **getElementTypes**()

> Returns a collection of the element types that were specified to be displayed in the table. The collection consists of strings (from the list of types displayed on the ElementTypes tab of the Features window for table layouts).
>
> **Returns:**
>> the element types that were specified to be displayed in the table

---

getFromElementTypes

IRPCollection **getFromElementTypes**()

> For "relation tables", returns a collection of the element types specified as the "from" element types. The collection consists of strings (from the list of types displayed on the From Element Types tab of the Features window for table layouts).
>
> **Returns:**
>> the types specified as the "from" element types for the table layout

---

getFromElementTypesQueryToUse

IRPTableLayout **getFromElementTypesQueryToUse**()

> For "relation tables", returns the query that was specified to determine the "from" element types.
>
> **Returns:**
>> the query that was specified to determine the "from" element types for the table layout

---

getFromElementTypesUseQueryOrElementsList

int **getFromElementTypesUseQueryOrElementsList**()

> For "relation tables", checks whether a query or collection of element types was used to specify the "from" element types.
>
> **Returns:**
>> one of the constants contained in the class IRPTableLayout.QueryOrElementsList: QUERY if a query was used, ELEMENTS_LIST if a collection of element types was used.

---

getRelationTable

int **getRelationTable**()

> Checks whether the table was defined as a "relation table".
> **Returns:**
> > 1 if the table was defined as a "relation table", 0 otherwise

getResultList

IRPCollection **getResultList**(IRPModelElement scope)

> method GetResultList
> **Throws:**
> > RhapsodyRuntimeException

getToElementTypes

IRPCollection **getToElementTypes**()

> For "relation tables", returns a collection of the element types specified as the "to" element types. The collection consists of strings (from the list of types displayed on the To Element Types tab of the Features window for table layouts).
> **Returns:**
> > the types specified as the "to" element types for the table layout

getToElementTypesQueryToUse

IRPTableLayout **getToElementTypesQueryToUse**()

> For "relation tables", returns the query that was specified to determine the "to" element types.
> **Returns:**
> > the query that was specified to determine the "to" element types for the table layout

getToElementTypesUseQueryOrElementsList

int **getToElementTypesUseQueryOrElementsList**()

> For "relation tables", checks whether a query or collection of element types was used to specify the "to" element types.
> **Returns:**
> > one of the constants contained in the class IRPTableLayout.QueryOrElementsList: QUERY if a query was used, ELEMENTS_LIST if a collection of element types was used.

removeColumn

void **removeColumn**(int Index)

>Removes the specified column from the table layout.
>**Parameters:**
>>Index - the index representing the position of the column in the table. The index for the fist column in the table is 0.

setCollapseFirstColumn

void **setCollapseFirstColumn**(int collapse)

>Specifies whether or not the fist column should include controls for collapsing and expanding rows that have the same value in the first column.
>**Parameters:**
>>collapse - use 1 if the first column should include collapse/expand controls, 0 otherwise.

setColumnContext

void **setColumnContext**(int Index,
                       java.lang.String Context)

>If you have defined a context pattern, this method can be used to specify a label from the context pattern, for the specified column.
>**Parameters:**
>>Index - the index of the column (index of first column is 0)
>>Context - a label from the context pattern that was defined
>**Throws:**
>>RhapsodyRuntimeException

setColumnDefaultWidth

void **setColumnDefaultWidth**(int Index,
                           int width)

>Sets the default width of the specified column. If a user double-clicks the column border after manually changing the width, the width will return to this value.
>**Parameters:**
>>Index - the index of the column whose default width should be set (index of first column is 0)
>>width - the default width to use for the column (in pixels)
>**Throws:**
>>RhapsodyRuntimeException

setColumnImplementationAllowNew

void **setColumnImplementationAllowNew**(int Index,
                                        int value)

> For columns that use customized cell behavior, this method can be used to include the New option in the list provided by the picker.
> **Parameters:**
>> `Index` - the index of the column (index of first column is 0)
>> `value` - use `1` if the New option should be included in the list, `0` if it should not

setColumnImplementationAllowSelect

void **setColumnImplementationAllowSelect**(int Index,
                                           int value)

> For columns that use customized cell behavior, this method can be used to include the Select option in the list provided by the picker.
> **Parameters:**
>> `Index` - the index of the column (index of first column is 0)
>> `value` - use `1` if the Select option should be included in the list, `0` if it should not
> **Throws:**
>> RhapsodyRuntimeException

setColumnImplementationCellType

void **setColumnImplementationCellType**(int Index,
                                        java.lang.String cellType)

> For columns that use customized cell behavior, this method is used to specify the type of information that will be displayed in the column's cells - string, model element, or list of model elements.
> **Parameters:**
>> `Index` - the index of the column (index of first column is 0)
>> `cellType` - the type of information that will be displayed in the column's cells. The valid values for this parameter are the constants that are defined in the class IRPTableLayout.Column.ImplementationCellType, for example IRPTableLayout.Column.ImplementationCellType.MODEL_ELEMENT.
> **Throws:**
>> RhapsodyRuntimeException

setColumnImplementationDisplayProperty

void **setColumnImplementationDisplayProperty**(int Index,
                                               java.lang.String propertyToDisplay)

> For columns that use customized cell behavior, this method is used to specify the type of element information that should be displayed when the cell value type is set to model element or list of model elements, for example, the name or value of the element.

setColumnImplementationAllowNew   662

Parameters:

> Index - the index of the column (index of first column is 0)
>
> propertyToDisplay - the type of element information that should be displayed for the element or elements in the cell. The valid values for this parameter are the constants defined in the class IRPTableLayout.Column.GeneralAttribute, such as IRPTableLayout.Column.GeneralAttribute.NAME.

Throws:

> RhapsodyRuntimeException

---

setColumnImplementationGetterCode

```
void setColumnImplementationGetterCode(int Index,
                                       java.lang.String code)
```

For columns that use customized cell behavior, this method is used to specify the Java code for the getter for the cells in the column.

Parameters:

> Index - the index of the column (index of first column is 0)
>
> code - the Java code to use for the getter

Throws:

> RhapsodyRuntimeException

---

setColumnImplementationImports

```
void setColumnImplementationImports(int Index,
                                    java.lang.String imports)
```

For columns that use customized cell behavior, this method can be used to specify classes required by your code. Corresponds to the Imports field in the User Defined Implementation dialog. The list of imports should be comma-separated.

Parameters:

> Index - the index of the column (index of first column is 0)
>
> imports - a comma-separated list of classes to import

Throws:

> RhapsodyRuntimeException

---

setColumnImplementationPickerCode

```
void setColumnImplementationPickerCode(int Index,
                                       java.lang.String code)
```

For columns that use customized cell behavior, this method is used to specify the Java code for the picker for the cells in the column.

Parameters:

> Index - the index of the column (index of first column is 0)
>
> code - the Java code to use for the picker

Throws:

> RhapsodyRuntimeException

---

setColumnImplementationDisplayProperty 663

setColumnImplementationSetterCode

```
void setColumnImplementationSetterCode(int Index,
                                       java.lang.String code)
```

For columns that use customized cell behavior, this method is used to specify the Java code for the setter for the cells in the column.

Parameters:

Index - the index of the column (index of first column is 0)

code - the Java code to use for the setter

Throws:

[RhapsodyRuntimeException](#)

setColumnName

```
void setColumnName(int Index,
                   java.lang.String name)
```

Sets the name of the specified column.

Parameters:

Index - the index of the column whose name should be set (index of first column is 0)

name - the name to use for the column

Throws:

[RhapsodyRuntimeException](#)

setColumnProperty

```
void setColumnProperty(int Index,
                       java.lang.String Property)
```

Sets the Property of the specified column. Corresponds to the Property field on the Columns tab for table layouts.

Parameters:

Index - the index of the column (index of first column is 0)

Property - the Property to use for the specified column. The values that can be used for this parameter are the constants defined in the classes nested under [IRPTableLayout.Column](#), for example, IRPTableLayout.Column.GeneralAttribute.NAME. Note that the Property must match the column type. For example, if the type of the column was set to IRPTableLayout.Column.ANNOTATION_ATTRIBUTE, the available values for the Property of the column are the constants defined in the class IRPTableLayout.Column.AnnotationAttribute, such as IRPTableLayout.Column.AnnotationAttribute.ID and IRPTableLayout.Column.AnnotationAttribute.SPECIFICATION.

Throws:

[RhapsodyRuntimeException](#)

setColumnType

```
void setColumnType(int Index,
                   java.lang.String type)
```

Sets the type of the specified table column. The type must be one of the constants defined in the class IRPTableLayout.Column.

Parameters:

Index - the index of the column (index of first column is 0)

type - the type to use for the column (one of the constants defined in the class IRPTableLayout.Column, for example, IRPTableLayout.Column.GENERAL_ATTRIBUTE)

Throws:

RhapsodyRuntimeException

setElementTypes

```
void setElementTypes(IRPCollection elements)
```

Specifies the list of element types that should be displayed in the table. The parameter must be a collection of strings (from the list of types displayed on the ElementTypes tab of the Features window for table layouts).

Parameters:

elements - the element types that should be displayed in the table

setFromElementTypes

```
void setFromElementTypes(IRPCollection elements)
```

For "relation tables", specifies the list of element types to use as the "from" element types. The parameter must be a collection of strings (from the list of types displayed on the From Element Types tab of the Features window for table layouts).

Parameters:

elements - collection of element types to use as the "from" element types for the table layout

setFromElementTypesQueryToUse

```
void setFromElementTypesQueryToUse(IRPTableLayout query)
```

For "relation tables", specifies the query to use to determine the "from" element types for the table layout.

Parameters:

query - the query to use to determine the "from" element types for the table layout. To clear a previous query, use null for the parameter.

setFromElementTypesUseQueryOrElementsList

void **setFromElementTypesUseQueryOrElementsList**(int queryOrElementsList)

For "relation tables", specifies whether a query or collection of element types should be used to determine the "from" element types for the table layout.
Parameters:
>	queryOrElementsList - one of the constants contained in the class IRPTableLayout.QueryOrElementsList: QUERY if a query should be used, ELEMENTS_LIST if a collection of element types should be used.

setRelationTable

void **setRelationTable**(int relation)

Specifies whether the table should be defined as a "relation table".
Parameters:
>	relation - use 1 if the table should be defined as a "relation table", 0 otherwise.

setToElementTypes

void **setToElementTypes**([IRPCollection](#) elements)

For "relation tables", specifies the list of element types to use as the "to" element types for the table layout. The parameter must be a collection of strings (from the list of types displayed on the To Element Types tab of the Features window for table layouts).
Parameters:
>	elements - collection of element types to use as the "to" element types for the table layout

setToElementTypesQueryToUse

void **setToElementTypesQueryToUse**([IRPTableLayout](#) query)

For "relation tables", specifies the query to use to determine the "to" element types for the table layout.
Parameters:
>	query - the query to use to determine the "to" element types for the table layout. To clear a previous query, use null for the parameter.

setToElementTypesUseQueryOrElementsList

void **setToElementTypesUseQueryOrElementsList**(int queryOrElementsList)

For "relation tables", specifies whether a query or collection of element types should be used to determine the "to" element types for the table layout.
Parameters:

`queryOrElementsList` - one of the constants contained in the class
IRPTableLayout.QueryOrElementsList: QUERY if a query should be used,
ELEMENTS_LIST if a collection of element types should be used.

getColumnCount

int **getColumnCount**()

Returns the number of columns in the table layout.
Returns:
the number of columns in the table layout
Throws:
[RhapsodyRuntimeException](#)

---

com.telelogic.rhapsody.core
Class IRPTableLayout.QueryOrElementsList

```
java.lang.Object
  └─com.telelogic.rhapsody.core.IRPTableLayout.QueryOrElementsList
```

Enclosing interface:
> IRPTableLayout

---

public static final class **IRPTableLayout.QueryOrElementsList**
extends java.lang.Object

This class contains constant values for use with the methods setFromElementTypesUseQueryOrElementsList and setToElementTypesUseQueryOrElementsList.

Field Summary

static int	**ELEMENTS_LIST** When ELEMENTS_LIST is used as the parameter for the methods setFromElementTypesUseQueryOrElementsList and setToElementTypesUseQueryOrElementsList, it indicates that elements selected in the element types list will be used to specify the "from" element types or "to" element types for the relation table.
static int	**QUERY** When QUERY is used as the parameter for the methods setFromElementTypesUseQueryOrElementsList and setToElementTypesUseQueryOrElementsList, it indicates that a query is going to be used to determine the "from" element types or "to" element types for the relation table.

Constructor Summary

IRPTableLayout.QueryOrElementsList()	

Method Summary

Methods inherited from class java.lang.Object

getColumnCount   668

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

Field Detail

QUERY

```
public static final int QUERY
```

> When QUERY is used as the parameter for the methods
> setFromElementTypesUseQueryOrElementsList and setToElementTypesUseQueryOrElementsList, it
> indicates that a query is going to be used to determine the "from" element types or "to" element types
> for the relation table.
>
> **See Also:**
> > [Constant Field Values](#)

ELEMENTS_LIST

```
public static final int ELEMENTS_LIST
```

> When ELEMENTS_LIST is used as the parameter for the methods
> setFromElementTypesUseQueryOrElementsList and setToElementTypesUseQueryOrElementsList, it
> indicates that elements selected in the element types list will be used to specify the "from" element
> types or "to" element types for the relation table.
>
> **See Also:**
> > [Constant Field Values](#)

Constructor Detail

IRPTableLayout.QueryOrElementsList

```
public IRPTableLayout.QueryOrElementsList()
```

com.telelogic.rhapsody.core
Class IRPTableView.ContentFormat

```
java.lang.Object
  └─ com.telelogic.rhapsody.core.IRPTableView.ContentFormat
```

Enclosing interface:
> IRPTableView

---

public static final class **IRPTableView.ContentFormat**
extends java.lang.Object

This class contains values that specify export format

Field Summary

static java.lang.String	**CSV** Export in Comma Separated Value (CSV) format.
static java.lang.String	**HTML** Export in HTML format.
static java.lang.String	**XML** Export in XML format.

Constructor Summary

IRPTableView.ContentFormat()	

Method Summary

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

HTML

`public static final java.lang.String HTML`

Export in HTML format. Exported only string representations.

See Also:
Constant Field Values

XML

`public static final java.lang.String XML`

Export in XML format. For each model element, its GUID is exported as well.

See Also:
Constant Field Values

CSV

`public static final java.lang.String CSV`

Export in Comma Separated Value (CSV) format. Exported only string representations.

See Also:
Constant Field Values

Constructor Detail

IRPTableView.ContentFormat

`public IRPTableView.ContentFormat()`

Package **Class** Use Tree Serialized Deprecated Index Help

PREV CLASS NEXT CLASS   FRAMES NO FRAMES All Classes
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

HTML   671

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

com.telelogic.rhapsody.core
Interface IRPTableView

All Superinterfaces:
IRPModelElement, IRPUnit

public interface **IRPTableView**
extends IRPUnit

The IRPTableView interface represents Table View elements in Rhapsody models.

Nested Class Summary

static class	**IRPTableView.ContentFormat** This class contains values that specify export format

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

java.lang.String	**getContent**(java.lang.String format) Retrieves the content of the table in the specified format.
java.lang.String	**getHTMLContent**() Returns the content of the table as HTML.
IRPCollection	**getImageCollection**(java.lang.String sFolder, java.lang.String sFilename, java.lang.String sExtension) method GetImageCollection
int	**getIncludeDescendants**() get property includeDescendants
IRPTableLayout	**getItsTableLayout**() method GetItsTableLayout
IRPCollection	**getScope**() method GetScope
void	**open**() method open

Method Summary

void	**setIncludeDescendants**(int includeDescendants) set property includeDescendants
void	**setItsTableLayout**(IRPTableLayout pVal) Specifies the table layout to use for this table view.
void	**setScope**(IRPCollection pCollection) Specifies the scope to use for this table view.

Methods inherited from interface com.telelogic.rhapsody.core.**IRPUnit**

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

---

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

---

Method Detail

getContent

java.lang.String **getContent**(java.lang.String format)

Retrieves the content of the table in the specified format. The value of the parameter should be one of the values defined in the class IRPTableView.ContentFormat. Note that when you call this method, the table is also displayed in Rhapsody.

Parameters:

format - one of the formats defined in the class IRPTableView.ContentFormat, for example, IRPTableView.ContentFormat.CSV

Returns:

the content of the table in the specified format

getHTMLContent

java.lang.String **getHTMLContent**()

Returns the content of the table as HTML. The content returned begins and ends with the "table" tag. Note that when this method is called, the table is opened in Rational Rhapsody before the HTML is returned.

Returns:

the content of the table as HTML

getImageCollection

IRPCollection **getImageCollection**(java.lang.String sFolder,
                                     java.lang.String sFilename,
                                     java.lang.String sExtension)

method GetImageCollection
Throws:

RhapsodyRuntimeException

getItsTableLayout

IRPTableLayout **getItsTableLayout**()

method GetItsTableLayout
Throws:

RhapsodyRuntimeException

getScope

IRPCollection **getScope**()

method GetScope
Throws:

setItsTableLayout

void **setItsTableLayout**([IRPTableLayout](#) pVal)

> Specifies the table layout to use for this table view.
> **Parameters:**
> > pVal - the table layout to use for this table view

setScope

void **setScope**([IRPCollection](#) pCollection)

> Specifies the scope to use for this table view.
> **Parameters:**
> > pCollection - the scope to use for this table view. Note that the parameter is a Rhapsody collection, but at the moment, only the first value in the collection is used for the scope.

getIncludeDescendants

int **getIncludeDescendants**()

> get property includeDescendants
> **Throws:**
> > [RhapsodyRuntimeException](#)

open

void **open**()

> method open
> **Throws:**
> > [RhapsodyRuntimeException](#)

setIncludeDescendants

void **setIncludeDescendants**(int includeDescendants)

> set property includeDescendants
> **Throws:**
> > [RhapsodyRuntimeException](#)

com.telelogic.rhapsody.core

com.telelogic.rhapsody.core
Interface IRPTag

All Superinterfaces:
> IRPModelElement, IRPUnit, IRPVariable

---

public interface **IRPTag**
extends IRPVariable

The IRPTag interface represents tags in a Rational Rhapsody model. To create a new tag, use the method IRPModelElement.addNewAggr.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPTag	**getBase**() Returns the base tag on which the local copy of the tag is based.
IRPProfile	**getFromProfile**() For tags whose source is a profile that was added to the project (as opposed to tags defined locally in the project), this method returns the profile in which the tag was defined.
java.lang.String	**getMultiplicity**() Returns the multiplicity that was specified for the tag.
java.lang.String	**getTagMetaClass**() Returns the name of the metaclass to which the tag is applicable.
java.lang.String	**getValue**() Returns the value of the tag.
void	**setMultiplicity**(java.lang.String multiplicity) Specifies the multiplicity for the tag.
void	**setTagContextValue**(IRPCollection elements, IRPCollection multiplicities) Sets the value of the tag to a specific instance of another model element.

Method Summary

void	**setTagMetaClass**(java.lang.String tagMetaClass) Specifies the metaclass to which the tag should be applicable, for example, "Class".
void	**setValue**(java.lang.String value) Sets the value of the tag.

Methods inherited from interface com.telelogic.rhapsody.core.IRPVariable

addElementDefaultValue, addStringDefaultValue, getDeclaration, getDefaultValue, getType, getValueSpecifications, setDeclaration, setDefaultValue, setType, setTypeDeclaration

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getBase

[IRPTag](#) **getBase**()

> Returns the base tag on which the local copy of the tag is based. This method is relevant for tags that belong to stereotypes and tags that belong to profiles, but not for tags that were added locally to a specific model element.
>
> **Returns:**
> > the base tag on which the local copy of the tag is based

getFromProfile

[IRPProfile](#) **getFromProfile**()

> For tags whose source is a profile that was added to the project (as opposed to tags defined locally in the project), this method returns the profile in which the tag was defined.
>
> **Returns:**
> > the profile in which the tag was defined

getMultiplicity

java.lang.String **getMultiplicity**()

> Returns the multiplicity that was specified for the tag.
>
> **Returns:**
> > the multiplicity that was specified for the tag

getTagMetaClass

java.lang.String **getTagMetaClass**()

> Returns the name of the metaclass to which the tag is applicable. Relevant only for tags that belong to a profile.
>
> **Returns:**
> > the name of the metaclass to which the tag is applicable

getValue

java.lang.String **getValue**()

> Returns the value of the tag.
>
> **Returns:**
> > the value of the tag

getBase

setMultiplicity

void **setMultiplicity**(java.lang.String multiplicity)

Specifies the multiplicity for the tag.

Parameters:

multiplicity - the multiplicity to use for the tag. You can use strings such as "1" or "14" to specify a specific number, or you can use one of the values listed in the Features dialog for tags: "0,1", "*", or "1..*".

setTagContextValue

void **setTagContextValue**([IRPCollection](#) elements,
                           [IRPCollection](#) multiplicities)

Sets the value of the tag to a specific instance of another model element.

Parameters:

elements - collection of model elements representing the full path to the element. This collection is used to set the value of the tag to the full path of the target element. The collection must consist of objects of type IRPModelElement.

multiplicities - collection of the relevant indices for each of the model elements in the first collection (the "elements" parameter). This makes it possible to point to a specific instance of the target model element when multiplicity is greater than one. The collection must consist of integers provided as strings.

setTagMetaClass

void **setTagMetaClass**(java.lang.String tagMetaClass)

Specifies the metaclass to which the tag should be applicable, for example, "Class". Relevant only for tags that belong to a profile.

Parameters:

tagMetaClass - the metaclass to which the tag should be applicable

setValue

void **setValue**(java.lang.String value)

Sets the value of the tag.

Parameters:

value - the value to use for the tag

[Package](#)   **Class**   **[Use](#)**   **[Tree](#)**   **[Serialized](#)**   **[Deprecated](#)**   **[Index](#)**   **[Help](#)**

---

com.telelogic.rhapsody.core
Interface IRPTemplateInstantiation

All Superinterfaces:
> IRPModelElement

---

public interface **IRPTemplateInstantiation**
extends IRPModelElement

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPCollection	**getTemplateInstantiationParameters**() get property templateInstantiationParameters

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines,

com.telelogic.rhapsody.core

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks,
setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID,
setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner,
setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue,
setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation,
unlockOnDesignManager

Method Detail

getTemplateInstantiationParameters

IRPCollection **getTemplateInstantiationParameters**()

get property templateInstantiationParameters
Throws:
RhapsodyRuntimeException

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**
PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | **METHOD**   DETAIL: FIELD | CONSTR | **METHOD**

com.telelogic.rhapsody.core
Interface IRPTemplateInstantiationParameter

All Superinterfaces:
> [IRPModelElement](#)

---

public interface **IRPTemplateInstantiationParameter**
extends [IRPModelElement](#)

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.[IRPModelElement](#)
[IRPModelElement.OSLCLink](#)

Method Summary

java.lang.String	**[getArgValue](#)**() get property declaration
[IRPClassifier](#)	**[getType](#)**() get property type
void	**[setArgValue](#)**(java.lang.String argValue) set property declaration
void	**[setType](#)**([IRPClassifier](#) type) set property type

Methods inherited from interface com.telelogic.rhapsody.core.[IRPModelElement](#)
[addAssociation](#), [addDependency](#), [addDependencyBetween](#), [addDependencyTo](#), [addLinkToElement](#), [addNewAggr](#), [addProperty](#), [addRedefines](#), [addRemoteDependencyTo](#), [addSpecificStereotype](#), [addStereotype](#), [becomeTemplateInstantiationOf](#), [changeTo](#), [clone](#), [createOSLCLink](#), [deleteDependency](#), [deleteFromProject](#), [deleteOSLCLink](#), [errorMessage](#), [findElementsByFullName](#), [findNestedElement](#), [findNestedElementRecursive](#), [getAllTags](#), [getAnnotations](#), [getAssociationClasses](#), [getBinaryID](#), [getConstraints](#), [getConstraintsByHim](#), [getControlledFiles](#), [getDecorationStyle](#), [getDependencies](#), [getDescription](#), [getDescriptionHTML](#), [getDescriptionPlainText](#), [getDescriptionRTF](#), [getDisplayName](#), [getDisplayNameRTF](#), [getErrorMessage](#), [getFullPathName](#), [getFullPathNameIn](#), [getGUID](#), [getHyperLinks](#), [getIconFileName](#), [getInterfaceName](#), [getIsExternal](#), [getIsOfMetaClass](#), [getIsShowDisplayName](#), [getIsUnresolved](#), [getLocalTags](#), [getMainDiagram](#), [getMetaClass](#), [getName](#), [getNestedElements](#), [getNestedElementsByMetaClass](#), [getNestedElementsRecursive](#), [getNewTermStereotype](#), [getOfTemplate](#), [getOSLCLinks](#),

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getArgValue

java.lang.String **getArgValue**()

get property declaration
Throws:
RhapsodyRuntimeException

getType

IRPClassifier **getType**()

get property type
Throws:
RhapsodyRuntimeException

setArgValue

void **setArgValue**(java.lang.String argValue)

set property declaration
Throws:
RhapsodyRuntimeException

setType

```
void setType(IRPClassifier type)
```

> set property type
> **Throws:**
> > RhapsodyRuntimeException

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

com.telelogic.rhapsody.core
Interface IRPTemplateParameter

All Superinterfaces:
> IRPModelElement, IRPUnit, IRPVariable

---

public interface **IRPTemplateParameter**
extends IRPVariable

The IRPTemplateParameter interface represents parameters of a template in Rational Rhapsody models.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

java.lang.String	**getParameterKind**() Returns the type of the template parameter.
IRPModelElement	**getRepresentative**() For internal use only.
void	**setClassType**() Sets the type of the parameter to "class".
void	**setParameterKind**(java.lang.String parameterKind) Used to specify the type of the template parameter.
void	**setRepresentative**(IRPModelElement representative) For internal use only.

Methods inherited from interface com.telelogic.rhapsody.core.IRPVariable
addElementDefaultValue, addStringDefaultValue, getDeclaration, getDefaultValue, getType, getValueSpecifications, setDeclaration, setDefaultValue, setType, setTypeDeclaration

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

---

Methods inherited from interface com.telelogic.rhapsody.core.**IRPUnit**
copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getParameterKind

```
java.lang.String getParameterKind()
```

Returns the type of the template parameter.
Returns:
the type of the template parameter

---

getRepresentative

IRPModelElement **getRepresentative**()

>   For internal use only.

---

setClassType

void **setClassType**()

>   Sets the type of the parameter to "class".

---

setParameterKind

void **setParameterKind**(java.lang.String parameterKind)

>   Used to specify the type of the template parameter.
>   **Parameters:**
>   parameterKind - the type to use for the template parameter

---

setRepresentative

void **setRepresentative**(IRPModelElement representative)

>   For internal use only.

[Package](#) **Class** [Use](#) [Tree](#) [Serialized](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPTimingDiagram

All Superinterfaces:
> IRPDiagram, IRPModelElement, IRPSequenceDiagram, IRPUnit

```
public interface IRPTimingDiagram
```
extends IRPSequenceDiagram

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

int	**getIsElaborated**() Checks whether the the timing diagram is an elaborated timing diagram.
void	**setIsElaborated**(int isElaborated) Specifies whether the diagram should be an elaborated timing diagram or a compact timing diagram.

Methods inherited from interface com.telelogic.rhapsody.core.IRPSequenceDiagram
getLogicalCollaboration, getRelatedUseCases

Methods inherited from interface com.telelogic.rhapsody.core.IRPDiagram
addFreeShapeByType, addImage, addNewEdgeByType, addNewEdgeForElement, addNewNodeByType, addNewNodeForElement, addTextBox, closeDiagram, completeRelations, getCorrespondingGraphicElements, getElementsInDiagram, getGraphicalElements, getLastVisualizationModifiedTime, getPicture, getPictureAs, getPictureAsDividedMetafiles, getPictureEx, getPicturesWithImageMap, isOpen, isShowDiagramFrame, openDiagram, openDiagramView, populateDiagram, removeGraphElements, setShowDiagramFrame

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit
copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime,

---

Methods inherited from interface com.telelogic.rhapsody.core.**IRPUnit**
getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getIsElaborated

```
int getIsElaborated()
```

Checks whether the the timing diagram is an elaborated timing diagram.

Returns:
> indication of whether the diagram is an elaborated timing diagram. 1 means that the diagram is an elaborated timing diagram, 0 means that the diagram is a compact timing diagram.

setIsElaborated

```
void setIsElaborated(int isElaborated)
```

> Specifies whether the diagram should be an elaborated timing diagram or a compact timing diagram.
> **Parameters:**
>> isElaborated - Use 1 to indicate that the diagram should be an elaborated timing diagram, 0 to indicate that the diagram should be a compact timing diagram. Note that the type of the timing diagram should not be changed after you have already added elements to the diagram.

com.telelogic.rhapsody.core
Interface IRPTransition

All Superinterfaces:
>   IRPModelElement

---

public interface **IRPTransition**
extends IRPModelElement

The IRPTransition interface represents transitions in a statechart.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPTransition	**getInheritsFrom**() method getInheritsFrom
int	**getIsOverridden**() get isOverridden
IRPAction	**getItsAction**() method getItsAction
IRPGuard	**getItsGuard**() method getItsGuard
java.lang.String	**getItsLabel**() get property itsLabel
IRPStateVertex	**getItsSource**() get property itsSource
IRPStatechart	**getItsStatechart**() get property itsStatechart
IRPStateVertex	**getItsTarget**() get property itsTarget
IRPTrigger	

Method Summary

	getItsTrigger() method getItsTrigger
[IRPState](#)	**getOfState**() method getOfState
int	**isDefaultTransition**() method isDefaultTransition
int	**isStaticReaction**() method isStaticReaction
[IRPCollection](#)	**itsCompoundSource**() method itsCompoundSource
void	**overrideInheritance**() method overrideInheritance
[IRPTransition](#)	**resetLabelInheritance**() method resetLabelInheritance
[IRPAction](#)	**setItsAction**(java.lang.String action) method setItsAction
[IRPGuard](#)	**setItsGuard**(java.lang.String guard) method setItsGuard
void	**setItsLabel**(java.lang.String trigger, java.lang.String guard, java.lang.String action) property setItsLabel
void	**setItsSource**([IRPStateVertex](#) itsSource) set property itsSource
void	**setItsStatechart**([IRPStatechart](#) itsStatechart) set property itsStatechart
void	**setItsTarget**([IRPStateVertex](#) itsTarget) set property itsTarget
[IRPTrigger](#)	**setItsTrigger**(java.lang.String trigger) method setItsTrigger
void	**unoverrideInheritance**() method unoverrideInheritance

Methods inherited from interface com.telelogic.rhapsody.core.[IRPModelElement](#)

[addAssociation](#), [addDependency](#), [addDependencyBetween](#), [addDependencyTo](#), [addLinkToElement](#), [addNewAggr](#), [addProperty](#), [addRedefines](#), [addRemoteDependencyTo](#), [addSpecificStereotype](#), [addStereotype](#), [becomeTemplateInstantiationOf](#), [changeTo](#), [clone](#), [createOSLCLink](#), [deleteDependency](#), [deleteFromProject](#), [deleteOSLCLink](#), [errorMessage](#), [findElementsByFullName](#), [findNestedElement](#), [findNestedElementRecursive](#), [getAllTags](#), [getAnnotations](#), [getAssociationClasses](#), [getBinaryID](#), [getConstraints](#), [getConstraintsByHim](#), [getControlledFiles](#), [getDecorationStyle](#), [getDependencies](#), [getDescription](#), [getDescriptionHTML](#), [getDescriptionPlainText](#), [getDescriptionRTF](#), [getDisplayName](#), [getDisplayNameRTF](#), [getErrorMessage](#), [getFullPathName](#), [getFullPathNameIn](#),

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getInheritsFrom

IRPTransition **getInheritsFrom**()

> method getInheritsFrom
> **Throws:**
> > RhapsodyRuntimeException

---

getIsOverridden

int **getIsOverridden**()

> get isOverridden
> **Throws:**
> > RhapsodyRuntimeException

---

getItsAction

IRPAction **getItsAction**()

> method getItsAction
> **Throws:**
> > RhapsodyRuntimeException

---

getItsGuard

[IRPGuard](#) **getItsGuard**()

> method getItsGuard
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

getItsLabel

java.lang.String **getItsLabel**()

> get property itsLabel
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

getItsSource

[IRPStateVertex](#) **getItsSource**()

> get property itsSource
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

getItsStatechart

[IRPStatechart](#) **getItsStatechart**()

> get property itsStatechart
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

getItsTarget

[IRPStateVertex](#) **getItsTarget**()

> get property itsTarget
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

getItsTrigger

[IRPTrigger](#) **getItsTrigger**()

> method getItsTrigger
> **Throws:**

getItsGuard

RhapsodyRuntimeException

getOfState

IRPState **getOfState**()

method getOfState
Throws:
RhapsodyRuntimeException

isDefaultTransition

int **isDefaultTransition**()

method isDefaultTransition
Throws:
RhapsodyRuntimeException

isStaticReaction

int **isStaticReaction**()

method isStaticReaction
Throws:
RhapsodyRuntimeException

itsCompoundSource

IRPCollection **itsCompoundSource**()

method itsCompoundSource
Throws:
RhapsodyRuntimeException

overrideInheritance

void **overrideInheritance**()

method overrideInheritance
Throws:
RhapsodyRuntimeException

resetLabelInheritance

IRPTransition **resetLabelInheritance**()

      method resetLabelInheritance
      Throws:
            RhapsodyRuntimeException

---

setItsAction

IRPAction **setItsAction**(java.lang.String action)

      method setItsAction
      Throws:
            RhapsodyRuntimeException

---

setItsGuard

IRPGuard **setItsGuard**(java.lang.String guard)

      method setItsGuard
      Throws:
            RhapsodyRuntimeException

---

setItsLabel

void **setItsLabel**(java.lang.String trigger,
               java.lang.String guard,
               java.lang.String action)

      property setItsLabel
      Throws:
            RhapsodyRuntimeException

---

setItsSource

void **setItsSource**(IRPStateVertex itsSource)

      set property itsSource
      Throws:
            RhapsodyRuntimeException

---

setItsStatechart

void **setItsStatechart**(IRPStatechart itsStatechart)

resetLabelInheritance   698

set property itsStatechart
Throws:
        [RhapsodyRuntimeException](#)

---

setItsTarget

void **setItsTarget**([IRPStateVertex](#) itsTarget)

set property itsTarget
Throws:
        [RhapsodyRuntimeException](#)

---

setItsTrigger

[IRPTrigger](#) **setItsTrigger**(java.lang.String trigger)

method setItsTrigger
Throws:
        [RhapsodyRuntimeException](#)

---

unoverrideInheritance

void **unoverrideInheritance**()

method unoverrideInheritance
Throws:
        [RhapsodyRuntimeException](#)

---

com.telelogic.rhapsody.core
Interface IRPTrigger

All Superinterfaces:
> IRPModelElement

---

public interface **IRPTrigger**
extends IRPModelElement

The IRPTrigger interface represents the trigger of a transition in a statechart.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

java.lang.String	**getBody**() get property body
IRPInterfaceItem	**getItsOperation**() method getItsOperation
int	**isOperation**() method isOperation
int	**isTimeout**() method isTimeout
void	**setBody**(java.lang.String body) set property body

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies,

unoverrideInheritance   700

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

getBody

java.lang.String **getBody**()

> get property body
> **Throws:**
> > RhapsodyRuntimeException

getItsOperation

IRPInterfaceItem **getItsOperation**()

> method getItsOperation
> **Throws:**
> > RhapsodyRuntimeException

isOperation

int **isOperation**()

> method isOperation
> **Throws:**
> > RhapsodyRuntimeException

isTimeout

int **isTimeout**()

> method isTimeout
> **Throws:**
>> RhapsodyRuntimeException

setBody

void **setBody**(java.lang.String body)

> set property body
> **Throws:**
>> RhapsodyRuntimeException

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

PREV CLASS **NEXT CLASS**   **FRAMES**   **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

isOperation   702

 com.telelogic.rhapsody.core
Interface IRPType

All Superinterfaces:
> IRPClassifier, IRPModelElement, IRPUnit

```
public interface IRPType
extends IRPClassifier
```

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPEnumerationLiteral	**addEnumerationLiteral**(java.lang.String name) method addEnumerationLiteral
void	**deleteEnumerationLiteral**(IRPEnumerationLiteral literal) method deleteEnumerationLiteral
java.lang.String	**getDeclaration**() get property declaration
IRPCollection	**getEnumerationLiterals**() get property enumerationLiterals
int	**getIsPredefined**() get property isPredefined
int	**getIsTypedef**() get property isTypedef
int	**getIsTypedefConstant**() get property isTypedefConstant
int	**getIsTypedefOrdered**() get property isTypedefOrdered
int	**getIsTypedefReference**() get property isTypedefReference

setBody   703

Method Summary

java.lang.String	**getKind**() get property kind
[IRPClassifier](#)	**getTypedefBaseType**() get property typedefBaseType
java.lang.String	**getTypedefMultiplicity**() get property typedefMultiplicity
int	**isArray**() method isArray
int	**isEnum**() For types whose "kind" was set to Language, parses the declaration to see if the type is actually an enum.
int	**isEqualTo**() method isEqualTo
int	**isImplicit**() method isImplicit
int	**isKindEnumeration**() Checks whether the "kind" of the type is Enumeration.
int	**isKindLanguage**() Checks whether the "kind" of the type was set to Language.
int	**isKindStruct**() Checks whether the "kind" of the type is Structure.
int	**isKindTypedef**() Checks whether the "kind" of the type is Typedef.
int	**isKindUnion**() Checks whether the "kind" of the type is Union.
int	**isPointer**() method isPointer
int	**isPointerToPointer**() method isPointerToPointer
int	**isReference**() method isReference
int	**isReferenceToPointer**() method isReferenceToPointer
int	**isStruct**() For types whose "kind" was set to Language, parses the declaration to see if the type is actually a struct.
int	**isTemplate**() method isTemplate
int	

Method Summary

	isUnion() For types whose "kind" was set to Language, parses the declaration to see if the type is actually a union.
void	**setDeclaration**(java.lang.String declaration) set property declaration
void	**setIsTypedefConstant**(int isTypedefConstant) set property isTypedefConstant
void	**setIsTypedefOrdered**(int isTypedefOrdered) set property isTypedefOrdered
void	**setIsTypedefReference**(int isTypedefReference) set property isTypedefReference
void	**setKind**(java.lang.String kind) set property kind
void	**setTypedefBaseType**(IRPClassifier typedefBaseType) set property typedefBaseType
void	**setTypedefMultiplicity**(java.lang.String typedefMultiplicity) set property typedefMultiplicity

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPClassifier

addActivityDiagram, addAttribute, addFlowItems, addFlows, addGeneralization, addOperation, addRelation, addRelationTo, addStatechart, addUnidirectionalRelation, addUnidirectionalRelationTo, deleteAttribute, deleteFlowItems, deleteFlows, deleteGeneralization, deleteOperation, deleteRelation, findAttribute, findBaseClassifier, findDerivedClassifier, findGeneralization, findInterfaceItem, findNestedClassifier, findNestedClassifierRecursive, findRelation, findTrigger, getActivityDiagram, getAttributes, getAttributesIncludingBases, getBaseClassifiers, getBehavioralDiagrams, getDerivedClassifiers, getFlowItems, getFlows, getGeneralizations, getInterfaceItems, getInterfaceItemsIncludingBases, getLinks, getNestedClassifiers, getOperations, getPorts, getRelations, getRelationsIncludingBases, getSequenceDiagrams, getSourceArtifacts, getStatechart

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags,

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addEnumerationLiteral

IRPEnumerationLiteral **addEnumerationLiteral**(java.lang.String name)

> method addEnumerationLiteral
> **Throws:**
> > RhapsodyRuntimeException

---

deleteEnumerationLiteral

void **deleteEnumerationLiteral**(IRPEnumerationLiteral literal)

> method deleteEnumerationLiteral
> **Throws:**
> > RhapsodyRuntimeException

---

getDeclaration

java.lang.String **getDeclaration**()

> get property declaration

Throws:
RhapsodyRuntimeException

getEnumerationLiterals

IRPCollection **getEnumerationLiterals**()

get property enumerationLiterals
Throws:
RhapsodyRuntimeException

getIsPredefined

int **getIsPredefined**()

get property isPredefined
Throws:
RhapsodyRuntimeException

getIsTypedef

int **getIsTypedef**()

get property isTypedef
Throws:
RhapsodyRuntimeException

getIsTypedefConstant

int **getIsTypedefConstant**()

get property isTypedefConstant
Throws:
RhapsodyRuntimeException

getIsTypedefOrdered

int **getIsTypedefOrdered**()

get property isTypedefOrdered
Throws:
RhapsodyRuntimeException

getIsTypedefReference

int **getIsTypedefReference**()

> get property isTypedefReference
> **Throws:**
>> RhapsodyRuntimeException

---

getKind

java.lang.String **getKind**()

> get property kind
> **Throws:**
>> RhapsodyRuntimeException

---

getTypedefBaseType

IRPClassifier **getTypedefBaseType**()

> get property typedefBaseType
> **Throws:**
>> RhapsodyRuntimeException

---

getTypedefMultiplicity

java.lang.String **getTypedefMultiplicity**()

> get property typedefMultiplicity
> **Throws:**
>> RhapsodyRuntimeException

---

isArray

int **isArray**()

> method isArray
> **Throws:**
>> RhapsodyRuntimeException

---

isEnum

int **isEnum**()

> For types whose "kind" was set to Language, parses the declaration to see if the type is actually an enum.

Returns:

1 if the type is an enum, 0 otherwise

Throws:

RhapsodyRuntimeException

isEqualTo

int **isEqualTo**()

method isEqualTo

Throws:

RhapsodyRuntimeException

isImplicit

int **isImplicit**()

method isImplicit

Throws:

RhapsodyRuntimeException

isKindEnumeration

int **isKindEnumeration**()

Checks whether the "kind" of the type is Enumeration.

Returns:

1 if the "kind" of the type is Enumeration, 0 otherwise

Throws:

RhapsodyRuntimeException

isKindLanguage

int **isKindLanguage**()

Checks whether the "kind" of the type was set to Language.

Returns:

1 if the "kind" of the type is Language, 0 otherwise

Throws:

RhapsodyRuntimeException

isKindStruct

int **isKindStruct**()

Checks whether the "kind" of the type is Structure.

Returns:

1 if the "kind" of the type is Structure, 0 otherwise

Throws:

isKindTypedef

int **isKindTypedef**()

Checks whether the "kind" of the type is Typedef.

Returns:

1 if the "kind" of the type is Typedef, 0 otherwise

Throws:

isKindUnion

int **isKindUnion**()

Checks whether the "kind" of the type is Union.

Returns:

1 if the "kind" of the type is Union, 0 otherwise

Throws:

isPointer

int **isPointer**()

method isPointer

Throws:

isPointerToPointer

int **isPointerToPointer**()

method isPointerToPointer

Throws:

isReference

int **isReference**()

method isReference

Throws:
RhapsodyRuntimeException

isReferenceToPointer

int **isReferenceToPointer**()

method isReferenceToPointer
Throws:
RhapsodyRuntimeException

isStruct

int **isStruct**()

For types whose "kind" was set to Language, parses the declaration to see if the type is actually a struct.
Returns:
1 if the type is a struct, 0 otherwise
Throws:
RhapsodyRuntimeException

isTemplate

int **isTemplate**()

method isTemplate
Throws:
RhapsodyRuntimeException

isUnion

int **isUnion**()

For types whose "kind" was set to Language, parses the declaration to see if the type is actually a union.
Returns:
1 if the type is a union, 0 otherwise
Throws:
RhapsodyRuntimeException

setDeclaration

void **setDeclaration**(java.lang.String declaration)

set property declaration

isReference 711

Throws:
RhapsodyRuntimeException

setIsTypedefConstant

void **setIsTypedefConstant**(int isTypedefConstant)

set property isTypedefConstant
Throws:
RhapsodyRuntimeException

setIsTypedefOrdered

void **setIsTypedefOrdered**(int isTypedefOrdered)

set property isTypedefOrdered
Throws:
RhapsodyRuntimeException

setIsTypedefReference

void **setIsTypedefReference**(int isTypedefReference)

set property isTypedefReference
Throws:
RhapsodyRuntimeException

setKind

void **setKind**(java.lang.String kind)

set property kind
Throws:
RhapsodyRuntimeException

setTypedefBaseType

void **setTypedefBaseType**(IRPClassifier typedefBaseType)

set property typedefBaseType
Throws:
RhapsodyRuntimeException

setTypedefMultiplicity

void **setTypedefMultiplicity**(java.lang.String typedefMultiplicity)

set property typedefMultiplicity
Throws:
RhapsodyRuntimeException

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

---

com.telelogic.rhapsody.core
Interface IRPUnit

All Superinterfaces:
> IRPModelElement

All Known Subinterfaces:
> IRPActivityDiagram, IRPActor, IRPAnnotation, IRPArgument, IRPAssociationClass, IRPAttribute, IRPClass, IRPClassifier, IRPCollaborationDiagram, IRPComment, IRPComponent, IRPComponentDiagram, IRPConstraint, IRPControlledFile, IRPDeploymentDiagram, IRPDiagram, IRPEvent, IRPEventReception, IRPFile, IRPFlowchart, IRPFlowItem, IRPInstance, IRPInterfaceItem, IRPLink, IRPMatrixLayout, IRPMatrixView, IRPModule, IRPNode, IRPObjectModelDiagram, IRPOperation, IRPPackage, IRPPanelDiagram, IRPPort, IRPProfile, IRPProject, IRPRelation, IRPRequirement, IRPSequenceDiagram, IRPStatechart, IRPStatechartDiagram, IRPStereotype, IRPStructureDiagram, IRPSysMLPort, IRPTableLayout, IRPTableView, IRPTag, IRPTemplateParameter, IRPTimingDiagram, IRPType, IRPUseCase, IRPUseCaseDiagram, IRPVariable

---

public interface **IRPUnit**
extends IRPModelElement

The IRPUnit interface represents model elements that can be saved as separate files.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPUnit	**copyToAnotherProject**(IRPModelElement parentInTarget) Makes an editable copy of the unit in a different project.
int	**getAddToModelMode**() Returns an indication of how the unit was added to the model.
java.lang.String	**getCMHeader**() Returns the header used by the Configuration Management tool for the unit.
int	**getCMState**()

Method Summary

	Returns the configuration management state of the unit.
java.lang.String	**getCurrentDirectory**() Gets the name of the directory that contains the file used to store the unit.
java.lang.String	**getFilename**() Gets the name of the file used to store the unit.
int	**getIncludeInNextLoad**() Checks whether the unit is going to be loaded the next time the model is loaded.
int	**getIsStub**() Checks whether the unit is currently unloaded.
java.lang.String	**getLanguage**() Gets the language of the unit.
java.lang.String	**getLastModifiedTime**() Returns the time at which the file representing the unit was last modified.
[IRPCollection](#)	**getNestedSaveUnits**() Returns a collection of any sub-elements of the unit that were saved as individual files.
int	**getNestedSaveUnitsCount**() Returns the number of sub-elements of the unit that were saved as individual files.
[IRPCollection](#)	**getStructureDiagrams**() Returns a collection of any structure diagrams that are sub-elements of the unit.
int	**isReadOnly**() Checks whether the file used to store the unit is read-only.
int	**isReferenceUnit**() Checks whether the unit was added to the model as a reference.
int	**isSeparateSaveUnit**() Checks whether the current IRPUnit object is saved in its own file.
[IRPUnit](#)	**load**(int withSubs) Loads the unit.
[IRPUnit](#)	**moveToAnotherProjectLeaveAReference**([IRPModelElement](#) parentInTarget) Moves the unit to a different project, and adds a reference to it in the original project.
[IRPUnit](#)	**referenceToAnotherProject**([IRPModelElement](#) parentInTarget) Creates a reference to the unit in a different project.
void	**save**(int withSubs) Saves the unit.
void	**setCMHeader**(java.lang.String cMHeader) Sets the Configuration Management tool header for the unit.
void	**setFilename**(java.lang.String filename) Specifies the name that should be used for the file representing the unit.

Method Summary

void	**setIncludeInNextLoad**(int includeInNextLoad) Toggles whether the unit is going to be loaded the next time the model is loaded.
void	**setLanguage**(java.lang.String newLanguage, int recursive) Specifies the programming language that should be used when code is generated for the unit.
void	**setReadOnly**(int pVal) Toggles the read-only status of the file used to store the unit.
void	**setSeparateSaveUnit**(int pVal) Specifies whether the current IRPUnit object should be saved in its own file.
void	**setUnitPath**(java.lang.String newPath) Specifies the path that should be used to locate the unit when it is added to a model "By Reference".
void	**unload**() Unloads the unit.

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

copyToAnotherProject

[IRPUnit](#) **copyToAnotherProject**([IRPModelElement](#) parentInTarget)

Makes an editable copy of the unit in a different project.
Parameters:
    parentInTarget - the model element that will be the parent of the new unit in the target
    project
Returns:
    the editable unit that was created in the target project

---

getAddToModelMode

int **getAddToModelMode**()

Returns an indication of how the unit was added to the model. See
[IRPApplication.AddToModel_Mode](#) for the possible values.
Returns:
    indication of how the unit was added to the model

---

getCMHeader

java.lang.String **getCMHeader**()

Returns the header used by the Configuration Management tool for the unit.
Returns:
    the header used by the Configuration Management tool for the unit

---

getCMState

int **getCMState**()

Returns the configuration management state of the unit.
Returns:
    the configuration management state of the unit

---

getCurrentDirectory

java.lang.String **getCurrentDirectory**()

Gets the name of the directory that contains the file used to store the unit. The string returned consists
of the full path except for the name of the file itself.
Returns:
    the name of the directory that contains the file used to store the unit

---

getFilename

`java.lang.String` **`getFilename`**`()`

Gets the name of the file used to store the unit. The string returned consists only of the filename, not the entire path.

Returns:
the name of the file used to store the unit

getIncludeInNextLoad

`int` **`getIncludeInNextLoad`**`()`

Checks whether the unit is going to be loaded the next time the model is loaded.

Returns:
1 if the unit is going to be loaded the next time the model is loaded, 0 if the unit is not going to be loaded

getIsStub

`int` **`getIsStub`**`()`

Checks whether the unit is currently unloaded.

Returns:
1 if the unit is not currently loaded, 0 if it is currently loaded

getLanguage

`java.lang.String` **`getLanguage`**`()`

Gets the language of the unit.

Returns:
the language of the unit

getLastModifiedTime

`java.lang.String` **`getLastModifiedTime`**`()`

Returns the time at which the file representing the unit was last modified.

Returns:
the time at which the file representing the unit was last modified

getNestedSaveUnits

[IRPCollection](#) **getNestedSaveUnits**()

>Returns a collection of any sub-elements of the unit that were saved as individual files.
>**Returns:**
>>any sub-elements of the unit that were saved as individual files

getNestedSaveUnitsCount

int **getNestedSaveUnitsCount**()

>Returns the number of sub-elements of the unit that were saved as individual files.
>**Returns:**
>>the number of sub-elements of the unit that were saved as individual files

getStructureDiagrams

[IRPCollection](#) **getStructureDiagrams**()

>Returns a collection of any structure diagrams that are sub-elements of the unit. Used primarily for structure diagrams that belong to individual classes.
>**Returns:**
>>any structure diagrams that are sub-elements of the unit

isReadOnly

int **isReadOnly**()

>Checks whether the file used to store the unit is read-only.
>**Returns:**
>>1 if the file is read-only, 0 if the file is not read-only

isReferenceUnit

int **isReferenceUnit**()

>Checks whether the unit was added to the model as a reference.
>**Returns:**
>>1 if the unit was added to the model as a reference, 0 if it was not added as a reference

isSeparateSaveUnit

int **isSeparateSaveUnit**()

Checks whether the current IRPUnit object is saved in its own file. (Keep in mind that IRPUnit objects represent any element that can in theory be saved as a separate file, even if this is not the case for a specific element in your model.)

Returns:

1 if the unit is saved in its own file, 0 otherwise

load

IRPUnit **load**(int withSubs)

Loads the unit.

Parameters:

withSubs - indication of whether the unit's subunits should be loaded as well (1 to load the subunits as well, 0 to load only the unit itself)

Returns:

the unit that was loaded

moveToAnotherProjectLeaveAReference

IRPUnit **moveToAnotherProjectLeaveAReference**(IRPModelElement parentInTarget)

Moves the unit to a different project, and adds a reference to it in the original project.

Parameters:

parentInTarget - the model element that will be the parent of the new unit in the target project

Returns:

the editable unit that was created in the target project

referenceToAnotherProject

IRPUnit **referenceToAnotherProject**(IRPModelElement parentInTarget)

Creates a reference to the unit in a different project.

Parameters:

parentInTarget - the model element that will be the parent of the reference (read-only) unit created in the target project

Returns:

the reference (read-only) unit that was created in the target project

save

void **save**(int withSubs)

Saves the unit.

Parameters:

withSubs - indication of whether the unit's subunits should be saved as well (1 to save the subunits as well, 0 to save only the unit itself)

setCMHeader

void **setCMHeader**(java.lang.String cMHeader)

Sets the Configuration Management tool header for the unit.
Parameters:
cMHeader - the Configuration Management tool header to use for the unit

setFilename

void **setFilename**(java.lang.String filename)

Specifies the name that should be used for the file representing the unit. The string should only include the first part of the filename, Rational Rhapsody handles the file extension. (Note that if you change the filename, the old file remains on disk.)
Parameters:
filename - the name that should be used for the file representing the unit

setIncludeInNextLoad

void **setIncludeInNextLoad**(int includeInNextLoad)

Toggles whether the unit is going to be loaded the next time the model is loaded.
Parameters:
includeInNextLoad - Use 1 to specify that the unit should be loaded the next time the model is loaded, 0 to specify that the unit should not be loaded

setLanguage

void **setLanguage**(java.lang.String newLanguage,
                int recursive)

Specifies the programming language that should be used when code is generated for the unit. This method can be used for mixed-language models.
Parameters:
newLanguage - use one of the following strings: C++ or cpp, C, Java, Ada, C#
recursive - use 1 if you want to set the language for all subunits of the element, otherwise use 0

```
jeepUnit.setLanguage("cpp", 0);
```

setReadOnly

void **setReadOnly**(int pVal)

> Toggles the read-only status of the file used to store the unit.
> **Parameters:**
>> pVal - Use 1 to change the file to read-only, 0 to change the file to read/write

setSeparateSaveUnit

void **setSeparateSaveUnit**(int pVal)

> Specifies whether the current IRPUnit object should be saved in its own file. (Keep in mind that IRPUnit objects represent any element that can in theory be saved as a separate file, even if this is not the case for a specific element in your model.)
> **Parameters:**
>> pVal - Use 1 to specify that the element should be saved in its own file. Use 0 to specify that the element should not be saved in its own file

setUnitPath

void **setUnitPath**(java.lang.String newPath)

> Specifies the path that should be used to locate the unit when it is added to a model "By Reference".
> **Parameters:**
>> newPath - the path that should be used to locate the unit when it is added to a model "By Reference"

unload

void **unload**()

> Unloads the unit.

| **Package** | **Class** | **Use** | **Tree** | **Serialized** | **Deprecated** | **Index** | **Help** |

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | **METHOD**   DETAIL: FIELD | CONSTR | **METHOD**

com.telelogic.rhapsody.core
Interface IRPUseCase

All Superinterfaces:
> IRPClassifier, IRPModelElement, IRPUnit

```
public interface IRPUseCase
extends IRPClassifier
```

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

void	**addDescribingDiagram**(IRPDiagram diagram) method addDescribingDiagram
void	**addExtensionPoint**(java.lang.String entryPoint) method addExtensionPoint
void	**deleteDescribingDiagram**(IRPDiagram diagram) method deleteDescribingDiagram
void	**deleteEntryPoint**(java.lang.String entryPoint) method deleteEntryPoint
void	**deleteExtensionPoint**(java.lang.String entryPoint) method deleteExtensionPoint
java.lang.String	**findEntryPoint**(IRPGeneralization gen) method findEntryPoint
java.lang.String	**findExtensionPoint**(IRPGeneralization gen) method findExtensionPoint
IRPDiagram	**getDescribingDiagram**(java.lang.String name) method getDescribingDiagram
IRPCollection	**getDescribingDiagrams**() get property describingDiagrams

unload   723

Method Summary

[IRPCollection](#)	**[getEntryPoints](#)**() get property entryPoints
[IRPCollection](#)	**[getExtensionPoints](#)**() get property extensionPoints
int	**[getIsBehaviorOverriden](#)**() get property isBehaviorOverriden
void	**[setIsBehaviorOverriden](#)**(int isBehaviorOverriden) set property isBehaviorOverriden

---

Methods inherited from interface com.telelogic.rhapsody.core.**[IRPClassifier](#)**

[addActivityDiagram](#), [addAttribute](#), [addFlowItems](#), [addFlows](#), [addGeneralization](#), [addOperation](#), [addRelation](#), [addRelationTo](#), [addStatechart](#), [addUnidirectionalRelation](#), [addUnidirectionalRelationTo](#), [deleteAttribute](#), [deleteFlowItems](#), [deleteFlows](#), [deleteGeneralization](#), [deleteOperation](#), [deleteRelation](#), [findAttribute](#), [findBaseClassifier](#), [findDerivedClassifier](#), [findGeneralization](#), [findInterfaceItem](#), [findNestedClassifier](#), [findNestedClassifierRecursive](#), [findRelation](#), [findTrigger](#), [getActivityDiagram](#), [getAttributes](#), [getAttributesIncludingBases](#), [getBaseClassifiers](#), [getBehavioralDiagrams](#), [getDerivedClassifiers](#), [getFlowItems](#), [getFlows](#), [getGeneralizations](#), [getInterfaceItems](#), [getInterfaceItemsIncludingBases](#), [getLinks](#), [getNestedClassifiers](#), [getOperations](#), [getPorts](#), [getRelations](#), [getRelationsIncludingBases](#), [getSequenceDiagrams](#), [getSourceArtifacts](#), [getStatechart](#)

Methods inherited from interface com.telelogic.rhapsody.core.**[IRPUnit](#)**

[copyToAnotherProject](#), [getAddToModelMode](#), [getCMHeader](#), [getCMState](#), [getCurrentDirectory](#), [getFilename](#), [getIncludeInNextLoad](#), [getIsStub](#), [getLanguage](#), [getLastModifiedTime](#), [getNestedSaveUnits](#), [getNestedSaveUnitsCount](#), [getStructureDiagrams](#), [isReadOnly](#), [isReferenceUnit](#), [isSeparateSaveUnit](#), [load](#), [moveToAnotherProjectLeaveAReference](#), [referenceToAnotherProject](#), [save](#), [setCMHeader](#), [setFilename](#), [setIncludeInNextLoad](#), [setLanguage](#), [setReadOnly](#), [setSeparateSaveUnit](#), [setUnitPath](#), [unload](#)

Methods inherited from interface com.telelogic.rhapsody.core.**[IRPModelElement](#)**

[addAssociation](#), [addDependency](#), [addDependencyBetween](#), [addDependencyTo](#), [addLinkToElement](#), [addNewAggr](#), [addProperty](#), [addRedefines](#), [addRemoteDependencyTo](#), [addSpecificStereotype](#), [addStereotype](#), [becomeTemplateInstantiationOf](#), [changeTo](#), [clone](#), [createOSLCLink](#), [deleteDependency](#), [deleteFromProject](#), [deleteOSLCLink](#), [errorMessage](#), [findElementsByFullName](#), [findNestedElement](#), [findNestedElementRecursive](#), [getAllTags](#), [getAnnotations](#), [getAssociationClasses](#), [getBinaryID](#), [getConstraints](#), [getConstraintsByHim](#), [getControlledFiles](#), [getDecorationStyle](#), [getDependencies](#), [getDescription](#), [getDescriptionHTML](#), [getDescriptionPlainText](#), [getDescriptionRTF](#), [getDisplayName](#), [getDisplayNameRTF](#), [getErrorMessage](#), [getFullPathName](#), [getFullPathNameIn](#), [getGUID](#), [getHyperLinks](#), [getIconFileName](#), [getInterfaceName](#), [getIsExternal](#), [getIsOfMetaClass](#), [getIsShowDisplayName](#), [getIsUnresolved](#), [getLocalTags](#), [getMainDiagram](#), [getMetaClass](#), [getName](#), [getNestedElements](#), [getNestedElementsByMetaClass](#), [getNestedElementsRecursive](#), [getNewTermStereotype](#), [getOfTemplate](#), [getOSLCLinks](#), [getOverlayIconFileName](#), [getOverriddenProperties](#), [getOverriddenPropertiesByPattern](#), [getOwnedDependencies](#), [getOwner](#), [getProject](#), [getPropertyValue](#), [getPropertyValueConditional](#), [getPropertyValueConditionalExplicit](#), [getPropertyValueExplicit](#), [getRedefines](#), [getReferences](#), [getRemoteDependencies](#), [getRemoteURI](#), [getRequirementTraceabilityHandle](#), [getSaveUnit](#), [getStereotype](#), [getStereotypes](#), [getTag](#), [getTemplateParameters](#), [getTi](#), [getToolTipHTML](#),

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Method Detail

addDescribingDiagram

void **addDescribingDiagram**(IRPDiagram diagram)

method addDescribingDiagram
Throws:
RhapsodyRuntimeException

addExtensionPoint

void **addExtensionPoint**(java.lang.String entryPoint)

method addExtensionPoint
Throws:
RhapsodyRuntimeException

deleteDescribingDiagram

void **deleteDescribingDiagram**(IRPDiagram diagram)

method deleteDescribingDiagram
Throws:
RhapsodyRuntimeException

deleteEntryPoint

void **deleteEntryPoint**(java.lang.String entryPoint)

method deleteEntryPoint
Throws:
RhapsodyRuntimeException

deleteExtensionPoint

void **deleteExtensionPoint**(java.lang.String entryPoint)

method deleteExtensionPoint
Throws:
RhapsodyRuntimeException

---

findEntryPoint

java.lang.String **findEntryPoint**(IRPGeneralization gen)

method findEntryPoint
Throws:
RhapsodyRuntimeException

---

findExtensionPoint

java.lang.String **findExtensionPoint**(IRPGeneralization gen)

method findExtensionPoint
Throws:
RhapsodyRuntimeException

---

getDescribingDiagram

IRPDiagram **getDescribingDiagram**(java.lang.String name)

method getDescribingDiagram
Throws:
RhapsodyRuntimeException

---

getDescribingDiagrams

IRPCollection **getDescribingDiagrams**()

get property describingDiagrams
Throws:
RhapsodyRuntimeException

---

getEntryPoints

IRPCollection **getEntryPoints**()

get property entryPoints
Throws:

RhapsodyRuntimeException

---

getExtensionPoints

IRPCollection **getExtensionPoints**()

> get property extensionPoints
> **Throws:**
> > RhapsodyRuntimeException

---

getIsBehaviorOverriden

int **getIsBehaviorOverriden**()

> get property isBehaviorOverriden
> **Throws:**
> > RhapsodyRuntimeException

---

setIsBehaviorOverriden

void **setIsBehaviorOverriden**(int isBehaviorOverriden)

> set property isBehaviorOverriden
> **Throws:**
> > RhapsodyRuntimeException

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPUseCaseDiagram

All Superinterfaces:
> IRPDiagram, IRPModelElement, IRPUnit

```
public interface IRPUseCaseDiagram
```
extends IRPDiagram

The IRPUseCaseDiagram interface represents use case diagrams in a Rational Rhapsody model.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

Methods inherited from interface com.telelogic.rhapsody.core.IRPDiagram
addFreeShapeByType, addImage, addNewEdgeByType, addNewEdgeForElement, addNewNodeByType, addNewNodeForElement, addTextBox, closeDiagram, completeRelations, getCorrespondingGraphicElements, getElementsInDiagram, getGraphicalElements, getLastVisualizationModifiedTime, getPicture, getPictureAs, getPictureAsDividedMetafiles, getPictureEx, getPicturesWithImageMap, isOpen, isShowDiagramFrame, openDiagram, openDiagramView, populateDiagram, removeGraphElements, setShowDiagramFrame

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit
copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype,

setIsBehaviorOverriden   728

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

---

Package   **Class**   **Use**   **Tree**   **Serialized**   **Deprecated**   **Index**   **Help**

PREV CLASS   **NEXT CLASS**   **FRAMES**   **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Interface IRPValueSpecification

All Superinterfaces:
>   IRPModelElement

All Known Subinterfaces:
>   IRPContextSpecification, IRPInstanceValue, IRPLiteralSpecification

---

```
public interface IRPValueSpecification
```
extends IRPModelElement

The interface IRPValueSpecification represents the UML concept of "value specification" and serves as the base interface for IRPContextSpecification, IRPInstanceValue, and IRPLiteralSpecification.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement
addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies,

Methods inherited from interface com.telelogic.rhapsody.core.**IRPModelElement**
getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation, unlockOnDesignManager

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**
PREV CLASS	**NEXT CLASS**			**FRAMES**	**NO FRAMES**	**All Classes**	
SUMMARY: NESTED | FIELD | CONSTR | METHOD				DETAIL: FIELD | CONSTR | METHOD			

com.telelogic.rhapsody.core Interface IRPValueSpecification   731

com.telelogic.rhapsody.core
Interface IRPVariable

All Superinterfaces:
> IRPModelElement, IRPUnit

All Known Subinterfaces:
> IRPArgument, IRPAttribute, IRPTag, IRPTemplateParameter

---

public interface **IRPVariable**
extends IRPUnit

The IRPVariable interface represents the characteristics shared by model elements such as attributes, variables, and arguments.

Nested Class Summary

Nested classes/interfaces inherited from interface com.telelogic.rhapsody.core.IRPModelElement
IRPModelElement.OSLCLink

Method Summary

IRPInstanceValue	**addElementDefaultValue**(IRPModelElement newDefaultVal) For tags with multiplicity greater than 1, this method can be used to add a model element as an additional value.
IRPLiteralSpecification	**addStringDefaultValue**(java.lang.String newDefaultVal) For tags with multiplicity greater than 1, this method can be used to add a string as an additional value.
java.lang.String	**getDeclaration**() Returns the type declaration if an on-the-fly type was used for the element rather than an existing type.
java.lang.String	**getDefaultValue**() Returns the default value that was set for the variable.
IRPClassifier	**getType**() Returns the type of the variable.
IRPCollection	**getValueSpecifications**()

Method Summary

	Returns a collection of the initial values that were declared for elements where the multiplicity is greater than one.
void	**setDeclaration**(java.lang.String declaration) Specifies an "on-the-fly" declaration for the type of the element instead of using an existing type.
void	**setDefaultValue**(java.lang.String defaultValue) Sets a new default value for the variable.
void	**setType**(IRPClassifier type) Sets the type of the variable.
void	**setTypeDeclaration**(java.lang.String newVal) Specifies an "on-the-fly" declaration for the type of the element but first checks whether there is an existing type that matches the string provided as an argument.

Methods inherited from interface com.telelogic.rhapsody.core.IRPUnit

copyToAnotherProject, getAddToModelMode, getCMHeader, getCMState, getCurrentDirectory, getFilename, getIncludeInNextLoad, getIsStub, getLanguage, getLastModifiedTime, getNestedSaveUnits, getNestedSaveUnitsCount, getStructureDiagrams, isReadOnly, isReferenceUnit, isSeparateSaveUnit, load, moveToAnotherProjectLeaveAReference, referenceToAnotherProject, save, setCMHeader, setFilename, setIncludeInNextLoad, setLanguage, setReadOnly, setSeparateSaveUnit, setUnitPath, unload

---

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

addAssociation, addDependency, addDependencyBetween, addDependencyTo, addLinkToElement, addNewAggr, addProperty, addRedefines, addRemoteDependencyTo, addSpecificStereotype, addStereotype, becomeTemplateInstantiationOf, changeTo, clone, createOSLCLink, deleteDependency, deleteFromProject, deleteOSLCLink, errorMessage, findElementsByFullName, findNestedElement, findNestedElementRecursive, getAllTags, getAnnotations, getAssociationClasses, getBinaryID, getConstraints, getConstraintsByHim, getControlledFiles, getDecorationStyle, getDependencies, getDescription, getDescriptionHTML, getDescriptionPlainText, getDescriptionRTF, getDisplayName, getDisplayNameRTF, getErrorMessage, getFullPathName, getFullPathNameIn, getGUID, getHyperLinks, getIconFileName, getInterfaceName, getIsExternal, getIsOfMetaClass, getIsShowDisplayName, getIsUnresolved, getLocalTags, getMainDiagram, getMetaClass, getName, getNestedElements, getNestedElementsByMetaClass, getNestedElementsRecursive, getNewTermStereotype, getOfTemplate, getOSLCLinks, getOverlayIconFileName, getOverriddenProperties, getOverriddenPropertiesByPattern, getOwnedDependencies, getOwner, getProject, getPropertyValue, getPropertyValueConditional, getPropertyValueConditionalExplicit, getPropertyValueExplicit, getRedefines, getReferences, getRemoteDependencies, getRemoteURI, getRequirementTraceabilityHandle, getSaveUnit, getStereotype, getStereotypes, getTag, getTemplateParameters, getTi, getToolTipHTML, getUserDefinedMetaClass, hasNestedElements, hasPanelWidget, highLightElement, isATemplate, isDescriptionRTF, isDisplayNameRTF, isModified, isRemote, locateInBrowser, lockOnDesignManager, openFeaturesDialog, removeProperty, removeRedefines, removeStereotype, setDecorationStyle, setDescription, setDescriptionAndHyperlinks, setDescriptionHTML, setDescriptionRTF, setDisplayName, setDisplayNameRTF, setGUID, setIsShowDisplayName, setMainDiagram, setName, setOfTemplate, setOwner, setPropertyValue, setRequirementTraceabilityHandle, setStereotype, setTagContextValue, setTagElementValue, setTagValue, setTi, synchronizeTemplateInstantiation,

Methods inherited from interface com.telelogic.rhapsody.core.IRPModelElement

unlockOnDesignManager

Method Detail

addElementDefaultValue

IRPInstanceValue **addElementDefaultValue**(IRPModelElement newDefaultVal)

> For tags with multiplicity greater than 1, this method can be used to add a model element as an additional value.
> **Parameters:**
>> newDefaultVal - the model element to add as an additional value
> **Returns:**
>> the value that was added

addStringDefaultValue

IRPLiteralSpecification **addStringDefaultValue**(java.lang.String newDefaultVal)

> For tags with multiplicity greater than 1, this method can be used to add a string as an additional value.
> **Parameters:**
>> newDefaultVal - the string to add as an additional value
> **Returns:**
>> the value that was added

getDeclaration

java.lang.String **getDeclaration**()

> Returns the type declaration if an on-the-fly type was used for the element rather than an existing type.
> **Returns:**
>> the type declaration if an on-the-fly type was used for the element

getDefaultValue

java.lang.String **getDefaultValue**()

> Returns the default value that was set for the variable.
> **Returns:**
>> the default value of the variable

getType

[IRPClassifier](#) **getType**()

>   Returns the type of the variable.
>   **Returns:**
>   the type of the variable

getValueSpecifications

[IRPCollection](#) **getValueSpecifications**()

>   Returns a collection of the initial values that were declared for elements where the multiplicity is
>   greater than one. Note that the type of the objects contained in the returned collection depends upon
>   the the type of element on which this method was called (there are a number of element types that
>   inherit from IRPVariable). When the method is called on a tag, the objects in the returned collection
>   are of type IRPContextSpecification. These objects are created for a tag when you call the method
>   IRPModelElement.setTagContextValue.
>   **Returns:**
>   the initial values that were declared for elements where the multiplicity is greater than one
>   **See Also:**
>   [IRPModelElement.setTagContextValue(com.telelogic.rhapsody.core.IRPTag,](#)
>   [com.telelogic.rhapsody.core.IRPCollection,](#)
>   [com.telelogic.rhapsody.core.IRPCollection)](#)

setDeclaration

void **setDeclaration**(java.lang.String declaration)

>   Specifies an "on-the-fly" declaration for the type of the element instead of using an existing type.
>   Note that the string that you provide will be used as the declaration for the type even if if matches an
>   existing type. For example, if you call this method with the argument "int", it will create an on-the-fly
>   declaration consisting of "int". Use the method setTypeDeclaration if you want Rhapsody to first
>   check whether there is an existing type with that name.
>   **Parameters:**
>   declaration - the on-the-fly declaration to use for the type of the element

setDefaultValue

void **setDefaultValue**(java.lang.String defaultValue)

>   Sets a new default value for the variable.
>   **Parameters:**
>   defaultValue - the default value to use for the variable

setType

void **setType**([IRPClassifier](#) type)

>Sets the type of the variable.
>**Parameters:**
>>`type` - the type to use for the variable

setTypeDeclaration

void **setTypeDeclaration**(java.lang.String newVal)

>Specifies an "on-the-fly" declaration for the type of the element but first checks whether there is an existing type that matches the string provided as an argument. If there is such a type, it will be used as the type of the model element. Note that this method is slower than the method setDeclaration because it first carries out a search. So if you definitely want to use an on-the-fly declaration, use the method setDeclaration instead.
>**Parameters:**
>>`newVal` - the type to use for the type of the element

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Class RhapsodyAppServer

```
java.lang.Object
  └─ com.telelogic.rhapsody.core.RhapsodyAppServer
```

public class **RhapsodyAppServer**
extends java.lang.Object

The RhapsodyAppServer class contains methods relating to accessing an instance of Rhapsody.

Field Summary

protected static com.telelogic.rhapsody.core.JavaPluginsManager	**m_javaPluginsManager** For internal use only.

Constructor Summary

RhapsodyAppServer()

Method Summary

static IRPApplication	**actualCreateRhapsodyApplicationDllServer**() For internal use only.
static IRPApplication	**actualCreateUninitializedRhapsodyApplicationDllS** For internal use only.
static void	**actualInitializeRhapsodyApplicationDllServer**(lon For internal use only.
static boolean	**addToClassPath**(java.lang.String cls) Adds one or more classes to the classpath.
static boolean	**addToLibPath**(java.lang.String libPath) Adds a directory to the libpath.
static IRPModelElement	**attachToIRPModelElement**(long nativeRPModelElement For internal use only.
static IRPApplication	**attachToRhapsodyApplication**(long nativeRhapsodyA For internal use only.

setTypeDeclaration   737

Method Summary

static void	**CloseSession**() For internal use only.
static void	**CloseSessionNative**() For internal use only.
protected static void	**createJavaPluginManager**(IRPApplication rhpApp) For internal use only.
static IRPApplication	**createRhapsodyApplication**() Creates a new instance of Rhapsody and provides acce
static IRPApplication	**createRhapsodyApplicationDllServer**() For internal use only.
static IRPApplication	**createUninitializedRhapsodyApplicationDllServer**(For internal use only.
static java.lang.Class	**findClass**(java.lang.String className) For internal use only.
static IRPApplication	**getActiveRhapsodyApplication**() Accesses the currently running instance of Rhapsody.
static IRPApplication	**getActiveRhapsodyApplicationByID**(java.lang.String Accesses the instance of Rhapsody that is registered in Object Table) with the specified ID.
static java.util.List	**getActiveRhapsodyApplicationIDList**() Returns a list of the strings representing the Rhapsody registered in the ROT (Running Object Table).
static com.telelogic.rhapsody.core.JavaPluginsManager	**getJavaPluginManager**() For internal use only.
static void	**initializeRhapsodyApplicationDllServer**(IRPApplic For internal use only.
static boolean	**registerAsActiveObject**(IRPApplication app) For internal use only.
static void	**resetCurrentContextClassFactory**() For internal use only.
protected static void	**setClassFactory**(RPExtendedRPClassesFactory facto boolean isDefaultFactory) For internal use only.
static void	**setCollectionCachingMode**(boolean mode) For internal use only.
static void	**setCollectionCashingMode**(boolean mode) For internal use only.
static void	**setCurrentContextClassFactory**(RPExtendedRPClasse For internal use only.

Method Summary

static void	**setDefultClassFactory**([RPExtendedRPClassesFactory](#) For internal use only.
static void	**setDelayedReleaseInterfacesMode**(boolean mode) For internal use only.
static void	**setLogFile**(java.lang.String logFile) Specifies a log file to use for recording API actions.
static void	**setReleaseInterfacesOnGBMode**(boolean mode) For internal use only.
static boolean	**unRegisterAsActiveObject**([IRPApplication](#) app) For internal use only.
static void	**writeToLog**(java.lang.String msg) Writes the specified text to the Rhapsody API log file.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

m_javaPluginsManager

protected static com.telelogic.rhapsody.core.JavaPluginsManager **m_javaPluginsManager**

> For internal use only.

Constructor Detail

RhapsodyAppServer

public **RhapsodyAppServer**()

Method Detail

addToClassPath

public static boolean **addToClassPath**(java.lang.String cls)

> Adds one or more classes to the classpath.

Parameters:

  `cls` - the path and name of the class or .jar file to add to the classpath, for example, "D:\\myclasses org.mypackage.MyClass" or "D:\\myclasses\\libraryFiles.jar"

Returns:

  true if the classpath was modified successfully, false otherwise

---

addToLibPath

public static boolean **addToLibPath**(java.lang.String libPath)

 Adds a directory to the libpath.

 Parameters:

  `libPath` - the directory to add to the libpath

 Returns:

  true if the libpath was modified successfully, false otherwise

---

attachToIRPModelElement

public static [IRPModelElement](#) **attachToIRPModelElement**(long nativeRPModelElement)

 For internal use only.

---

attachToRhapsodyApplication

public static [IRPApplication](#) **attachToRhapsodyApplication**(long nativeRhapsodyApp)

 For internal use only.

CloseSession

public static void **CloseSession**()

 For internal use only.

CloseSessionNative

public static void **CloseSessionNative**()

 For internal use only.

---

createRhapsodyApplication

public static [IRPApplication](#) **createRhapsodyApplication**()

Creates a new instance of Rhapsody and provides access to it. If you start Rhapsody with this method, you can display the GUI by calling IRPApplication.bringWindowToTop. You can terminate Rhapsody by calling IRPApplication.quit.

Returns:
> the IRPApplication object that represents the new instance of Rhapsody

---

createRhapsodyApplicationDllServer

public static [IRPApplication](#) **createRhapsodyApplicationDllServer**()

> For internal use only.

---

actualCreateRhapsodyApplicationDllServer

public static [IRPApplication](#) **actualCreateRhapsodyApplicationDllServer**()

> For internal use only.

---

createUninitializedRhapsodyApplicationDllServer

public static [IRPApplication](#) **createUninitializedRhapsodyApplicationDllServer**()

> For internal use only. returns an interface to **uninitialized** application

---

actualCreateUninitializedRhapsodyApplicationDllServer

public static [IRPApplication](#) **actualCreateUninitializedRhapsodyApplicationDllServer**()

> For internal use only.

---

initializeRhapsodyApplicationDllServer

public static void **initializeRhapsodyApplicationDllServer**([IRPApplication](#) rhpApp)

> For internal use only. initializations for an uninitialized application

---

actualInitializeRhapsodyApplicationDllServer

public static void **actualInitializeRhapsodyApplicationDllServer**(long comInterface)

> For internal use only.

---

findClass

```
public static java.lang.Class findClass(java.lang.String className)
```

>   For internal use only.

---

getActiveRhapsodyApplication

```
public static IRPApplication getActiveRhapsodyApplication()
```

>   Accesses the currently running instance of Rhapsody.
>   **Returns:**
>   >   the IRPApplication object that represents the instance of Rhapsody that is running

```
 static IRPApplication app = RhapsodyAppServer.getActiveRhapsodyApplication();
 if(app != null) {
    app.createNewProject("d:\\temp\\_sample_code", "Class_Tricks");
    IRPProject prj = app.openProject("d:\\temp\\_sample_code\\Class_Tricks.rpy");
 }
```

---

getActiveRhapsodyApplicationIDList

```
public static java.util.List getActiveRhapsodyApplicationIDList()
```

>   Returns a list of the strings representing the Rhapsody instances currently registered in the ROT
>   (Running Object Table). This method can be used in conjunction with the method
>   getActiveRhapsodyApplicationByID in order to communicate with a specific instance of Rhapsody
>   when there is more than one instance running.
>   **Returns:**
>   >   a list of the strings representing the Rhapsody instances currently registered in the ROT

---

getActiveRhapsodyApplicationByID

```
public static IRPApplication getActiveRhapsodyApplicationByID(java.lang.String serverName)
```

>   Accesses the instance of Rhapsody that is registered in the ROT (Running Object Table) with the
>   specified ID.
>   **Parameters:**
>   >   serverName - the ID of the Rhapsody instance in the ROT, as returned by the method
>   >   getActiveRhapsodyApplicationIDList. The strings used for instances of Rhapsody take the
>   >   form Rhapsody.Release:PID, for example, "Rhapsody.8.0.2.0:12236"
>   **Returns:**
>   >   the IRPApplication object that represents the specified instance of Rhapsody

resetCurrentContextClassFactory

`public static void` **`resetCurrentContextClassFactory`**`()`

> For internal use only.

setClassFactory

`protected static void` **`setClassFactory`**`(`[`RPExtendedRPClassesFactory`](#) `factory,`
                                `boolean isDefaultFactory)`

> For internal use only.

setCollectionCachingMode

`public static void` **`setCollectionCachingMode`**`(boolean mode)`

> For internal use only.

setCollectionCashingMode

`public static void` **`setCollectionCashingMode`**`(boolean mode)`

> For internal use only.

setCurrentContextClassFactory

`public static void` **`setCurrentContextClassFactory`**`(`[`RPExtendedRPClassesFactory`](#) `factory)`

> For internal use only.

setDefultClassFactory

`public static void` **`setDefultClassFactory`**`(`[`RPExtendedRPClassesFactory`](#) `factory)`

> For internal use only.

setDelayedReleaseInterfacesMode

`public static void` **`setDelayedReleaseInterfacesMode`**`(boolean mode)`

> For internal use only.

setLogFile

`public static void` **`setLogFile`**`(java.lang.String logFile)`

> Specifies a log file to use for recording API actions. You can write text to the log file using the writeToLog method. Note that if you use this method to specify a log file, that log file will be used even if your rhapsody.ini file specifies a different file with the JavaAPILogFile variable. You can stop logging by calling setLogFile with null as the parameter.
>
> **Parameters:**
> > `logFile` - the full path of the file to use as the log file, for example, "D:\\temp\\api_log_file.txt".

writeToLog

`public static void` **`writeToLog`**`(java.lang.String msg)`

> Writes the specified text to the Rhapsody API log file. Note that this will work only if you first call the method RhapsodyAppServer.setLogFile. It will not write to a log file that was specified using the JavaAPILogFile variable in the rhapsody.ini file.
>
> **Parameters:**
> > `msg` - the text to write to the log file

setReleaseInterfacesOnGBMode

`public static void` **`setReleaseInterfacesOnGBMode`**`(boolean mode)`

> For internal use only.

createJavaPluginManager

`protected static void` **`createJavaPluginManager`**`(`[IRPApplication](#) ` rhpApp)`

> For internal use only.

registerAsActiveObject

`public static boolean` **`registerAsActiveObject`**`(`[IRPApplication](#) ` app)`

> For internal use only.

unRegisterAsActiveObject

`public static boolean` **`unRegisterAsActiveObject`**`(`[IRPApplication](#) ` app)`

> For internal use only.

getJavaPluginManager

`public static com.telelogic.rhapsody.core.JavaPluginsManager` **getJavaPluginManager**()

> For internal use only.

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Class RhapsodyRuntimeException

```
java.lang.Object
  └─ java.lang.Throwable
       └─ java.lang.Exception
            └─ java.lang.RuntimeException
                 └─ com.telelogic.rhapsody.core.RhapsodyRuntimeException
```

All Implemented Interfaces:
>   java.io.Serializable

---

public class **RhapsodyRuntimeException**
extends java.lang.RuntimeException

See Also:
>   Serialized Form

Constructor Summary

RhapsodyRuntimeException(java.lang.String desc)	

Method Summary

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

RhapsodyRuntimeException

```
public RhapsodyRuntimeException(java.lang.String desc)
```

com.telelogic.rhapsody.core
Class RhpClassLoader

```
java.lang.Object
  └─ java.lang.ClassLoader
       └─ java.security.SecureClassLoader
            └─ java.net.URLClassLoader
                 └─ com.telelogic.rhapsody.core.RhpClassLoader
```

---

public class **RhpClassLoader**
extends java.net.URLClassLoader

Constructor Summary

RhpClassLoader(java.net.URL[] urls)	

Method Summary

protected java.lang.String	**findLibrary**(java.lang.String libName)
void	**setLocalLibPath**(java.lang.String path)

Methods inherited from class java.net.URLClassLoader

addURL, definePackage, findClass, findResource, findResources, getPermissions, getURLs, newInstance, newInstance

Methods inherited from class java.security.SecureClassLoader

defineClass, defineClass

Methods inherited from class java.lang.ClassLoader

clearAssertionStatus, defineClass, defineClass, defineClass, defineClass, definePackage, findLoadedClass, findSystemClass, getPackage, getPackages, getParent, getResource, getResourceAsStream, getResources, getSystemClassLoader, getSystemResource, getSystemResourceAsStream, getSystemResources, loadClass, loadClass, resolveClass, setClassAssertionStatus, setDefaultAssertionStatus, setPackageAssertionStatus, setSigners

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

Constructor Detail

RhpClassLoader

```
public RhpClassLoader(java.net.URL[] urls)
```

Method Detail

findLibrary

```
protected java.lang.String findLibrary(java.lang.String libName)
```

> **Overrides:**
> findLibrary in class java.lang.ClassLoader

---

setLocalLibPath

```
public void setLocalLibPath(java.lang.String path)
```

---

com.telelogic.rhapsody.core
Class RhpUtils

```
java.lang.Object
  └─ com.telelogic.rhapsody.core.RhpUtils
```

public class **RhpUtils**
extends java.lang.Object

Constructor Summary

RhpUtils()	

Method Summary

static boolean	**addToClassPath**(java.lang.String cls) Add class path to the system's class path
static boolean	**addToLibPath**(java.lang.String libPath)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

RhpUtils

public **RhpUtils**()

Method Detail

setLocalLibPath   750

addToLibPath

```
public static boolean addToLibPath(java.lang.String libPath)
```

addToClassPath

```
public static boolean addToClassPath(java.lang.String cls)
```

>
> Add class path to the system's class path
> **Returns:**
> > true if succeed

| **Package** | **Class** | **Use** | **Tree** | **Serialized** | **Deprecated** | **Index** | **Help** |

PREV CLASS **NEXT CLASS**   **FRAMES**   **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Class RPApplicationListener

```
java.lang.Object
  └─com.telelogic.rhapsody.core.RPApplicationListener
```

public abstract class **RPApplicationListener**
extends java.lang.Object

Constructor Summary

RPApplicationListener()	

Method Summary

boolean	**activeProjectAboutToChange**(IRPProject project)
boolean	**activeProjectHasChanged**(IRPProject project)
abstract boolean	**afterAddElement**(IRPModelElement pModelElement) Called after element is added
boolean	**afterApplicationClosed**()
abstract boolean	**afterProjectClose**(java.lang.String bstrProjectName) Called after project is closed
boolean	**afterProjectOpen**(IRPProject project)
boolean	**afterProjectSaved**(IRPProject project)
boolean	**beforeApplicationClosed**()
abstract boolean	**beforeProjectClose**(IRPProject pProject) Called before project is closed
boolean	**beforeProjectOpen**(java.lang.String projectPath)

Method Summary

boolean	**beforeProjectSaved**([IRPProject](#) project)
boolean	**connect**([IRPApplication](#) connectionPoint)
boolean	**disconnect**()
protected void	**finalize**()
abstract java.lang.String	**getId**() Gets the id of the listener
abstract boolean	**onDiagramOpen**([IRPDiagram](#) pDiagram) Called when diagram is opened
abstract boolean	**onDoubleClick**([IRPModelElement](#) pModelElement) Called on double click
boolean	**onElementsChanged**(java.lang.String elementsGUIDs)
abstract boolean	**onFeaturesOpen**([IRPModelElement](#) pModelElement) Called when element features dialog is opened
boolean	**onPerspectiveChange**(java.lang.String oldPerspective, java.lang.String newPerspective)
boolean	**onSelectionChanged**()

Methods inherited from class java.lang.Object

clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

RPApplicationListener

public **RPApplicationListener**()

Method Detail

connect

```
public boolean connect(IRPApplication connectionPoint)
```

disconnect

```
public boolean disconnect()
```

finalize

```
protected void finalize()
                throws java.lang.Throwable
```

> **Overrides:**
> finalize in class java.lang.Object
> **Throws:**
> java.lang.Throwable

afterAddElement

```
public abstract boolean afterAddElement(IRPModelElement pModelElement)
```

> Called after element is added
> **Throws:**
> RhapsodyRuntimeException

afterProjectClose

```
public abstract boolean afterProjectClose(java.lang.String bstrProjectName)
```

> Called after project is closed
> **Throws:**
> RhapsodyRuntimeException

beforeProjectClose

```
public abstract boolean beforeProjectClose(IRPProject pProject)
```

> Called before project is closed
> **Throws:**
> RhapsodyRuntimeException

getId

`public abstract java.lang.String` **`getId`**`()`

Gets the id of the listener
Throws:
RhapsodyRuntimeException

---

onDiagramOpen

`public abstract boolean` **`onDiagramOpen`**`(`IRPDiagram` pDiagram)`

Called when diagram is opened
Throws:
RhapsodyRuntimeException

---

onDoubleClick

`public abstract boolean` **`onDoubleClick`**`(`IRPModelElement` pModelElement)`

Called on double click
Throws:
RhapsodyRuntimeException

---

onFeaturesOpen

`public abstract boolean` **`onFeaturesOpen`**`(`IRPModelElement` pModelElement)`

Called when element features dialog is opened
Throws:
RhapsodyRuntimeException

---

activeProjectAboutToChange

`public boolean` **`activeProjectAboutToChange`**`(`IRPProject` project)`

---

activeProjectHasChanged

`public boolean` **`activeProjectHasChanged`**`(`IRPProject` project)`

---

afterApplicationClosed

`public boolean` **`afterApplicationClosed`**`()`

---

afterProjectOpen

```
public boolean afterProjectOpen(IRPProject project)
```

afterProjectSaved

```
public boolean afterProjectSaved(IRPProject project)
```

beforeApplicationClosed

```
public boolean beforeApplicationClosed()
```

beforeProjectOpen

```
public boolean beforeProjectOpen(java.lang.String projectPath)
```

beforeProjectSaved

```
public boolean beforeProjectSaved(IRPProject project)
```

onElementsChanged

```
public boolean onElementsChanged(java.lang.String elementsGUIDs)
```

onPerspectiveChange

```
public boolean onPerspectiveChange(java.lang.String oldPerspective,
                                   java.lang.String newPerspective)
```

onSelectionChanged

```
public boolean onSelectionChanged()
```

Package   **Class**   **Use**   **Tree**   **Serialized**   **Deprecated**   **Index**   **Help**

PREV CLASS   **NEXT CLASS**   **FRAMES**   **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

afterProjectOpen   756

com.telelogic.rhapsody.core
Class RPCodeGeneratorListener

```
java.lang.Object
  └─ com.telelogic.rhapsody.core.RPCodeGeneratorListener
```

public abstract class **RPCodeGeneratorListener**
extends java.lang.Object

Constructor Summary

RPCodeGeneratorListener()	

Method Summary

boolean	**connect**([IRPCodeGenerator](#) connectionPoint)
boolean	**disconnect**()
protected void	**finalize**()
abstract java.lang.String	**getId**() Gets the id of the listener
abstract void	**onCodeGenerationCompleted**() Called after code generation is completed

Methods inherited from class java.lang.Object

clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

RPCodeGeneratorListener

public **RPCodeGeneratorListener**()

Method Detail

connect

public boolean **connect**([IRPCodeGenerator](#) connectionPoint)

---

disconnect

public boolean **disconnect**()

---

finalize

```
protected void finalize()
                throws java.lang.Throwable
```

Overrides:
        finalize in class java.lang.Object
Throws:
        java.lang.Throwable

---

getId

public abstract java.lang.String **getId**()

Gets the id of the listener
Throws:
        [RhapsodyRuntimeException](#)

---

onCodeGenerationCompleted

public abstract void **onCodeGenerationCompleted**()

Called after code generation is completed
Throws:
        [RhapsodyRuntimeException](#)

[Package](#)   Class   [Use](#)   [Tree](#)   [Serialized](#)   [Deprecated](#)   [Index](#)   [Help](#)
[PREV CLASS](#)   [NEXT CLASS](#)   **[FRAMES](#)   [NO FRAMES](#)   [All Classes](#)**
SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)   DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

RPCodeGeneratorListener   758

Package **Class** **Use** **Tree** **Serialized** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Class RPCodeGenSimplifier

```
java.lang.Object
  └─ com.telelogic.rhapsody.core.RPCodeGenSimplifier
```

public abstract class **RPCodeGenSimplifier**
extends java.lang.Object

Constructor Summary

RPCodeGenSimplifier()	

Method Summary

abstract void	**beginSimplification**() before all simplifications
boolean	**connect**(IRPCodeGenSimplifiersRegistry connectionPoint)
boolean	**disconnect**()
abstract void	**doAbort**() abort the simplification
abstract void	**doExit**() exit and allow Rhapsody to exit
abstract void	**endSimplification**() after all simplifications
protected void	**finalize**()
abstract void	**postSimplify**(IRPModelElement userElement, IRPModelElement mainSimplifiedElement, java.lang.String simplificationRequested) post element simplification
abstract void	**simplify**(IRPModelElement userElement, IRPModelElement simplifiedElementOwner, java.lang.String simplificationRequested)

Method Summary

	simplify the user element

Methods inherited from class java.lang.Object

clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

RPCodeGenSimplifier

public **RPCodeGenSimplifier**()

Method Detail

connect

public boolean **connect**([IRPCodeGenSimplifiersRegistry](#) connectionPoint)

disconnect

public boolean **disconnect**()

finalize

```
protected void finalize()
              throws java.lang.Throwable
```

Overrides:
finalize in class java.lang.Object
Throws:
java.lang.Throwable

beginSimplification

public abstract void **beginSimplification**()

before all simplifications
Throws:
[RhapsodyRuntimeException](#)

doAbort

public abstract void **doAbort**()

> abort the simplification
> **Throws:**
>> RhapsodyRuntimeException

---

doExit

public abstract void **doExit**()

> exit and allow Rhapsody to exit
> **Throws:**
>> RhapsodyRuntimeException

---

endSimplification

public abstract void **endSimplification**()

> after all simplifications
> **Throws:**
>> RhapsodyRuntimeException

---

postSimplify

public abstract void **postSimplify**(IRPModelElement userElement,
                                    IRPModelElement mainSimplifiedElement,
                                    java.lang.String simplificationRequested)

> post element simplification
> **Throws:**
>> RhapsodyRuntimeException

---

simplify

public abstract void **simplify**(IRPModelElement userElement,
                                IRPModelElement simplifiedElementOwner,
                                java.lang.String simplificationRequested)

> simplify the user element
> **Throws:**
>> RhapsodyRuntimeException

---

com.telelogic.rhapsody.core
Class RPExtendedRPClassesFactory

```
java.lang.Object
  └ com.telelogic.rhapsody.core.RPExtendedRPClassesFactory
```

public abstract class **RPExtendedRPClassesFactory**
extends java.lang.Object

Field Summary

static java.lang.Object	**factoryMutex**

Constructor Summary

RPExtendedRPClassesFactory()

Method Summary

abstract java.lang.Class	**getExtendedClass**(java.lang.String RPClassName)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

factoryMutex

public static java.lang.Object **factoryMutex**

Constructor Detail

RPExtendedRPClassesFactory

```
public RPExtendedRPClassesFactory()
```

Method Detail

getExtendedClass

```
public abstract java.lang.Class getExtendedClass(java.lang.String RPClassName)
```

Package   **Class**   **Use**   **Tree**   **Serialized**   **Deprecated**   **Index**   **Help**

PREV CLASS   **NEXT CLASS**   **FRAMES**   **NO FRAMES**   **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

---

com.telelogic.rhapsody.core
Class RPExternalCheck

```
java.lang.Object
  └─ com.telelogic.rhapsody.core.RPExternalCheck
```

public abstract class **RPExternalCheck**
extends java.lang.Object

Constructor Summary

RPExternalCheck()	

Method Summary

abstract boolean	**check**([IRPModelElement](#) ElementToCheck, [IRPCollection](#) FailedElements) 　　　Called by Rhapsody to execute the check.
boolean	**connect**([IRPExternalCheckRegistry](#) connectionPoint)
boolean	**disconnect**()
abstract void	**doExit**() 　　　exit and allow Rhapsody to exit
protected void	**finalize**()
abstract boolean	**getCompleteness**() 　　　Return true if this is a check for completeness or false if this is a check for correctness
abstract java.lang.String	**getDomain**() 　　　Return the domain of the check which can be user defined or one from predefined list of , or .
abstract java.lang.String	**getMetaclasses**() 　　　Return a comma separated list of metaclasses or new terms - Rhapsody will call check for all elements in scope of check that are of the metaclass type in the list
abstract java.lang.String	

getExtendedClass　　　　　　　　　　　　　　　　　　　　　　　　　　766

Method Summary

	getName() Return the name of the check (also used as its error message)
abstract java.lang.String	**getSeverity**() Return the Severity of the check which one from predefined list of , ,
abstract boolean	**getShouldCallFromCG**() Return true if this check should be automatically called before code generation

Methods inherited from class java.lang.Object

clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

RPExternalCheck

public **RPExternalCheck**()

Method Detail

connect

public boolean **connect**(IRPExternalCheckRegistry connectionPoint)

---

disconnect

public boolean **disconnect**()

---

finalize

protected void **finalize**()
                throws java.lang.Throwable

Overrides:
        finalize in class java.lang.Object
Throws:
        java.lang.Throwable

---

check

```
public abstract boolean check(IRPModelElement ElementToCheck,
                              IRPCollection FailedElements)
```

Called by Rhapsody to execute the check. Return a list of elements to highlight if check fails (or empty list if the check is OK)
Throws:
        RhapsodyRuntimeException

---

getCompleteness

```
public abstract boolean getCompleteness()
```

Return true if this is a check for completeness or false if this is a check for correctness
Throws:
        RhapsodyRuntimeException

---

getDomain

```
public abstract java.lang.String getDomain()
```

Return the domain of the check which can be user defined or one from predefined list of , or . (For RIC)
Throws:
        RhapsodyRuntimeException

---

getMetaclasses

```
public abstract java.lang.String getMetaclasses()
```

Return a comma separated list of metaclasses or new terms - Rhapsody will call check for all elements in scope of check that are of the metaclass type in the list
Throws:
        RhapsodyRuntimeException

---

getName

```
public abstract java.lang.String getName()
```

Return the name of the check (also used as its error message)
Throws:
        RhapsodyRuntimeException

---

getSeverity

`public abstract java.lang.String` **`getSeverity`**`()`

> Return the Severity of the check which one from predefined list of , ,
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

getShouldCallFromCG

`public abstract boolean` **`getShouldCallFromCG`**`()`

> Return true if this check should be automatically called before code generation
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

doExit

`public abstract void` **`doExit`**`()`

> exit and allow Rhapsody to exit
> **Throws:**
> > [RhapsodyRuntimeException](#)

---

com.telelogic.rhapsody.core
Class RPExternalCodeGenerator

```
java.lang.Object
  └─ com.telelogic.rhapsody.core.RPExternalCodeGenerator
```

public abstract class **RPExternalCodeGenerator**
extends java.lang.Object

Constructor Summary

RPExternalCodeGenerator()	

Method Summary

abstract void	**abort**() method Abort
boolean	**connect**(<u>IRPExternalCodeGeneratorInvoker</u> connectionPoint)
boolean	**disconnect**()
abstract void	**exit**() method Exit
protected void	**finalize**()
abstract void	**generate**(<u>IRPModelElement</u> activeConfiguration, <u>IRPCollection</u> classifiersCollection, <u>IRPCollection</u> filesCollection, int generateMainFile, int generateMakefile) method Generate
abstract java.lang.String	**getFileName**(<u>IRPModelElement</u> modelElement, <u>IRPModelElement</u> configuration, int pathType, int withExtensions) method GetFileName
abstract java.lang.String	**getMainFileName**(<u>IRPModelElement</u> configuration, int pathType, int withExtensions) method GetMainFileName

Method Summary

abstract java.lang.String	**getMakefileName**([IRPModelElement](#) configuration, int pathType, int withExtension) method GetMakefileName
abstract java.lang.String	**getTargetfileName**([IRPModelElement](#) configuration, int pathType, int withExtension) method GetTargetfileName
abstract java.lang.String	**whoAmI**() method WhoAmI

Methods inherited from class java.lang.Object

clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

RPExternalCodeGenerator

public **RPExternalCodeGenerator**()

Method Detail

connect

public boolean **connect**([IRPExternalCodeGeneratorInvoker](#) connectionPoint)

---

disconnect

public boolean **disconnect**()

---

finalize

protected void **finalize**()
                throws java.lang.Throwable

Overrides:
        finalize in class java.lang.Object
Throws:
        java.lang.Throwable

---

abort

```
public abstract void abort()
```

> method Abort
> **Throws:**
> > RhapsodyRuntimeException

exit

```
public abstract void exit()
```

> method Exit
> **Throws:**
> > RhapsodyRuntimeException

getMainFileName

```
public abstract java.lang.String getMainFileName(IRPModelElement configuration,
                                                  int pathType,
                                                  int withExtensions)
```

> method GetMainFileName
> **Throws:**
> > RhapsodyRuntimeException

getTargetfileName

```
public abstract java.lang.String getTargetfileName(IRPModelElement configuration,
                                                    int pathType,
                                                    int withExtension)
```

> method GetTargetfileName
> **Throws:**
> > RhapsodyRuntimeException

whoAmI

```
public abstract java.lang.String whoAmI()
```

> method WhoAmI
> **Throws:**
> > RhapsodyRuntimeException

generate

```
public abstract void generate(IRPModelElement activeConfiguration,
                              IRPCollection classifiersCollection,
                              IRPCollection filesCollection,
                              int generateMainFile,
                              int generateMakefile)
```

method Generate
Throws:
  RhapsodyRuntimeException

getFileName

```
public abstract java.lang.String getFileName(IRPModelElement modelElement,
                                             IRPModelElement configuration,
                                             int pathType,
                                             int withExtensions)
```

method GetFileName
Throws:
  RhapsodyRuntimeException

getMakefileName

```
public abstract java.lang.String getMakefileName(IRPModelElement configuration,
                                                 int pathType,
                                                 int withExtension)
```

method GetMakefileName
Throws:
  RhapsodyRuntimeException

com.telelogic.rhapsody.core
Class RPExternalIDEManager

```
java.lang.Object
  └─ com.telelogic.rhapsody.core.RPExternalIDEManager
```

public abstract class **RPExternalIDEManager**
extends java.lang.Object

Constructor Summary

RPExternalIDEManager()

Method Summary

abstract void	**activateView**(<u>IRPAXViewCtrl</u> RhapsodyView) Activate view
abstract void	**closeDiagram**(<u>IRPDiagram</u> diagram) Closes diagram if opened
boolean	**connect**(<u>IRPExternalIDERegistry</u> connectionPoint)
abstract int	**createProgressTask**(int nGroupNumber, int nTaskNumber, java.lang.String sTaskName, int nTaskLength, int bCanCancel) Create a Progress Task
boolean	**disconnect**()
protected void	**finalize**()
abstract void	**finishProgressTask**(int nGroupNumber, int nTaskNumber) Finish a Progress Task
abstract <u>IRPAXViewCtrl</u>	**getActiveView**() method GetActiveView
abstract int	**isProgressTaskCanceled**(int nGroupNumber, int nTaskNumber) Check if a Progress Task is canceled
abstract void	

getMakefileName   774

Method Summary

	onIDETextMessage(java.lang.String message) method OnIDETextMessage
abstract void	**onInvokeSearch**(IRPModelElement lookinElement) method OnSearchRequest
abstract void	**onNotifyMessage**(java.lang.String messageType, IRPCollection pMessageInitialization, IRPCollection pMessageResult) method OnNotifyMessage
abstract void	**onShowInUnitView**(IRPModelElement modelElement) method OnShowInUnitView
abstract void	**openDiagram**(IRPDiagram diagram) method OpenDiagram
abstract void	**openFile**(java.lang.String filename) Opens
abstract void	**openFileAndSelectLine**(java.lang.String filename, int line) Opens file and selects line
abstract void	**openHotFeatures**() method OpenHotFeatures
abstract void	**openNewFeatures**(IRPModelElement element) method OpenNewFeatures
abstract java.lang.String	**openYesNoCancelQuestion**(java.lang.String dialogTitle, java.lang.String message, java.lang.String toggleMessage, int toggleState) Display YES OCancel message box with check-box to remmember the chosen reply
abstract void	**progressTaskStep**(int nGroupNumber, int nTaskNumber, int a_nStepsDone) Indicate a Progress Task step performed
abstract void	**refreshRequest**() Refresh Rhapsody project/workspace contents
abstract void	**setProcessSubTaskName**(int nGroupNumber, int nTaskNumber, java.lang.String sSubTaskName) Set a Progress Task subtask's name
abstract void	**showBrowser**(int showOrHide) ShowBrowser
abstract void	**showStatusBarMessage**(java.lang.String message) Display message in status bar

Methods inherited from class java.lang.Object

clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

RPExternalIDEManager

public **RPExternalIDEManager**()

Method Detail

connect

public boolean **connect**([IRPExternalIDERegistry](#) connectionPoint)

---

disconnect

public boolean **disconnect**()

---

finalize

```
protected void finalize()
                  throws java.lang.Throwable
```

> **Overrides:**
> finalize in class java.lang.Object
> **Throws:**
> java.lang.Throwable

---

activateView

public abstract void **activateView**([IRPAXViewCtrl](#) RhapsodyView)

> Activate view
> **Throws:**
> [RhapsodyRuntimeException](#)

---

createProgressTask

```
public abstract int createProgressTask(int nGroupNumber,
                                       int nTaskNumber,
                                       java.lang.String sTaskName,
                                       int nTaskLength,
                                       int bCanCancel)
```

> Create a Progress Task
> **Throws:**

finishProgressTask

```
public abstract void finishProgressTask(int nGroupNumber,
                                        int nTaskNumber)
```

Finish a Progress Task
Throws:
>   RhapsodyRuntimeException

getActiveView

```
public abstract IRPAXViewCtrl getActiveView()
```

method GetActiveView
Throws:
>   RhapsodyRuntimeException

isProgressTaskCanceled

```
public abstract int isProgressTaskCanceled(int nGroupNumber,
                                           int nTaskNumber)
```

Check if a Progress Task is canceled
Throws:
>   RhapsodyRuntimeException

onIDETextMessage

```
public abstract void onIDETextMessage(java.lang.String message)
```

method OnIDETextMessage
Throws:
>   RhapsodyRuntimeException

onInvokeSearch

```
public abstract void onInvokeSearch(IRPModelElement lookinElement)
```

method OnSearchRequest
Throws:
>   RhapsodyRuntimeException

onNotifyMessage

```
public abstract void onNotifyMessage(java.lang.String messageType,
                                     IRPCollection pMessageInitialization,
                                     IRPCollection pMessageResult)
```

method OnNotifyMessage
Throws:
    RhapsodyRuntimeException

onShowInUnitView

```
public abstract void onShowInUnitView(IRPModelElement modelElement)
```

method OnShowInUnitView
Throws:
    RhapsodyRuntimeException

openFile

```
public abstract void openFile(java.lang.String filename)
```

Opens
Throws:
    RhapsodyRuntimeException

openFileAndSelectLine

```
public abstract void openFileAndSelectLine(java.lang.String filename,
                                           int line)
```

Opens file and selects line
Throws:
    RhapsodyRuntimeException

openHotFeatures

```
public abstract void openHotFeatures()
```

method OpenHotFeatures
Throws:
    RhapsodyRuntimeException

openNewFeatures

public abstract void **openNewFeatures**(<u>IRPModelElement</u> element)

method OpenNewFeatures
Throws:
     <u>RhapsodyRuntimeException</u>

---

openYesNoCancelQuestion

public abstract java.lang.String **openYesNoCancelQuestion**(java.lang.String dialogTitle,
                                           java.lang.String message,
                                           java.lang.String toggleMessage,
                                           int toggleState)

Display YES OCancel message box with check-box to remmember the chosen reply
Throws:
     <u>RhapsodyRuntimeException</u>

---

progressTaskStep

public abstract void **progressTaskStep**(int nGroupNumber,
                              int nTaskNumber,
                              int a_nStepsDone)

Indicate a Progress Task step performed
Throws:
     <u>RhapsodyRuntimeException</u>

---

refreshRequest

public abstract void **refreshRequest**()

Refresh Rhapsody project/workspace contents
Throws:
     <u>RhapsodyRuntimeException</u>

---

setProcessSubTaskName

public abstract void **setProcessSubTaskName**(int nGroupNumber,
                                        int nTaskNumber,
                                        java.lang.String sSubTaskName)

Set a Progress Task subtask's name
Throws:
     <u>RhapsodyRuntimeException</u>

---

showBrowser

public abstract void **showBrowser**(int showOrHide)

ShowBrowser
Throws:
RhapsodyRuntimeException

showStatusBarMessage

public abstract void **showStatusBarMessage**(java.lang.String message)

Display message in status bar
Throws:
RhapsodyRuntimeException

closeDiagram

public abstract void **closeDiagram**(IRPDiagram diagram)

Closes diagram if opened
Throws:
RhapsodyRuntimeException

openDiagram

public abstract void **openDiagram**(IRPDiagram diagram)

method OpenDiagram
Throws:
RhapsodyRuntimeException

com.telelogic.rhapsody.core
Class RPExternalRoundtrip

```
java.lang.Object
    └─com.telelogic.rhapsody.core.RPExternalRoundtrip
```

public abstract class **RPExternalRoundtrip**
extends java.lang.Object

Constructor Summary

RPExternalRoundtrip()

Method Summary

boolean	**connect**(<u>IRPExternalRoundtripInvoker</u> connectionPoint)
abstract <u>IRPModelElement</u>	**createCodeModel**(<u>IRPCollection</u> roundTrippedFileList) method CreateCodeModel
boolean	**disconnect**()
protected void	**finalize**()
abstract int	**isModelChanged**() property isModelChanged
abstract int	**okToAddAggregate**(<u>IRPModelElement</u> code_aggregate, <u>IRPModelElement</u> model_parent) method OkToAddAggregate
abstract int	**okToMakeAction**(<u>IRPModelElement</u> model_object, java.lang.String action) method OkToMakeAction
abstract int	**shouldAddAggregate**(<u>IRPModelElement</u> code_aggregate, <u>IRPModelElement</u> model_parent) method ShouldAddAggregate
abstract int	**shouldMergeAggregate**(<u>IRPModelElement</u> model_aggregate, <u>IRPModelElement</u> model_parent, <u>IRPModelElement</u> code_aggregate, <u>IRPModelElement</u> code_parent)

openDiagram 781

Method Summary

		method ShouldMergeAggregate
abstract	int	**shouldMergeAssociation**(java.lang.String assoc_name, [IRPModelElement](#) model_assoc, [IRPModelElement](#) code_assoc, [IRPModelElement](#) model_object, [IRPModelElement](#) code_object) method ShouldMergeAssociation
abstract	int	**shouldMergeAttribute**(java.lang.String attribute_name, java.lang.String model_value, java.lang.String code_value, [IRPModelElement](#) model_object, [IRPModelElement](#) code_object, java.lang.String value) method ShouldMergeAttribute
abstract	int	**shouldRemoveAggregate**([IRPModelElement](#) model_aggregate, [IRPModelElement](#) model_parent, [IRPModelElement](#) code_parent) method ShouldRemoveAggregate

Methods inherited from class java.lang.Object

clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

RPExternalRoundtrip

public **RPExternalRoundtrip**()

Method Detail

connect

public boolean **connect**([IRPExternalRoundtripInvoker](#) connectionPoint)

---

disconnect

public boolean **disconnect**()

---

finalize

protected void **finalize**()
                throws java.lang.Throwable

> **Overrides:**
>   finalize in class java.lang.Object

Throws:
        java.lang.Throwable

createCodeModel

public abstract [IRPModelElement](#) **createCodeModel**([IRPCollection](#) roundTrippedFileList)

method CreateCodeModel
Throws:
        [RhapsodyRuntimeException](#)

okToAddAggregate

public abstract int **okToAddAggregate**([IRPModelElement](#) code_aggregate,
                                    [IRPModelElement](#) model_parent)

method OkToAddAggregate
Throws:
        [RhapsodyRuntimeException](#)

okToMakeAction

public abstract int **okToMakeAction**([IRPModelElement](#) model_object,
                                    java.lang.String action)

method OkToMakeAction
Throws:
        [RhapsodyRuntimeException](#)

shouldAddAggregate

public abstract int **shouldAddAggregate**([IRPModelElement](#) code_aggregate,
                                       [IRPModelElement](#) model_parent)

method ShouldAddAggregate
Throws:
        [RhapsodyRuntimeException](#)

shouldMergeAggregate

public abstract int **shouldMergeAggregate**([IRPModelElement](#) model_aggregate,
                                          [IRPModelElement](#) model_parent,
                                          [IRPModelElement](#) code_aggregate,
                                          [IRPModelElement](#) code_parent)

method ShouldMergeAggregate
Throws:

shouldMergeAssociation

```
public abstract int shouldMergeAssociation(java.lang.String assoc_name,
                                           IRPModelElement model_assoc,
                                           IRPModelElement code_assoc,
                                           IRPModelElement model_object,
                                           IRPModelElement code_object)
```

> method ShouldMergeAssociation
> **Throws:**
> > [RhapsodyRuntimeException](#)

shouldMergeAttribute

```
public abstract int shouldMergeAttribute(java.lang.String attribute_name,
                                         java.lang.String model_value,
                                         java.lang.String code_value,
                                         IRPModelElement model_object,
                                         IRPModelElement code_object,
                                         java.lang.String value)
```

> method ShouldMergeAttribute
> **Throws:**
> > [RhapsodyRuntimeException](#)

shouldRemoveAggregate

```
public abstract int shouldRemoveAggregate(IRPModelElement model_aggregate,
                                          IRPModelElement model_parent,
                                          IRPModelElement code_parent)
```

> method ShouldRemoveAggregate
> **Throws:**
> > [RhapsodyRuntimeException](#)

isModelChanged

```
public abstract int isModelChanged()
```

> property isModelChanged
> **Throws:**
> > [RhapsodyRuntimeException](#)

[Package](#)   Class   [Use](#)   [Tree](#)   [Serialized](#)   [Deprecated](#)   [Index](#)   [Help](#)
[PREV CLASS](#) [NEXT CLASS](#)   **[FRAMES](#)   [NO FRAMES](#)   [All Classes](#)**

shouldMergeAggregate   784

---

com.telelogic.rhapsody.core
Class RPIntegratorListener

```
java.lang.Object
  └ com.telelogic.rhapsody.core.RPIntegratorListener
```

---

public abstract class **RPIntegratorListener**
extends java.lang.Object

Constructor Summary

RPIntegratorListener()

Method Summary

abstract void	**afterImportModel**(IRPModelElement rootElement) Called after import model is completed
boolean	**connect**(IRPIntegrator connectionPoint)
boolean	**disconnect**()
protected void	**finalize**()
abstract java.lang.String	**getId**() Gets the id of the listener
abstract java.lang.String	**subscribedTo**() Get the application's name that Rhapsody integrates with

Methods inherited from class java.lang.Object
clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

RPIntegratorListener

public **RPIntegratorListener**()

Method Detail

connect

public boolean **connect**([IRPIntegrator](#) connectionPoint)

disconnect

public boolean **disconnect**()

finalize

```
protected void finalize()
                  throws java.lang.Throwable
```

> **Overrides:**
>   finalize in class java.lang.Object
> **Throws:**
>   java.lang.Throwable

afterImportModel

public abstract void **afterImportModel**([IRPModelElement](#) rootElement)

> Called after import model is completed
> **Throws:**
>   [RhapsodyRuntimeException](#)

getId

public abstract java.lang.String **getId**()

> Gets the id of the listener
> **Throws:**
>   [RhapsodyRuntimeException](#)

subscribedTo

```
public abstract java.lang.String subscribedTo()
```

> Get the application's name that Rhapsody integrates with
> **Throws:**
>> RhapsodyRuntimeException

| **Package** | **Class** | **Use** | **Tree** | **Serialized** | **Deprecated** | **Index** | **Help** |

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Class RPJavaPluginsManager

```
java.lang.Object
  └─com.telelogic.rhapsody.core.RPJavaPluginsManager
```

public abstract class **RPJavaPluginsManager**
extends java.lang.Object

Constructor Summary

RPJavaPluginsManager()

Method Summary

abstract java.lang.String	**call2StringPluginMethod**(java.lang.String PluginClassName, java.lang.String methodName, java.lang.String argument, java.lang.String exargument) Calls a method of a plugins main class with two strings arguments
abstract boolean	**callElementCollectionPluginMethod**(java.lang.String PluginClassName, java.lang.String methodName, IRPModelElement element, IRPCollection collection) Calls a method of a plugins main class with ModelElement and collection
abstract boolean	**callPluginMethod**(java.lang.String PluginClassName, java.lang.String methodName, IRPCollection args) Calls a method of a plugins main class
abstract java.lang.String	**callStringPluginMethod**(java.lang.String PluginClassName, java.lang.String methodName, java.lang.String argument) Calls a method of a plugins main class with string in/out
boolean	**connect**(IRPJavaPlugins connectionPoint)
boolean	**disconnect**()
protected void	**finalize**()
abstract java.lang.String	**getId**() Gets the id of the listener

Method Summary

abstract boolean	**hasPluginWithMethod**(java.lang.String PluginClassName, java.lang.String methodName) 　　　　Check if a method exists on a plugin
abstract boolean	**hasPluginWithMethodArgs**(java.lang.String PluginClassName, java.lang.String methodName, java.lang.String arguments) 　　　　Check if a method with arguments exists on a plugin
abstract boolean	**loadPlugin**(java.lang.String PluginClassName, IRPCollection classURLS, IRPCollection libURLS) 　　　　Loads the Java plugin main class
abstract boolean	**unloadPlugin**(java.lang.String PluginClassName, int finalCall) 　　　　Unload plugin

Methods inherited from class java.lang.Object

clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

RPJavaPluginsManager

public **RPJavaPluginsManager**()

Method Detail

connect

public boolean **connect**(IRPJavaPlugins connectionPoint)

disconnect

public boolean **disconnect**()

finalize

protected void **finalize**()
                 throws java.lang.Throwable

> **Overrides:**
> 　　finalize in class java.lang.Object
> **Throws:**

RPJavaPluginsManager   790

```
java.lang.Throwable
```

call2StringPluginMethod

```
public abstract java.lang.String call2StringPluginMethod(java.lang.String PluginClassName,
                                                         java.lang.String methodName,
                                                         java.lang.String argument,
                                                         java.lang.String exargument)
```

Calls a method of a plugins main class with two strings arguments
Throws:
> RhapsodyRuntimeException

callElementCollectionPluginMethod

```
public abstract boolean callElementCollectionPluginMethod(java.lang.String PluginClassName,
                                                          java.lang.String methodName,
                                                          IRPModelElement element,
                                                          IRPCollection collection)
```

Calls a method of a plugins main class with ModelElement and collection
Throws:
> RhapsodyRuntimeException

callPluginMethod

```
public abstract boolean callPluginMethod(java.lang.String PluginClassName,
                                         java.lang.String methodName,
                                         IRPCollection args)
```

Calls a method of a plugins main class
Throws:
> RhapsodyRuntimeException

callStringPluginMethod

```
public abstract java.lang.String callStringPluginMethod(java.lang.String PluginClassName,
                                                        java.lang.String methodName,
                                                        java.lang.String argument)
```

Calls a method of a plugins main class with string in/out
Throws:
> RhapsodyRuntimeException

getId

```
public abstract java.lang.String getId()
```

> Gets the id of the listener
> **Throws:**
> > RhapsodyRuntimeException

hasPluginWithMethod

```
public abstract boolean hasPluginWithMethod(java.lang.String PluginClassName,
                                            java.lang.String methodName)
```

> Check if a method exists on a plugin
> **Throws:**
> > RhapsodyRuntimeException

hasPluginWithMethodArgs

```
public abstract boolean hasPluginWithMethodArgs(java.lang.String PluginClassName,
                                                java.lang.String methodName,
                                                java.lang.String arguments)
```

> Check if a method with arguments exists on a plugin
> **Throws:**
> > RhapsodyRuntimeException

loadPlugin

```
public abstract boolean loadPlugin(java.lang.String PluginClassName,
                                   IRPCollection classURLS,
                                   IRPCollection libURLS)
```

> Loads the Java plugin main class
> **Throws:**
> > RhapsodyRuntimeException

unloadPlugin

```
public abstract boolean unloadPlugin(java.lang.String PluginClassName,
                                     int finalCall)
```

> Unload plugin
> **Throws:**
> > RhapsodyRuntimeException

com.telelogic.rhapsody.core
Class RPowPaneMgrEvents

```
java.lang.Object
   └─com.telelogic.rhapsody.core.RPowPaneMgrEvents
```

public abstract class **RPowPaneMgrEvents**
extends java.lang.Object

Constructor Summary

RPowPaneMgrEvents()	

Method Summary

abstract void	**addListElement**(java.lang.String sObjID, int nRow, int nCol, java.lang.String sText) method AddListElement
abstract void	**addPaneWnd**(int nType, int nSubType, java.lang.String sTitle) method AddPaneWnd
abstract void	**addTextContent**(java.lang.String sObjID, java.lang.String sContent) method AddTextContent
abstract void	**clearListContent**(java.lang.String sObjID) method ClearListContent
abstract void	**clearTextContent**(java.lang.String sObjID) method ClearTextContent
abstract void	**closePaneWnd**(java.lang.String sObjID) method ClosePaneWnd
boolean	**connect**(_IRPowPaneMgr_ connectionPoint)
boolean	**disconnect**()
protected void	**finalize**()
abstract void	

Method Summary

	setPaneWndTitle(java.lang.String sObjID, java.lang.String sTitle) method SetPaneWndTitle
abstract void	**showPaneWnd**(java.lang.String sObjID) method ShowPaneWnd

Methods inherited from class java.lang.Object

clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

RPowPaneMgrEvents

public **RPowPaneMgrEvents**()

Method Detail

connect

public boolean **connect**(IRPowPaneMgr connectionPoint)

disconnect

public boolean **disconnect**()

finalize

protected void **finalize**()
                throws java.lang.Throwable

> **Overrides:**
>   finalize in class java.lang.Object
> **Throws:**
>   java.lang.Throwable

addListElement

public abstract void **addListElement**(java.lang.String sObjID,
                                  int nRow,
                                  int nCol,

```
                              java.lang.String sText)
```

method AddListElement
Throws:
RhapsodyRuntimeException

addPaneWnd

```
public abstract void addPaneWnd(int nType,
                                int nSubType,
                                java.lang.String sTitle)
```

method AddPaneWnd
Throws:
RhapsodyRuntimeException

addTextContent

```
public abstract void addTextContent(java.lang.String sObjID,
                                    java.lang.String sContent)
```

method AddTextContent
Throws:
RhapsodyRuntimeException

clearListContent

```
public abstract void clearListContent(java.lang.String sObjID)
```

method ClearListContent
Throws:
RhapsodyRuntimeException

clearTextContent

```
public abstract void clearTextContent(java.lang.String sObjID)
```

method ClearTextContent
Throws:
RhapsodyRuntimeException

closePaneWnd

```
public abstract void closePaneWnd(java.lang.String sObjID)
```

method ClosePaneWnd

Throws:
[RhapsodyRuntimeException](#)

---

setPaneWndTitle

```
public abstract void setPaneWndTitle(java.lang.String sObjID,
                                     java.lang.String sTitle)
```

method SetPaneWndTitle
Throws:
[RhapsodyRuntimeException](#)

---

showPaneWnd

```
public abstract void showPaneWnd(java.lang.String sObjID)
```

method ShowPaneWnd
Throws:
[RhapsodyRuntimeException](#)

Package | **Class** | **Use** | **Tree** | **Serialized** | **Deprecated** | **Index** | **Help**

PREV CLASS **NEXT CLASS**   **FRAMES** **NO FRAMES**   **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

---

com.telelogic.rhapsody.core
Class RPRoundTripListener

```
java.lang.Object
  └─ com.telelogic.rhapsody.core.RPRoundTripListener
```

---

public abstract class **RPRoundTripListener**
extends java.lang.Object

Constructor Summary

RPRoundTripListener()

Method Summary

abstract void	**afterRoundTrip**(IRPCollection Items) Called after round trip was finished
abstract void	**beforeRoundTrip**(IRPCollection Items) Called before round trip is started
boolean	**connect**(IRPRoundTrip connectionPoint)
boolean	**disconnect**()
protected void	**finalize**()
abstract java.lang.String	**getId**() Gets the id of the listener

Methods inherited from class java.lang.Object
clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

---

showPaneWnd   798

RPRoundTripListener

```
public RPRoundTripListener()
```

<div style="background-color:#ccccee; border:1px solid black;">

Method Detail

</div>

connect

```
public boolean connect(IRPRoundTrip connectionPoint)
```

---

disconnect

```
public boolean disconnect()
```

---

finalize

```
protected void finalize()
                throws java.lang.Throwable
```

> **Overrides:**
> finalize in class java.lang.Object
> **Throws:**
> java.lang.Throwable

---

afterRoundTrip

```
public abstract void afterRoundTrip(IRPCollection Items)
```

> Called after round trip was finished
> **Throws:**
> RhapsodyRuntimeException

---

beforeRoundTrip

```
public abstract void beforeRoundTrip(IRPCollection Items)
```

> Called before round trip is started
> **Throws:**
> RhapsodyRuntimeException

---

getId

```
public abstract java.lang.String getId()
```

Gets the id of the listener
Throws:
RhapsodyRuntimeException

com.telelogic.rhapsody.core
Class RPRTCListener

```
java.lang.Object
   └─ com.telelogic.rhapsody.core.RPRTCListener
```

public abstract class **RPRTCListener**
extends java.lang.Object

Constructor Summary

RPRTCListener()	

Method Summary

abstract boolean	**afterSave**() Called after save is done in Rhapsody
abstract boolean	**afterUnitSave**(IRPUnit pUnit) Called after unit is saved in Rhapsody
boolean	**connect**(IRPApplication connectionPoint)
boolean	**disconnect**()
abstract java.lang.String	**executeCommand**(IRPUnit pUnit, java.lang.String command, java.lang.String parameters) Called on request to execute a Command
protected void	**finalize**()
abstract java.lang.String	**getFileUUID**(java.lang.String strUnits) Called on request to get unit's UUID
abstract java.lang.String	**getId**() Gets the id of the listener
abstract int	**onGetStatus**(IRPUnit pUnit) Called on request to get unit's status
abstract boolean	

Method Summary

	onLocateInPendingChanges([IRPUnit](#) pUnit) Called on request to locate unit in pending changes view
abstract boolean	**onLocateInRepositoryFiles**([IRPUnit](#) pUnit) Called on request to locate unit in repository files' view
abstract boolean	**onLock**([IRPUnit](#) pUnit) Called on request to lock unit
abstract boolean	**onRefreshStatus**() Called to refresh status cache in RTC
abstract boolean	**onShowHistory**([IRPUnit](#) pUnit) Called on request to show CM revisions' history of a unit
abstract boolean	**onUnlock**([IRPUnit](#) pUnit) Called on request to un-lock unit

Methods inherited from class java.lang.Object

clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

RPRTCListener

public **RPRTCListener**()

Method Detail

connect

public boolean **connect**([IRPApplication](#) connectionPoint)

---

disconnect

public boolean **disconnect**()

---

finalize

protected void **finalize**()
         throws java.lang.Throwable

Overrides:

RPRTCListener

```
        finalize in class java.lang.Object
Throws:
        java.lang.Throwable
```

executeCommand

```
public abstract java.lang.String executeCommand(IRPUnit pUnit,
                                                java.lang.String command,
                                                java.lang.String parameters)
```

Called on request to execute a Command
Throws:
     RhapsodyRuntimeException

afterSave

```
public abstract boolean afterSave()
```

Called after save is done in Rhapsody
Throws:
     RhapsodyRuntimeException

afterUnitSave

```
public abstract boolean afterUnitSave(IRPUnit pUnit)
```

Called after unit is saved in Rhapsody
Throws:
     RhapsodyRuntimeException

getFileUUID

```
public abstract java.lang.String getFileUUID(java.lang.String strUnits)
```

Called on request to get unit's UUID
Throws:
     RhapsodyRuntimeException

getId

```
public abstract java.lang.String getId()
```

Gets the id of the listener
Throws:
     RhapsodyRuntimeException

onGetStatus

```
public abstract int onGetStatus(IRPUnit pUnit)
```

>   Called on request to get unit's status
>   **Throws:**
>   RhapsodyRuntimeException

onLocateInPendingChanges

```
public abstract boolean onLocateInPendingChanges(IRPUnit pUnit)
```

>   Called on request to locate unit in pending changes view
>   **Throws:**
>   RhapsodyRuntimeException

onLocateInRepositoryFiles

```
public abstract boolean onLocateInRepositoryFiles(IRPUnit pUnit)
```

>   Called on request to locate unit in repository files' view
>   **Throws:**
>   RhapsodyRuntimeException

onLock

```
public abstract boolean onLock(IRPUnit pUnit)
```

>   Called on request to lock unit
>   **Throws:**
>   RhapsodyRuntimeException

onRefreshStatus

```
public abstract boolean onRefreshStatus()
```

>   Called to refresh status cache in RTC
>   **Throws:**
>   RhapsodyRuntimeException

onShowHistory

```
public abstract boolean onShowHistory(IRPUnit pUnit)
```

>   Called on request to show CM revisions' history of a unit
>   **Throws:**

onUnlock

```
public abstract boolean onUnlock(IRPUnit pUnit)
```

Called on request to un-lock unit

Throws:

RhapsodyRuntimeException

---

com.telelogic.rhapsody.core
Class RPSearchListener

```
java.lang.Object
  └─com.telelogic.rhapsody.core.RPSearchListener
```

public abstract class **RPSearchListener**
extends java.lang.Object

Constructor Summary

RPSearchListener()	

Method Summary

boolean	**connect**(IRPSearchManager connectionPoint)
boolean	**disconnect**()
protected void	**finalize**()
abstract boolean	**onNewSearchResult**(IRPSearchResult pSearchResult) Called during search
abstract void	**searchEnded**(IRPSearchQuery pSearchQuery) Called after search ends
abstract boolean	**searchStarted**(IRPSearchQuery pSearchQuery) Called before search starts

Methods inherited from class java.lang.Object

clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

---

Constructor Detail

RPSearchListener

```
public RPSearchListener()
```

Method Detail

connect

```
public boolean connect(IRPSearchManager connectionPoint)
```

disconnect

```
public boolean disconnect()
```

finalize

```
protected void finalize()
                throws java.lang.Throwable
```

> **Overrides:**
>   finalize in class java.lang.Object
> **Throws:**
>   java.lang.Throwable

onNewSearchResult

```
public abstract boolean onNewSearchResult(IRPSearchResult pSearchResult)
```

> Called during search
> **Throws:**
>   RhapsodyRuntimeException

searchEnded

```
public abstract void searchEnded(IRPSearchQuery pSearchQuery)
```

> Called after search ends
> **Throws:**
>   RhapsodyRuntimeException

searchStarted

```
public abstract boolean searchStarted(IRPSearchQuery pSearchQuery)
```

Called before search starts

Throws:

RhapsodyRuntimeException

Package	**Class**	**Use**	**Tree**	**Serialized**	**Deprecated**	**Index**	**Help**

PREV CLASS   NEXT CLASS   **FRAMES   NO FRAMES   All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

com.telelogic.rhapsody.core
Class RPUserPlugin

```
java.lang.Object
  └─com.telelogic.rhapsody.core.RPUserPlugin
```

public abstract class **RPUserPlugin**
extends java.lang.Object

Constructor Summary

RPUserPlugin()	

Method Summary

abstract void	**OnMenuItemSelect**(java.lang.String menuItem)
abstract void	**OnTrigger**(java.lang.String trigger)
abstract boolean	**RhpPluginCleanup**()
abstract void	**RhpPluginFinalCleanup**()
abstract void	**RhpPluginInit**(IRPApplication rpyApplication)
abstract void	**RhpPluginInvokeItem**()
void	**RhpPluginInvokeItem**(java.lang.String str)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

RPUserPlugin

```
public RPUserPlugin()
```

Method Detail

RhpPluginInit

```
public abstract void RhpPluginInit(IRPApplication rpyApplication)
```

RhpPluginInvokeItem

```
public abstract void RhpPluginInvokeItem()
```

RhpPluginInvokeItem

```
public void RhpPluginInvokeItem(java.lang.String str)
```

OnMenuItemSelect

```
public abstract void OnMenuItemSelect(java.lang.String menuItem)
```

OnTrigger

```
public abstract void OnTrigger(java.lang.String trigger)
```

RhpPluginCleanup

```
public abstract boolean RhpPluginCleanup()
```

RhpPluginFinalCleanup

```
public abstract void RhpPluginFinalCleanup()
```

[Package](#) **Class** **[Use](#)** **[Tree](#)** **[Serialized](#)** **[Deprecated](#)** **[Index](#)** **[Help](#)**

[PREV CLASS](#) [NEXT CLASS](#)   **[FRAMES](#) [NO FRAMES](#) [All Classes](#)**

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)   DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

RPUserPlugin   810

com.telelogic.rhapsody.core
Class SearchFindAsEnum

```
java.lang.Object
  └─ com.telelogic.rhapsody.core.SearchFindAsEnum
```

public class **SearchFindAsEnum**
extends java.lang.Object

Field Summary

static char	**RP_SEARCH_EMPTY_ONLY** search for empty string only
static char	**RP_SEARCH_EXACT** search for exact string
static char	**RP_SEARCH_REGEX** search as regular expression
static char	**RP_SEARCH_WILDCARD** search as wildcard

Constructor Summary

SearchFindAsEnum()

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

RP_SEARCH_EXACT

`public static final char` **`RP_SEARCH_EXACT`**

> search for exact string

> **See Also:**
> > [Constant Field Values](#)

RP_SEARCH_WILDCARD

`public static final char` **`RP_SEARCH_WILDCARD`**

> search as wildcard

> **See Also:**
> > [Constant Field Values](#)

RP_SEARCH_REGEX

`public static final char` **`RP_SEARCH_REGEX`**

> search as regular expression

> **See Also:**
> > [Constant Field Values](#)

RP_SEARCH_EMPTY_ONLY

`public static final char` **`RP_SEARCH_EMPTY_ONLY`**

> search for empty string only

> **See Also:**
> > [Constant Field Values](#)

Constructor Detail

SearchFindAsEnum

`public` **`SearchFindAsEnum`**`()`

---

com.telelogic.rhapsody.core

PREV CLASS NEXT CLASS   **FRAMES** **NO FRAMES** **All Classes**
SUMMARY: NESTED | FIELD | CONSTR | METHOD   DETAIL: FIELD | CONSTR | METHOD

A

abort() - Method in class com.telelogic.rhapsody.core.RPExternalCodeGenerator
    method Abort
activateView(IRPAXViewCtrl) - Method in class com.telelogic.rhapsody.core.RPExternalIDEManager
    Activate view
activeProject() - Method in interface com.telelogic.rhapsody.core.IRPApplication
    Returns an IRPProject object representing the project currently open in Rhapsody
activeProjectAboutToChange() - Method in interface com.telelogic.rhapsody.core.IRPInternalOEMPlugin
    Notify the Plugin upon ActiveProjectAboutToChange
activeProjectAboutToChange(IRPProject) - Method in class
com.telelogic.rhapsody.core.RPApplicationListener

activeProjectHasChanged() - Method in interface com.telelogic.rhapsody.core.IRPInternalOEMPlugin
    Notify the Plugin upon ActiveProjectHasChanged
activeProjectHasChanged(IRPProject) - Method in class
com.telelogic.rhapsody.core.RPApplicationListener

actualCreateRhapsodyApplicationDllServer() - Static method in class
com.telelogic.rhapsody.core.RhapsodyAppServer
    For internal use only.
actualCreateUninitializedRhapsodyApplicationDllServer() - Static method in class
com.telelogic.rhapsody.core.RhapsodyAppServer
    For internal use only.
actualInitializeRhapsodyApplicationDllServer(long) - Static method in class
com.telelogic.rhapsody.core.RhapsodyAppServer
    For internal use only.
addAcceptEventAction(String, IRPState) - Method in interface com.telelogic.rhapsody.core.IRPFlowchart
    Adds a new Accept Event Action element to the activity.
addAcceptTimeEvent(String, IRPState) - Method in interface com.telelogic.rhapsody.core.IRPFlowchart
    Adds a new Accept Time Event element to the activity.
addActionBlock(IRPClassifierRole) - Method in interface com.telelogic.rhapsody.core.IRPCollaboration
    Adds a new action block to the specified classifier.
addActivityDiagram() - Method in interface com.telelogic.rhapsody.core.IRPClassifier
    Creates a new activity diagram.
addActivityDiagram() - Method in interface com.telelogic.rhapsody.core.IRPPackage
    Adds a new activity diagram to the package.
addActivityFinal() - Method in interface com.telelogic.rhapsody.core.IRPState
    Adds an ActivityFinal element to an Activity.
addActivityParameter(String) - Method in interface com.telelogic.rhapsody.core.IRPFlowchart
    Adds an activity parameter to the frame of the activity
addActor(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
    Adds a new actor to the package.

addAnchor(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPAnnotation
>Adds an anchor from the annotation to the specified model element.

addAndLine(IRPGraphNode, int, int, int, int) - Method in interface
com.telelogic.rhapsody.core.IRPStatechartDiagram
>Adds an And Line to the specified state.

addArgument(String) - Method in interface com.telelogic.rhapsody.core.IRPInterfaceItem
>Adds a new argument to the end of the argument list.

addArgumentBeforePosition(String, int) - Method in interface
com.telelogic.rhapsody.core.IRPInterfaceItem
>Adds a new argument at the specified position in the argument list.

addArgumentValue(String, int) - Method in interface com.telelogic.rhapsody.core.IRPSendAction
>Provides an argument value for an argument of the event associated with the Send Action element.

addAssociation(IRPRelation, IRPRelation, String) - Method in interface
com.telelogic.rhapsody.core.IRPModelElement
>Creates an association class using the specified IRPRelation elements.

addAttribute(String) - Method in interface com.telelogic.rhapsody.core.IRPClassifier
>Adds a new attribute to the classifier.

addCallBehavior(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPFlowchart
>Adds a new Call Behavior element to the activity.

addCallOperation(String, IRPState) - Method in interface com.telelogic.rhapsody.core.IRPFlowchart
>Adds a new Call Operation element to the activity.

addCancelledTimeout(IRPClassifierRole) - Method in interface
com.telelogic.rhapsody.core.IRPCollaboration
>Adds a cancelled timeout to the specified instance line.

addClass(String) - Method in interface com.telelogic.rhapsody.core.IRPClass
>Adds a class to the current class.

addClass(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
>Adds a new class to the package.

addClassifierRole(String, IRPClassifier) - Method in interface
com.telelogic.rhapsody.core.IRPCollaboration
>Adds an instance line to a sequence diagram.

addClassifierRoleByName(String, String) - Method in interface
com.telelogic.rhapsody.core.IRPCollaboration
>method addClassifierRoleByName

addClassifierRoleForInstance(IRPInstance) - Method in interface
com.telelogic.rhapsody.core.IRPCollaboration
>method addClassifierRoleForInstance

addCollaborationDiagram(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
>Adds a new collaboration diagram to the package.

addColumn(String, String, String) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout
>Adds a new column to the table layout.

addColumnEx(String, String, String, String) - Method in interface
com.telelogic.rhapsody.core.IRPTableLayout
>Adds a new column to the table layout.

addComponent(String) - Method in interface com.telelogic.rhapsody.core.IRPProject
>Adds a new Component to the project.

addComponentDiagram(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
>Adds a new component diagram to the package.

addComponentInstance(String) - Method in interface com.telelogic.rhapsody.core.IRPNode
>method addComponentInstance

A

addConditionMark(IRPClassifierRole) - Method in interface
com.telelogic.rhapsody.core.IRPCollaboration
> Adds a condition mark to the specified instance line.

addConfiguration(String) - Method in interface com.telelogic.rhapsody.core.IRPComponent
> Adds a new configuration to the component.

addConnector(String) - Method in interface com.telelogic.rhapsody.core.IRPState
> Adds a connector element of the specified type to the state.

addConstructor(String) - Method in interface com.telelogic.rhapsody.core.IRPClass
> Adds a constructor for the current class.

addConveyed(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPFlow
> method addConveyed

addCtor(IRPInterfaceItem, String, IRPClassifierRole, IRPClassifierRole) - Method in interface
com.telelogic.rhapsody.core.IRPCollaboration
> Adds a Create Arrow to a sequence diagram.

addCustomViewOnBrowser(IRPPackage) - Method in interface com.telelogic.rhapsody.core.IRPProject
> Applies the specified custom view to the model browser.

addCustomViewOnDiagram(IRPDiagram, IRPPackage) - Method in interface
com.telelogic.rhapsody.core.IRPProject
> Applies the specified custom view to the specified diagram.

addDataFlow(IRPSysMLPort, String, IRPClassifierRole, IRPClassifierRole) - Method in interface
com.telelogic.rhapsody.core.IRPCollaboration
> method addDataFlow

addDependency(String, String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Adds a dependency from the model element to the model element specified by the parameters.

addDependencyBetween(IRPModelElement, IRPModelElement) - Method in interface
com.telelogic.rhapsody.core.IRPModelElement
> Creates a dependency between the two specified elements.

addDependencyTo(IRPModelElement) - Method in interface
com.telelogic.rhapsody.core.IRPModelElement
> Adds a dependency upon another model element.

addDeploymentDiagram(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
> Adds a new deployment diagram to the package.

addDescribingDiagram(IRPDiagram) - Method in interface com.telelogic.rhapsody.core.IRPUseCase
> method addDescribingDiagram

addDestructionEvent(IRPClassifierRole) - Method in interface
com.telelogic.rhapsody.core.IRPCollaboration
> Adds a destruction event to the specified lifeline.

addDestructor() - Method in interface com.telelogic.rhapsody.core.IRPClass
> Adds a destructor for the current class.

addDiagramToViewsList(IRPDiagram) - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
> Adds the specified diagram to the list of views to be searched for the search text.

addDtor(IRPInterfaceItem, String, IRPClassifierRole, IRPClassifierRole) - Method in interface
com.telelogic.rhapsody.core.IRPCollaboration
> Adds a Destroy Arrow to a sequence diagram.

addDurationConstraint(String, IRPMessage, IRPMessage) - Method in interface
com.telelogic.rhapsody.core.IRPCollaboration
> Adds a Duration Constraint to the specified state invariants.

addDurationObservation(String, IRPMessage, IRPMessage) - Method in interface
com.telelogic.rhapsody.core.IRPCollaboration
> Adds a Duration Observation to the specified states invariants.

addElement(IRPClassifier, String) - Method in interface com.telelogic.rhapsody.core.IRPFile

method addElement Choose from = undefFragment, textFragment, implFragment, specFragment, moduleFragment

addElementDefaultValue(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPVariable

For tags with multiplicity greater than 1, this method can be used to add a model element as an additional value.

addElementValue(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPInstanceSlot

method addElementValue

addEnumerationLiteral(String) - Method in interface com.telelogic.rhapsody.core.IRPType

method addEnumerationLiteral

addEvent(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage

Adds a new event to the package.

addEventReception(String) - Method in interface com.telelogic.rhapsody.core.IRPClass

Adds an event reception to the current class.

addExtensionPoint(String) - Method in interface com.telelogic.rhapsody.core.IRPUseCase

method addExtensionPoint

addFile(String) - Method in interface com.telelogic.rhapsody.core.IRPComponent

Adds a new File to the component.

addFilterElementType(String) - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Adds an element type to the list of element types that the search should be applied to.

addFilterSearchInField(String) - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Adds an element field to the list of element fields that the search should be applied to, for example, element name or element description.

addFilterStereotype(IRPStereotype) - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Specifies that the search should be limited to model elements with a specific stereotype applied to them.

addFilterSubQuery(IRPTableLayout, int) - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Adds a subquery to the list of subqueries specified for the search.

addFlow(String, IRPStateVertex) - Method in interface com.telelogic.rhapsody.core.IRPStateVertex

Adds a control flow or object flow from this element to the specified element.

addFlowItems(String) - Method in interface com.telelogic.rhapsody.core.IRPClassifier

Adds a new item flow to the classifier.

addFlowItems(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage

Adds an item flow to the package.

addFlows(String) - Method in interface com.telelogic.rhapsody.core.IRPClassifier

Adds a new flow to the classifier.

addFlows(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage

Adds a flow to the package.

addFolder(String) - Method in interface com.telelogic.rhapsody.core.IRPComponent

Adds a new Folder to the component.

addFoundMessage(IRPClassifierRole) - Method in interface com.telelogic.rhapsody.core.IRPCollaboration

Adds a Found Message to the specified lifeline.

addFreeShapeByType(String, IRPCollection, IRPCollection) - Method in interface com.telelogic.rhapsody.core.IRPDiagram

Adds a free shape of the type specified, using the x coordinates and y coordinates provided.

addFreeShapeByType(String, IRPCollection, IRPCollection) - Method in interface com.telelogic.rhapsody.core.IRPStatechart

Adds a free shape of the type specified, using the x coordinates and y coordinates provided.

addGeneralization(IRPClassifier) - Method in interface com.telelogic.rhapsody.core.IRPClassifier

Adds a generalization relationship between the classifier and the classifier specified as a parameter.

addGlobalFunction(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Adds a global function to the package.

addGlobalObject(String, String, String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Adds an Object to the package.

addGlobalVariable(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Adds a global variable to the package.

addGraphicalItem(IRPGraphElement) - Method in interface com.telelogic.rhapsody.core.IRPCollection
>   Adds a graphical element to a collection.

addImage(String, int, int, int, int) - Method in interface com.telelogic.rhapsody.core.IRPDiagram
>   Adds an image to the diagram, using the specified file, starting point, width, and height.

addImage(String, int, int, int, int) - Method in interface com.telelogic.rhapsody.core.IRPStatechart
>   Adds an image to the statechart, using the specified file, starting point, width, and height.

addImplicitObject(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Adds an implicit object to the package.

addInitialInstance(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPConfiguration
>   method addInitialInstance

addInstance(long, String) - Method in interface com.telelogic.rhapsody.core.IRPDiagSynthAPI
>   DiagSynthAPI : add instance to sequence diagram

addInState(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPObjectNode
>   Adds the specified state to the list of "In State" states for the object node.

addInteractionOccurrence() - Method in interface com.telelogic.rhapsody.core.IRPCollaboration
>   Adds an interaction occurrence.

addInteractionOperator() - Method in interface com.telelogic.rhapsody.core.IRPCollaboration
>   Adds an interaction operator to a sequence diagram.

addInternalTransition(IRPInterfaceItem) - Method in interface com.telelogic.rhapsody.core.IRPState
>   method addInternalTransition

addItem(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPCollection
>   Adds a model element to a collection.

addLink(IRPInstance, IRPInstance, IRPRelation, IRPPort, IRPPort) - Method in interface com.telelogic.rhapsody.core.IRPClass
>   This method is used to create a link between two parts belonging to a class.

addLink(IRPInstance, IRPInstance, IRPRelation, IRPPort, IRPPort) - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Creates a link between two objects in the package.

addLink(IRPInstance, IRPInstance, IRPRelation, IRPSysMLPort, IRPPackage) - Method in interface com.telelogic.rhapsody.core.IRPSysMLPort
>   This method is used to create a link between flowports on two parts.

addLinkBetweenSYSMLPorts(IRPInstance, IRPInstance, IRPRelation, IRPSysMLPort, IRPSysMLPort) - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Creates a link between two objects.

addLinkToElement(IRPModelElement, IRPRelation, IRPModelElement, IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
>   Creates a link between this model element and the model element specified as an argument.

addLinkToPartViaPort(IRPInstance, IRPInstance, IRPInstance, IRPRelation) - Method in interface com.telelogic.rhapsody.core.IRPClass
>   This method is used to create a delegation connector between a class and one of its parts.

addListElement(String, int, int, String) - Method in class com.telelogic.rhapsody.core.RPowPaneMgrEvents
>   method AddListElement

addLostMessage(IRPClassifierRole) - Method in interface com.telelogic.rhapsody.core.IRPCollaboration
>   Adds a Lost Message to the specified lifeline.

A

addMatrixToViewsList(IRPMatrixView) - Method in interface
com.telelogic.rhapsody.core.IRPSearchQuery
>   Adds the specified matrix to the list of views to be searched for the search text.
addMessage(IRPInterfaceItem, String, IRPClassifierRole, IRPClassifierRole) - Method in interface
com.telelogic.rhapsody.core.IRPCollaboration
>   Adds a message to a sequence diagram.
addMetaClass(String) - Method in interface com.telelogic.rhapsody.core.IRPStereotype
>   Adds a metaclass to the list of metaclasses that the stereotype can be applied to.
addModelElement(IRPModelElement, String) - Method in interface com.telelogic.rhapsody.core.IRPFile
>   method addModelElement Choose from = undefFragment, textFragment, implFragment,
>   specFragment, moduleFragment
addModule(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Adds a new File element to the package.
addNestedComponent(String) - Method in interface com.telelogic.rhapsody.core.IRPComponent
>   Adds a new nested component to the component.
addNestedPackage(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Adds a nested package to the package.
addNewAggr(String, String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
>   Adds a new model element to the current element, for example, adding a class to a package.
addNewEdgeByType(String, IRPGraphElement, int, int, IRPGraphElement, int, int) - Method in
interface com.telelogic.rhapsody.core.IRPDiagram
>   Adds a connector element of the specified type to the diagram, using the source and target elements
>   specified.
addNewEdgeByType(String, IRPGraphElement, int, int, IRPGraphElement, int, int) - Method in
interface com.telelogic.rhapsody.core.IRPStatechart
>   Adds a connector element of the specified type to the statechart, using the source and target elements
>   specified.
addNewEdgeForElement(IRPModelElement, IRPGraphNode, int, int, IRPGraphNode, int, int) -
Method in interface com.telelogic.rhapsody.core.IRPDiagram
>   Adds a connector graphical element to the diagram to represent the specified model element.
addNewEdgeForElement(IRPModelElement, IRPGraphNode, int, int, IRPGraphNode, int, int) -
Method in interface com.telelogic.rhapsody.core.IRPStatechart
>   Adds a connector graphical element to the statechart to represent the specified model element.
addNewNodeByType(String, int, int, int, int) - Method in interface
com.telelogic.rhapsody.core.IRPDiagram
>   Adds a diagram element of the specified type to the diagram, using the position and dimensions
>   specified.
addNewNodeByType(String, int, int, int, int) - Method in interface
com.telelogic.rhapsody.core.IRPStatechart
>   Adds a statechart element of the specified type to the statechart, using the position and dimensions
>   specified.
addNewNodeForElement(IRPModelElement, int, int, int, int) - Method in interface
com.telelogic.rhapsody.core.IRPDiagram
>   Adds a graphical element to the diagram to represent the specified model element.
addNewNodeForElement(IRPModelElement, int, int, int, int) - Method in interface
com.telelogic.rhapsody.core.IRPStatechart
>   Adds a graphical element to the statechart to represent the specified model element.
addNode(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Adds a Node element to the package.
addObjectModelDiagram(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Adds a new object model diagram to the package.

A   820

addObjectNode(String, IRPState) - Method in interface com.telelogic.rhapsody.core.<u>IRPFlowchart</u>
> Adds a new Object Node element to the activity.

addOperation(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPClassifier</u>
> Adds a new operation.

addPackage(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPProject</u>
> Adds a new package to the project.

addPackageToInstrumentationScope(IRPPackage) - Method in interface com.telelogic.rhapsody.core.<u>IRPConfiguration</u>
> method addPackageToInstrumentationScope

addPackageToScope(IRPPackage) - Method in interface com.telelogic.rhapsody.core.<u>IRPFile</u>
> method addPackageToScope

addPanelDiagram(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPPackage</u>
> Adds a new panel diagram to the package.

addPaneWnd(int, int, String) - Method in class com.telelogic.rhapsody.core.<u>RPowPaneMgrEvents</u>
> method AddPaneWnd

addProfile(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPProject</u>
> Adds a new profile to the project.

addProfileToModel(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPApplication</u>
> addProfileToModel

addProperty(String, String, String) - Method in interface com.telelogic.rhapsody.core.<u>IRPGraphElement</u>
> method addProperty

addProperty(String, String, String) - Method in interface com.telelogic.rhapsody.core.<u>IRPModelElement</u>
> Adds a new property to the model element and assigns a value to it.

addProvidedInterface(IRPClass) - Method in interface com.telelogic.rhapsody.core.<u>IRPPort</u>
> method addRProvidedInterface

addQualifier(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.<u>IRPRelation</u>
> method addQualifier

addReception(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPClass</u>
> Adds a reception to the current class.

addRedefines(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.<u>IRPModelElement</u>
> method addRedefines

addReferenceActivity(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.<u>IRPFlowchart</u>
> Adds a new Call Behavior element to the activity.

addRelation(String, String, String, String, String, String, String, String, String) - Method in interface com.telelogic.rhapsody.core.<u>IRPClassifier</u>
> Adds a new association to the classifier.

addRelationTo(IRPClassifier, String, String, String, String, String, String, String) - Method in interface com.telelogic.rhapsody.core.<u>IRPClassifier</u>
> Adds a new association to the classifier.

addRelationToTheWhole(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPInstance</u>
> method addRelationToTheWhole

addRemoteDependencyTo(IRPModelElement, String) - Method in interface com.telelogic.rhapsody.core.<u>IRPModelElement</u>
> For Design Manager projects, used to create a dependency from a model element to a remote element.

addReplyMessage(IRPInterfaceItem, String, IRPClassifierRole, IRPClassifierRole) - Method in interface com.telelogic.rhapsody.core.<u>IRPCollaboration</u>
> method addReplyMessage

addRepresented(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.<u>IRPFlowItem</u>
> Adds an element to the collection of information elements that are represented by the item flow.

addRequiredInterface(IRPClass) - Method in interface com.telelogic.rhapsody.core.<u>IRPPort</u>
> method addRequiredInterface

A   821

addScopeElement(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPComponent
> Adds the specified model element to the scope of the component.

addScopeElementWithoutAggregates(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPComponent
> method addScopeElementWithoutAggregates

addSearchScope(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
> Adds an element to the scope for the search.

addSelectedToFavorites() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> Adds the currently selected item to the Favorites list.

addSequenceDiagram(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
> Adds a new sequence diagram to the package.

addSourceExecutionOccurrence() - Method in interface com.telelogic.rhapsody.core.IRPMessage
> method addSourceExecutionOccurrence

addSpecificStereotype(IRPStereotype) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Applies the specified stereotype to the model element.

addSpellCheckerResult(String) - Method in interface com.telelogic.rhapsody.core.IRPProject
> For internal use only.

addState(String) - Method in interface com.telelogic.rhapsody.core.IRPState
> Adds a new substate to this state.

addStatechart() - Method in interface com.telelogic.rhapsody.core.IRPClassifier
> Creates a new statechart.

addStatechart() - Method in interface com.telelogic.rhapsody.core.IRPPackage
> Adds a new statechart to the package.

addStateInvariant(String, IRPClassifierRole) - Method in interface com.telelogic.rhapsody.core.IRPCollaboration
> Adds a State Invariant to the specified lifeline.

addStaticReaction(IRPInterfaceItem) - Method in interface com.telelogic.rhapsody.core.IRPState
> Adds an internal transition to the state.

addStereotype(String, String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Applies the specified stereotype to the model element if the project contains a stereotype with the name specified and applicable to the metaclass specified.

addStringDefaultValue(String) - Method in interface com.telelogic.rhapsody.core.IRPVariable
> For tags with multiplicity greater than 1, this method can be used to add a string as an additional value.

addStringValue(String) - Method in interface com.telelogic.rhapsody.core.IRPInstanceSlot
> method addStringValue

addSuperclass(IRPClass) - Method in interface com.telelogic.rhapsody.core.IRPClass
> Specifies a base class that the current class should inherit from.

addSwimlane(String) - Method in interface com.telelogic.rhapsody.core.IRPFlowchart
> Adds a new swimlane to the activity.

addSwimlane(String) - Method in interface com.telelogic.rhapsody.core.IRPSwimlane
> method addSwimlane

addSynthSDToModel2(IRPSequenceDiagram, long, int) - Method in interface com.telelogic.rhapsody.core.IRPDiagSynthAPI
> DiagSynthAPI : add synth sequence diagarm to model

addSystemBorder() - Method in interface com.telelogic.rhapsody.core.IRPCollaboration
> Adds a System Border element to a sequence diagram.

addTableToViewsList(IRPTableView) - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
> Adds the specified table to the list of views to be searched for the search text.

addTabNotify(int, int, String, String) - Method in interface com.telelogic.rhapsody.core.IRPPowPaneMgr

A

method AddTabNotify

addTargetExecutionOccurrence() - Method in interface com.telelogic.rhapsody.core.IRPMessage

      method addTargetExecutionOccurrence

addTerminationState() - Method in interface com.telelogic.rhapsody.core.IRPState

      Adds a termination state to a statechart.

addTextBox(String, int, int, int, int) - Method in interface com.telelogic.rhapsody.core.IRPDiagram

      Adds a text box using the specified text, starting point, width, and height.

addTextBox(String, int, int, int, int) - Method in interface com.telelogic.rhapsody.core.IRPStatechart

      Adds a text box using the specified text, starting point, width, and height.

addTextContent(String, String) - Method in class com.telelogic.rhapsody.core.RPowPaneMgrEvents

      method AddTextContent

addTextElement(String) - Method in interface com.telelogic.rhapsody.core.IRPFile

      method addTextElement

addTimeConstraint(String, IRPMessage) - Method in interface com.telelogic.rhapsody.core.IRPCollaboration

      Adds a Time Constraint to the specified state invariant.

addTimeInterval(IRPClassifierRole) - Method in interface com.telelogic.rhapsody.core.IRPCollaboration

      Adds a Time Interval to the specified lifeline.

addTimeObservation(String, IRPMessage) - Method in interface com.telelogic.rhapsody.core.IRPCollaboration

      Adds a Time Observation to the specified state invariant.

addTimeout(IRPInterfaceItem, String, IRPClassifierRole, IRPClassifierRole) - Method in interface com.telelogic.rhapsody.core.IRPCollaboration

      Adds a timeout to a sequence diagram.

addTimingDiagram(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage

      Adds a new timing diagram to the package.

addToClassPath(String) - Static method in class com.telelogic.rhapsody.core.RhapsodyAppServer

      Adds one or more classes to the classpath.

addToClassPath(String) - Static method in class com.telelogic.rhapsody.core.RhpUtils

      Add class path to the system's class path

addToInstrumentationScope(IRPClassifier) - Method in interface com.telelogic.rhapsody.core.IRPConfiguration

      method addToInstrumentationScope

addToLibPath(String) - Static method in class com.telelogic.rhapsody.core.RhapsodyAppServer

      Adds a directory to the libpath.

addToLibPath(String) - Static method in class com.telelogic.rhapsody.core.RhpUtils

addToModel(String, int) - Method in interface com.telelogic.rhapsody.core.IRPApplication

      add To Model

addToModelByReference(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication

      add To Model by reference

addToModelEx(String, int, int, int) - Method in interface com.telelogic.rhapsody.core.IRPApplication

      Adds a unit to the model.

addToModelFromURL(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication

      add To Model From URL

addToScope(IRPFile, IRPCollection, IRPCollection) - Method in interface com.telelogic.rhapsody.core.IRPComponent

      method addToScope

addToScope(IRPClassifier) - Method in interface com.telelogic.rhapsody.core.IRPFile

      method addToScope

addTransition(IRPStateVertex) - Method in interface com.telelogic.rhapsody.core.IRPStateVertex

Adds a transition from this element to the specified element.

addTriggeredOperation(String) - Method in interface com.telelogic.rhapsody.core.IRPClass
Adds a new triggered operation to the current class.

addType(String) - Method in interface com.telelogic.rhapsody.core.IRPClass
Adds a new type to the current class.

addType(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
Adds a new type to the package.

addUnidirectionalRelation(String, String, String, String, String, String) - Method in interface
com.telelogic.rhapsody.core.IRPClassifier
Adds a new directed association to the classifier.

addUnidirectionalRelationTo(IRPClassifier, String, String, String, String) - Method in interface
com.telelogic.rhapsody.core.IRPClassifier
Adds a new directed association to the classifier.

addUseCase(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
Adds a new use case to the package.

addUseCaseDiagram(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
Adds a new use case diagram to the package.

advanceCodeGenProgressBar() - Method in interface
com.telelogic.rhapsody.core.IRPBaseExternalCodeGeneratorTool
method advanceCodeGenProgressBar

afterAddElement(IRPModelElement) - Method in class
com.telelogic.rhapsody.core.RPApplicationListener
Called after element is added

afterApplicationClosed() - Method in class com.telelogic.rhapsody.core.RPApplicationListener

afterImportModel(IRPModelElement) - Method in class com.telelogic.rhapsody.core.RPIntegratorListener
Called after import model is completed

afterProjectClose(String) - Method in class com.telelogic.rhapsody.core.RPApplicationListener
Called after project is closed

afterProjectOpen(IRPProject) - Method in class com.telelogic.rhapsody.core.RPApplicationListener

afterProjectSaved(IRPProject) - Method in class com.telelogic.rhapsody.core.RPApplicationListener

afterRoundTrip(IRPCollection) - Method in class com.telelogic.rhapsody.core.RPRoundTripListener
Called after round trip was finished

afterSave() - Method in class com.telelogic.rhapsody.core.RPRTCListener
Called after save is done in Rhapsody

afterUnitSave(IRPUnit) - Method in class com.telelogic.rhapsody.core.RPRTCListener
Called after unit is saved in Rhapsody

AGGREGATE - Static variable in class
com.telelogic.rhapsody.core.IRPSearchQuery.References.RelationKind

ALL - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.ViewsToSearch

allElementsInScope() - Method in interface com.telelogic.rhapsody.core.IRPComponent
Adds all the elements in the model to the scope of the component.

allowAutoSave(int) - Method in interface com.telelogic.rhapsody.core.IRPProject
Can be used to temporarily disable autosaving of the model regardless of the current value of the
property General::Model::AutoSaveInterval.

allowBrowserRefresh(int) - Method in interface com.telelogic.rhapsody.core.IRPApplication
allowBrowserRefresh

A

allowGERefresh(int) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> allowGERefresh

allowNonUniqueNames(int) - Method in interface com.telelogic.rhapsody.core.IRPProject
> For internal use only.

AND - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.SubQueriesOperator

ANNOTATION_ATTRIBUTE - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column
> Value used for Type parameter of addColumn method

appendFailedElementsComments(String) - Method in interface com.telelogic.rhapsody.core.IRPExternalCheckRegistry
> method appendFailedElementsComments

applyBrowserCustomViewsOnDiagrams(int) - Method in interface com.telelogic.rhapsody.core.IRPProject
> Applies the custom views applied to the browser to all diagrams as well.

applyDefaultFormat() - Method in interface com.telelogic.rhapsody.core.IRPGraphElement
> method applyDefaultFormat

applyNewTermsProfile(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> Called to apply a NewTerms Profile to the active project

applyRoundtripDiffMerge(int, IRPProject, IRPCollection) - Method in interface com.telelogic.rhapsody.core.IRPProject
> For internal use only.

arcCheckOut(String, String, int, int) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> archive Check Out

AS_REFERENCE - Static variable in class com.telelogic.rhapsody.core.IRPApplication.AddToModel_Mode
> A reference to the unit should be added to the model (unit cannot be modified).

AS_UNIT_WITH_COPY - Static variable in class com.telelogic.rhapsody.core.IRPApplication.AddToModel_Mode
> The unit should be added to the model and its file should be copied to the project directory.

AS_UNIT_WITHOUT_COPY - Static variable in class com.telelogic.rhapsody.core.IRPApplication.AddToModel_Mode
> The unit should be added to the model as an editable unit, but its file should not be copied to the project directory.

attachToIRPModelElement(long) - Static method in class com.telelogic.rhapsody.core.RhapsodyAppServer
> For internal use only.

attachToRhapsodyApplication(long) - Static method in class com.telelogic.rhapsody.core.RhapsodyAppServer
> For internal use only.

B

becomeActiveProject() - Method in interface com.telelogic.rhapsody.core.IRPProject
> Makes this project the active project in Rhapsody.

becomeTemplateInstantiationOf(IRPModelElement) - Method in interface
com.telelogic.rhapsody.core.IRPModelElement
> Makes the current model element a template instantiation of the specified template.

beforeApplicationClosed() - Method in class com.telelogic.rhapsody.core.RPApplicationListener

beforeProjectClose(IRPProject) - Method in class com.telelogic.rhapsody.core.RPApplicationListener
> Called before project is closed

beforeProjectOpen(String) - Method in class com.telelogic.rhapsody.core.RPApplicationListener

beforeProjectSaved(IRPProject) - Method in class com.telelogic.rhapsody.core.RPApplicationListener

beforeRoundTrip(IRPCollection) - Method in class com.telelogic.rhapsody.core.RPRoundTripListener
> Called before round trip is started

beginSimplification() - Method in class com.telelogic.rhapsody.core.RPCodeGenSimplifier
> before all simplifications

bringToFront() - Method in interface com.telelogic.rhapsody.core.IRPGraphNode
> method bringToFront

bringWindowToTop() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> bring window to top

build() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> Builds an application using the active component and configuration.

buildEntireProject() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> buildEntireProject

buildWithDependencies() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> buildWithDependencies

C

call2StringPluginMethod(String, String, String, String) - Method in class
com.telelogic.rhapsody.core.RPJavaPluginsManager
> Calls a method of a plugins main class with two strings arguments

callElementCollectionPluginMethod(String, String, IRPModelElement, IRPCollection) - Method in class
com.telelogic.rhapsody.core.RPJavaPluginsManager
> Calls a method of a plugins main class with ModelElement and collection

callPluginMethod(String, String, IRPCollection) - Method in class
com.telelogic.rhapsody.core.RPJavaPluginsManager
> Calls a method of a plugins main class

callStringPluginMethod(String, String, String) - Method in class
com.telelogic.rhapsody.core.RPJavaPluginsManager
> Calls a method of a plugins main class with string in/out

canCopy() - Method in interface com.telelogic.rhapsody.core.IRPSelection
> Check if COPY operation is available for current selection

canCut() - Method in interface com.telelogic.rhapsody.core.IRPSelection
> Check if CUT operation is available for current selection

canDelete() - Method in interface com.telelogic.rhapsody.core.IRPSelection
> Check if DELETE operation is available for current selection

canPaste() - Method in interface com.telelogic.rhapsody.core.IRPSelection
> Check if PASTE operation is available for current selection

canRedo() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> Check if Redo action is available

canUndo() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> Check if Undo action is available

changeTo(String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Changes the model element to the type of element specified by the parameter provided.

check(IRPModelElement, IRPCollection) - Method in class com.telelogic.rhapsody.core.RPExternalCheck
> Called by Rhapsody to execute the check.

checkEventsBaseIdsSolveCollisions() - Method in interface com.telelogic.rhapsody.core.IRPProject
> Checks the values of the event base IDs for all packages in the model, detects collisions between the
> IDs, and resolves any incorrect values and collisions.

checkIn(String, String, int, int, String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> check In

checkModel() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> check model

checkOut(String, String, int, int) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> check Out

CLASSIFIER - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.GeneralAttribute
> Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for
> the Type parameter of addColumn method.

clean() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> clean

cleanUnresolvedElements(IRPModelElement) - Method in interface
com.telelogic.rhapsody.core.IRPProject
> Removes any unresolved elements from the model, starting at the level of the specified element and working downward.

clearListContent(String) - Method in class com.telelogic.rhapsody.core.RPowPaneMgrEvents
> method ClearListContent

clearOutputWindow(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> clear output window

clearTextContent(String) - Method in class com.telelogic.rhapsody.core.RPowPaneMgrEvents
> method ClearTextContent

clone(String, IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Clones a model element.

close() - Method in interface com.telelogic.rhapsody.core.IRPASCIIFile
> close file

close() - Method in interface com.telelogic.rhapsody.core.IRPProject
> Closes the project.

closeAllAnimatedSequenceDiagrams(int) - Method in interface
com.telelogic.rhapsody.core.IRPApplication
> Close All Animated Sequence diagrams without save

closeCSVFile(String) - Method in interface com.telelogic.rhapsody.core.IRPProject
> Closes the tab in the Output window for the specified csv file.

closeDiagram() - Method in interface com.telelogic.rhapsody.core.IRPDiagram
> Closes the diagram.

closeDiagram() - Method in interface com.telelogic.rhapsody.core.IRPStatechart
> Closes the statechart.

closeDiagram(IRPDiagram) - Method in class com.telelogic.rhapsody.core.RPExternalIDEManager
> Closes diagram if opened

closePaneWnd(String) - Method in class com.telelogic.rhapsody.core.RPowPaneMgrEvents
> method ClosePaneWnd

CloseSession() - Static method in class com.telelogic.rhapsody.core.RhapsodyAppServer
> For internal use only.

CloseSessionNative() - Static method in class com.telelogic.rhapsody.core.RhapsodyAppServer
> For internal use only.

closeTabNotify(String) - Method in interface com.telelogic.rhapsody.core.IRPowPaneMgr
> method CloseTabNotify

com.telelogic.rhapsody.core - package com.telelogic.rhapsody.core

COMMENT_SPECIFICATION - Static variable in class
com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

compareSequenceDiagram(IRPSequenceDiagram, IRPSequenceDiagram) - Method in interface
com.telelogic.rhapsody.core.IRPApplication
> Compares the two sequence diagrams specified as parameters.

COMPARTMENT - Static variable in class com.telelogic.rhapsody.core.IRPGraphElement.ImageLayout
> Set image layout in a compartment

completeRelations(IRPCollection, int) - Method in interface com.telelogic.rhapsody.core.IRPDiagram
> Adds connectors to the diagram to reflect the existing relations between the specified elements.

CONFIGURATION_INITIALIZATION - Static variable in class
com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

C

connect(IRPApplication) - Method in class com.telelogic.rhapsody.core.RPApplicationListener

connect(IRPCodeGenerator) - Method in class com.telelogic.rhapsody.core.RPCodeGeneratorListener

connect(IRPCodeGenSimplifiersRegistry) - Method in class com.telelogic.rhapsody.core.RPCodeGenSimplifier

connect(IRPExternalCheckRegistry) - Method in class com.telelogic.rhapsody.core.RPExternalCheck

connect(IRPExternalCodeGeneratorInvoker) - Method in class com.telelogic.rhapsody.core.RPExternalCodeGenerator

connect(IRPExternalIDERegistry) - Method in class com.telelogic.rhapsody.core.RPExternalIDEManager

connect(IRPExternalRoundtripInvoker) - Method in class com.telelogic.rhapsody.core.RPExternalRoundtrip

connect(IRPIntegrator) - Method in class com.telelogic.rhapsody.core.RPIntegratorListener

connect(IRPJavaPlugins) - Method in class com.telelogic.rhapsody.core.RPJavaPluginsManager

connect(IRPPowPaneMgr) - Method in class com.telelogic.rhapsody.core.RPPowPaneMgrEvents

connect(IRPRoundTrip) - Method in class com.telelogic.rhapsody.core.RPRoundTripListener

connect(IRPApplication) - Method in class com.telelogic.rhapsody.core.RPRTCListener

connect(IRPSearchManager) - Method in class com.telelogic.rhapsody.core.RPSearchListener

connectToArchive(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
       connect To Archive
connectToImportedModel(String, String, String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
       Used to connect to an externally-managed model.
connectToTarget(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
       connectToTarget
CONSTRAINT_SPECIFICATION - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

copySelected() - Method in interface com.telelogic.rhapsody.core.IRPSelection
       Perform COPY operation under current selection
copyToAnotherProject(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPUnit
       Makes an editable copy of the unit in a different project.
createAndInsertProject(String, String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
       Create a new project and insert it into current workspace
createAutoFlowChart() - Method in interface com.telelogic.rhapsody.core.IRPOperation
       Creates a flowchart for the operation.
createCodeModel(IRPCollection) - Method in class com.telelogic.rhapsody.core.RPExternalRoundtrip
       method CreateCodeModel
createDefaultTransition(IRPState) - Method in interface com.telelogic.rhapsody.core.IRPConnector

Creates a default transition leading to this connector, within the state specified.

createDefaultTransition(IRPState) - Method in interface com.telelogic.rhapsody.core.IRPState

Creates a default transition within the state.

createDomainFromProfile(IRPProfile, String) - Method in interface com.telelogic.rhapsody.core.IRPApplication

Create Domain from Profile After Login

createGraphics() - Method in interface com.telelogic.rhapsody.core.IRPStatechart

Creates the graphical representation of the elements in the statechart.

createGraphics() - Method in interface com.telelogic.rhapsody.core.IRPStatechartDiagram

Creates the graphical representation of the elements in the statechart.

createJavaPluginManager(IRPApplication) - Static method in class com.telelogic.rhapsody.core.RhapsodyAppServer

For internal use only.

createNestedStatechart() - Method in interface com.telelogic.rhapsody.core.IRPState

Creates a sub-statechart for the state.

createNewCollection() - Method in interface com.telelogic.rhapsody.core.IRPApplication

creates a new Rhapsody collection object

createNewProject(String, String) - Method in interface com.telelogic.rhapsody.core.IRPApplication

Creates a new Rhapsody project

createOSLCLink(String, String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement

Creates an OSLC link between the element and the element represented by the specified URL.

createProgressTask(int, int, String, int, int) - Method in class com.telelogic.rhapsody.core.RPExternalIDEManager

Create a Progress Task

createRhapsodyApplication() - Static method in class com.telelogic.rhapsody.core.RhapsodyAppServer

Creates a new instance of Rhapsody and provides access to it.

createRhapsodyApplicationDllServer() - Static method in class com.telelogic.rhapsody.core.RhapsodyAppServer

For internal use only.

createSD2(IRPSequenceDiagram, String) - Method in interface com.telelogic.rhapsody.core.IRPDiagSynthAPI

DiagSynthAPI : create sequence diagram

createSearchQuery() - Method in interface com.telelogic.rhapsody.core.IRPSearchManager

method Create search query

createUninitializedRhapsodyApplicationDllServer() - Static method in class com.telelogic.rhapsody.core.RhapsodyAppServer

For internal use only.

CSV - Static variable in class com.telelogic.rhapsody.core.IRPMatrixView.ContentFormat

Export in Comma Separated Value (CSV) format.

CSV - Static variable in class com.telelogic.rhapsody.core.IRPTableView.ContentFormat

Export in Comma Separated Value (CSV) format.

cutSelected() - Method in interface com.telelogic.rhapsody.core.IRPSelection

Perform CUT operation under current selection

D

dbgCheckComIn(short) - Method in interface com.telelogic.rhapsody.core.IRPApplication
  For debug - check communication in
dbgCheckComOut() - Method in interface com.telelogic.rhapsody.core.IRPApplication
  For debug - check communication out
dblClickNotify(int, int, String) - Method in interface com.telelogic.rhapsody.core.IRPowListListener
  method DblClickNotify
dblClickNotify(int, String) - Method in interface com.telelogic.rhapsody.core.IRPowTextListener
  method DblClickNotify
deferredAddToModel(String, int, String, int) - Method in interface
com.telelogic.rhapsody.core.IRPApplication
  Add Rhapsody unit to current project
deleteActor(IRPActor) - Method in interface com.telelogic.rhapsody.core.IRPPackage
  Deletes the specified actor.
deleteArgument(IRPArgument) - Method in interface com.telelogic.rhapsody.core.IRPOperation
  Deletes a specific argument from the operation.
deleteAttribute(IRPAttribute) - Method in interface com.telelogic.rhapsody.core.IRPClassifier
  Deletes the specified attribute.
deleteClass(String) - Method in interface com.telelogic.rhapsody.core.IRPClass
  Deletes the specified class from the current class.
deleteClass(IRPClass) - Method in interface com.telelogic.rhapsody.core.IRPPackage
  Deletes the specified class.
deleteCollaborationDiagram(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
  Deletes the collaboration diagram with the specified name.
deleteComponent(IRPComponent) - Method in interface com.telelogic.rhapsody.core.IRPProject
  Deletes the specified Component.
deleteComponentDiagram(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
  Deletes the component diagram with the specified name.
deleteComponentInstance(String) - Method in interface com.telelogic.rhapsody.core.IRPNode
  method deleteComponentInstance
deleteConfiguration(IRPConfiguration) - Method in interface com.telelogic.rhapsody.core.IRPComponent
  Deletes the specified configuration.
deleteConnector(IRPConnector) - Method in interface com.telelogic.rhapsody.core.IRPState
  Deletes the specified connector element.
deleteConstructor(IRPOperation) - Method in interface com.telelogic.rhapsody.core.IRPClass
  Deletes the specified constructor from the current class.
deleteDependency(IRPDependency) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
  Deletes the specified dependency from the model.
deleteDeploymentDiagram(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
  Deletes the deployment diagram with the specified name.
deleteDescribingDiagram(IRPDiagram) - Method in interface com.telelogic.rhapsody.core.IRPUseCase
  method deleteDescribingDiagram

deleteDestructor() - Method in interface com.telelogic.rhapsody.core.IRPClass
> Deletes the destructor for the class.

deleteEntryPoint(String) - Method in interface com.telelogic.rhapsody.core.IRPUseCase
> method deleteEntryPoint

deleteEnumerationLiteral(IRPEnumerationLiteral) - Method in interface
com.telelogic.rhapsody.core.IRPType
> method deleteEnumerationLiteral

deleteEvent(IRPEvent) - Method in interface com.telelogic.rhapsody.core.IRPPackage
> Deletes the specified event.

deleteEventReception(IRPEventReception) - Method in interface com.telelogic.rhapsody.core.IRPClass
> Deletes the specified event reception.

deleteExtensionPoint(String) - Method in interface com.telelogic.rhapsody.core.IRPUseCase
> method deleteExtensionPoint

deleteFile(IRPFile) - Method in interface com.telelogic.rhapsody.core.IRPComponent
> Deletes the specified File.

deleteFlowchart() - Method in interface com.telelogic.rhapsody.core.IRPOperation
> Deletes the flowchart or activity defined for the operation.

deleteFlowItems(IRPFlowItem) - Method in interface com.telelogic.rhapsody.core.IRPClassifier
> Deletes the specified item flow.

deleteFlowItems(IRPFlowItem) - Method in interface com.telelogic.rhapsody.core.IRPPackage
> Deletes the specified item flow.

deleteFlows(IRPFlow) - Method in interface com.telelogic.rhapsody.core.IRPClassifier
> Deletes the specified flow.

deleteFlows(IRPFlow) - Method in interface com.telelogic.rhapsody.core.IRPPackage
> Deletes the specified flow.

deleteFromProject() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Deletes the current model element from the model.

deleteGeneralization(IRPClassifier) - Method in interface com.telelogic.rhapsody.core.IRPClassifier
> Deletes the generalization relationship between the classifier and the classifier specified as a parameter.

deleteGlobalFunction(IRPOperation) - Method in interface com.telelogic.rhapsody.core.IRPPackage
> Deletes the specified global function.

deleteGlobalObject(IRPRelation) - Method in interface com.telelogic.rhapsody.core.IRPPackage
> Deletes the specified object.

deleteGlobalVariable(IRPAttribute) - Method in interface com.telelogic.rhapsody.core.IRPPackage
> Deletes the specified global variable.

deleteInitialInstance(IRPModelElement) - Method in interface
com.telelogic.rhapsody.core.IRPConfiguration
> method deleteInitialInstance

deleteInternalTransition(IRPTransition) - Method in interface com.telelogic.rhapsody.core.IRPState
> method deleteInternalTransition

deleteNode(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
> Deletes the Node element with the specified name.

deleteObjectModelDiagram(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
> Deletes the object model diagram with the specified name.

deleteOperation(IRPOperation) - Method in interface com.telelogic.rhapsody.core.IRPClassifier
> Deletes the specified operation.

deleteOSLCLink(String, String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Deletes the specified OSLC link from the model.

deletePackage() - Method in interface com.telelogic.rhapsody.core.IRPPackage
> Deletes the package.

D

deletePanelDiagram(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Deletes the panel diagram with the specified name.

deleteProjectFromList(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
>   Delete specified project from current workspace

deleteReception(IRPEventReception) - Method in interface com.telelogic.rhapsody.core.IRPClass
>   Deletes the specified reception from the current class.

deleteRelation(IRPRelation) - Method in interface com.telelogic.rhapsody.core.IRPClassifier
>   Deletes the specified relation.

deleteSelected() - Method in interface com.telelogic.rhapsody.core.IRPSelection
>   Perform DELETE operation under current selection

deleteSequenceDiagram(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Deletes the sequence diagram with the specified name.

deleteState(IRPState) - Method in interface com.telelogic.rhapsody.core.IRPStatechart
>   Deletes the specified state from the statechart.

deleteStaticReaction(IRPTransition) - Method in interface com.telelogic.rhapsody.core.IRPState
>   Deletes the specified internal transition.

deleteSuperclass(IRPClass) - Method in interface com.telelogic.rhapsody.core.IRPClass
>   Removes the inheritance relationship with the specified base class.

deleteTimingDiagram(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Deletes the timing diagram with the specified name.

deleteTransition(IRPTransition) - Method in interface com.telelogic.rhapsody.core.IRPStateVertex
>   Deletes the specified transition.

deleteType(String) - Method in interface com.telelogic.rhapsody.core.IRPClass
>   Deletes the specified type from the current class.

deleteType(IRPType) - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Deletes the specified type.

deleteUseCase(IRPUseCase) - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Deletes the specified use case.

deleteUseCaseDiagram(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Deletes the use case diagram with the specified name.

DEPENDENCY - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column.DependsOn
>   Value to be used for Property parameter of addColumn method, when FlowAttribute is selected for
>   the Type parameter of addColumn method.

DEPENDS_ON - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column
>   Value used for Type parameter of addColumn method.

DERIVES - Static variable in class com.telelogic.rhapsody.core.IRPModelElement.OSLCLink.Types
>   OSLC link type: Derives

DESCRIPTION - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.GeneralAttribute
>   Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for
>   the Type parameter of addColumn method.

DESCRIPTIONS - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

destroyWindow() - Method in interface com.telelogic.rhapsody.core.IRPPlugInWindow
>   Destroy window

DETAILED - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.ViewsToSearch

DIAGRAM_ELEMENT - Static variable in class
com.telelogic.rhapsody.core.IRPSearchQuery.References.RelationKind

disconnect() - Method in class com.telelogic.rhapsody.core.RPApplicationListener

D

disconnect() - Method in class com.telelogic.rhapsody.core.RPCodeGeneratorListener

disconnect() - Method in class com.telelogic.rhapsody.core.RPCodeGenSimplifier

disconnect() - Method in class com.telelogic.rhapsody.core.RPExternalCheck

disconnect() - Method in class com.telelogic.rhapsody.core.RPExternalCodeGenerator

disconnect() - Method in class com.telelogic.rhapsody.core.RPExternalIDEManager

disconnect() - Method in class com.telelogic.rhapsody.core.RPExternalRoundtrip

disconnect() - Method in class com.telelogic.rhapsody.core.RPIntegratorListener

disconnect() - Method in class com.telelogic.rhapsody.core.RPJavaPluginsManager

disconnect() - Method in class com.telelogic.rhapsody.core.RPowPaneMgrEvents

disconnect() - Method in class com.telelogic.rhapsody.core.RPRoundTripListener

disconnect() - Method in class com.telelogic.rhapsody.core.RPRTCListener

disconnect() - Method in class com.telelogic.rhapsody.core.RPSearchListener

disconnectFromTarget() - Method in interface com.telelogic.rhapsody.core.IRPApplication
    disconnectFromTarget
dMRefreshRecursive(IRPUnit) - Method in interface com.telelogic.rhapsody.core.IRPApplication
    dMRefreshRecursive
dMSyncAndRefresh(IRPProject, int, int) - Method in interface
com.telelogic.rhapsody.core.IRPApplication
    dMSyncAndRefresh
doAbort() - Method in class com.telelogic.rhapsody.core.RPCodeGenSimplifier
    abort the simplification
doCommand(long) - Method in interface com.telelogic.rhapsody.core.IRPAXViewCtrl
    Execute command by command id
doExit() - Method in class com.telelogic.rhapsody.core.RPCodeGenSimplifier
    exit and allow Rhapsody to exit
doExit() - Method in class com.telelogic.rhapsody.core.RPExternalCheck
    exit and allow Rhapsody to exit
downloadToTarget(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
    downloadToTarget

Package Class Use **Tree** **Serialized** **Deprecated** **Index** **Help**
PREV LETTER **NEXT LETTER**   **FRAMES** **NO FRAMES** **All Classes**
A B C D E F G H I L M N O P Q R S T U V W X

D   834

E

ELEMENT_TYPE - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.GeneralAttribute
>   Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for
>   the Type parameter of addColumn method.

ELEMENTS_LIST - Static variable in class
com.telelogic.rhapsody.core.IRPMatrixLayout.QueryOrElementsList
>   When ELEMENTS_LIST is used as the parameter for the methods
>   setFromElementTypesUseQueryOrElementsList and setToElementTypesUseQueryOrElementsList, it
>   indicates that elements selected in the element types list will be used to specify the "from" element
>   types or "to" element types for the matrix.

ELEMENTS_LIST - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.QueryOrElementsList
>   When ELEMENTS_LIST is used as the parameter for the methods
>   setFromElementTypesUseQueryOrElementsList and setToElementTypesUseQueryOrElementsList, it
>   indicates that elements selected in the element types list will be used to specify the "from" element
>   types or "to" element types for the relation table.

embedFlow(IRPFlow) - Method in interface com.telelogic.rhapsody.core.IRPGraphEdge
>   method embedFlow

embedNewFlow() - Method in interface com.telelogic.rhapsody.core.IRPGraphEdge
>   method embedNewFlow

empty() - Method in interface com.telelogic.rhapsody.core.IRPCollection
>   Used to empty out a collection.

endSimplification() - Method in class com.telelogic.rhapsody.core.RPCodeGenSimplifier
>   after all simplifications

endTransactionOfNoCGInterest() - Method in interface com.telelogic.rhapsody.core.IRPProject
>   For internal use only.

endUndoTransaction() - Method in interface com.telelogic.rhapsody.core.IRPApplication
>   end undo transaction

enterAnimationCommand(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
>   enter Animation Command

ENUMERATION_LITERAL_VALUE - Static variable in class
com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

errorMessage() - Method in interface com.telelogic.rhapsody.core.IRPApplication
>   Returns error message for last method called.

errorMessage() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
>   Returns error message for last method called.

EXACTLY - Static variable in class
com.telelogic.rhapsody.core.IRPSearchQuery.References.QuantityOperator

executeCommand(String, IRPCollection, IRPCollection) - Method in interface
com.telelogic.rhapsody.core.IRPApplication
> method execute command

executeCommand(String, IRPCollection, IRPCollection) - Method in interface
com.telelogic.rhapsody.core.IRPAXViewCtrl
> Execute command

executeCommand(IRPUnit, String, String) - Method in class com.telelogic.rhapsody.core.RPRTCListener
> Called on request to execute a Command

executeCommandLine(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> Execute command line

executeTransformationSequence(String, int) - Method in interface
com.telelogic.rhapsody.core.IRPApplication
> executeTransformationSequence

exit() - Method in class com.telelogic.rhapsody.core.RPExternalCodeGenerator
> method Exit

expandStringKeywords(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> expand environment-variable keywords in the provided string

EXTERNAL - Static variable in class com.telelogic.rhapsody.core.IRPModelElement.OSLCLink.Types
> OSLC link type: External

F

factoryMutex - Static variable in class com.telelogic.rhapsody.core.RPExtendedRPClassesFactory

finalize() - Method in class com.telelogic.rhapsody.core.RPApplicationListener

finalize() - Method in class com.telelogic.rhapsody.core.RPCodeGeneratorListener

finalize() - Method in class com.telelogic.rhapsody.core.RPCodeGenSimplifier

finalize() - Method in class com.telelogic.rhapsody.core.RPExternalCheck

finalize() - Method in class com.telelogic.rhapsody.core.RPExternalCodeGenerator

finalize() - Method in class com.telelogic.rhapsody.core.RPExternalIDEManager

finalize() - Method in class com.telelogic.rhapsody.core.RPExternalRoundtrip

finalize() - Method in class com.telelogic.rhapsody.core.RPIntegratorListener

finalize() - Method in class com.telelogic.rhapsody.core.RPJavaPluginsManager

finalize() - Method in class com.telelogic.rhapsody.core.RPowPaneMgrEvents

finalize() - Method in class com.telelogic.rhapsody.core.RPRoundTripListener

finalize() - Method in class com.telelogic.rhapsody.core.RPRTCListener

finalize() - Method in class com.telelogic.rhapsody.core.RPSearchListener

findActor(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
     Returns the actor with the specified name.
findAllByName(String, String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
     Searches the package for a model element of the specified type with the specified name.
findAttribute(String) - Method in interface com.telelogic.rhapsody.core.IRPClassifier
     Returns the attribute with the name specified.
findBaseClassifier(String) - Method in interface com.telelogic.rhapsody.core.IRPClassifier
     Returns the base classifier with the specified name.
findClass(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
     Returns the class with the specified name.
findClass(String) - Static method in class com.telelogic.rhapsody.core.RhapsodyAppServer
     For internal use only.
findComponent(String) - Method in interface com.telelogic.rhapsody.core.IRPProject

Returns the Component with the specified name.

findComponentInstance(String) - Method in interface com.telelogic.rhapsody.core.IRPNode
method findComponentInstance

findConfiguration(String) - Method in interface com.telelogic.rhapsody.core.IRPComponent
Returns the configuration with the specified name.

findDerivedClassifier(String) - Method in interface com.telelogic.rhapsody.core.IRPClassifier
Returns the derived classifier with the specified name.

findElementByBinaryID(byte[]) - Method in interface com.telelogic.rhapsody.core.IRPProject
Retrieves a model element based on its binary ID.

findElementByFileName(String, String) - Method in interface com.telelogic.rhapsody.core.IRPProject
Returns the top-level element in the specified Rhapsody unit file.

findElementByGUID(String) - Method in interface com.telelogic.rhapsody.core.IRPProject
Retrieves a model element based on its GUID.

findElementsByFullName(String, String) - Method in interface
com.telelogic.rhapsody.core.IRPModelElement
Searches for the specified model element in the specified path under the current model element.

findElementsWithOSLCLink(String, String) - Method in interface com.telelogic.rhapsody.core.IRPProject
Returns a collection of all the model elements that have an OSLC link of the specified type to the
specified target element.

findEntryPoint(IRPGeneralization) - Method in interface com.telelogic.rhapsody.core.IRPUseCase
method findEntryPoint

findEvent(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
Returns the event with the specified name.

findExtensionPoint(IRPGeneralization) - Method in interface com.telelogic.rhapsody.core.IRPUseCase
method findExtensionPoint

findGeneralization(String) - Method in interface com.telelogic.rhapsody.core.IRPClassifier
Returns the element representing the generalization relationship between this classifier and the
classifier whose name was specified as a parameter.

findGlobalFunction(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
Returns the global function with the specified name.

findGlobalObject(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
Returns the Object with the specified name.

findGlobalVariable(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
Returns the global variable with the specified name.

findInterfaceItem(String) - Method in interface com.telelogic.rhapsody.core.IRPClassifier
Gets the operation or event reception that matches the signature provided.

findLibrary(String) - Method in class com.telelogic.rhapsody.core.RhpClassLoader

findNestedClassifier(String) - Method in interface com.telelogic.rhapsody.core.IRPClassifier
Searches for the nested classifier with the name specified.

findNestedClassifierRecursive(String) - Method in interface com.telelogic.rhapsody.core.IRPClassifier
Searches recursively for the classifier with the name specified.

findNestedElement(String, String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
Searches for the specified model element.

findNestedElementRecursive(String, String) - Method in interface
com.telelogic.rhapsody.core.IRPModelElement
Searches recursively for the specified model element.

findNode(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
Returns the Node element with the specified name.

findRelation(String) - Method in interface com.telelogic.rhapsody.core.IRPClassifier
Returns the association whose name was specified as a parameter.

findTrigger(String) - Method in interface com.telelogic.rhapsody.core.IRPClassifier
> Returns the trigger with the specified name in the classifier's statechart.

findTrigger(IRPInterfaceItem) - Method in interface com.telelogic.rhapsody.core.IRPStatechart
> Checks whether the specified IRPInterfaceItem element serves as the trigger of a transition in the statechart.

findType(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
> Returns the type with the specified name.

findUsage(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPPackage
> Returns a collection of the elements in the current package that are related to the specified model element.

findUseCase(String) - Method in interface com.telelogic.rhapsody.core.IRPPackage
> Returns the use case with the specified name.

finishProgressTask(int, int) - Method in class com.telelogic.rhapsody.core.RPExternalIDEManager
> Finish a Progress Task

fixpack() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> Get Rhapsody fixpack

FLOW_ATTRIBUTE - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column
> Value used for Type parameter of addColumn method

forceOutput2Console(int) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> Force output to system console

forceRoundtrip() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> forceRoundtrip

forceRoundtripElements(IRPCollection) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> method forceRoundtripElements

FROM_ELEMENT - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.RelationAttributeFrom
> Value to be used for Property parameter of addColumn method, when RelationAttributeFrom is selected for the Type parameter of addColumn method.

FULL_PATH_NAME - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.GeneralAttribute

---

G

gatewayExportToXML(String, String) - Method in interface com.telelogic.rhapsody.core.IRPProject
>	For internal use only.

gatewayExportToXML2(String, String, IRPProject) - Method in interface com.telelogic.rhapsody.core.IRPProject
>	For internal use only.

GENERAL_ATTRIBUTE - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column
>	Value used for Type parameter of addColumn method

generate() - Method in interface com.telelogic.rhapsody.core.IRPApplication
>	Generates code for the entire project, using the active component and configuration.

generate(IRPModelElement, IRPCollection, IRPCollection, int, int) - Method in class com.telelogic.rhapsody.core.RPExternalCodeGenerator
>	method Generate

generateElements(IRPCollection) - Method in interface com.telelogic.rhapsody.core.IRPApplication
>	method generateElements

generateEntireProject() - Method in interface com.telelogic.rhapsody.core.IRPApplication
>	generateEntireProject

generateMainAndMakeFiles() - Method in interface com.telelogic.rhapsody.core.IRPApplication
>	Generate Main and Make Files

generateReport(String, String, String, String, int, int) - Method in interface com.telelogic.rhapsody.core.IRPProject
>	Generates a ReporterPLUS report for the model.

generateSequence(String, IRPPackage) - Method in interface com.telelogic.rhapsody.core.IRPCollaboration
>	Generates a sequence diagram from the content of the IRPCollaboration object.

generateWithDependencies() - Method in interface com.telelogic.rhapsody.core.IRPApplication
>	generateWithDependencies

getActivationCondition() - Method in interface com.telelogic.rhapsody.core.IRPCollaboration
>	get property activationCondition

getActivationMode() - Method in interface com.telelogic.rhapsody.core.IRPCollaboration
>	get property activationMode

getActivator(IRPMessage) - Method in interface com.telelogic.rhapsody.core.IRPCollaboration
>	method getActivator

getActiveComponent() - Method in interface com.telelogic.rhapsody.core.IRPProject
>	Returns the active component.

getActiveConfiguration() - Method in interface com.telelogic.rhapsody.core.IRPProject
>	Returns the active configuration.

getActiveCustomViewsOnBrowser() - Method in interface com.telelogic.rhapsody.core.IRPProject
>	Returns a collection of the custom views currently applied to the browser.

getActiveCustomViewsOnDiagram(IRPDiagram) - Method in interface com.telelogic.rhapsody.core.IRPProject
>	Returns a collection of the custom views currently applied to the specified diagram.

getActiveRhapsodyApplication() - Static method in class com.telelogic.rhapsody.core.RhapsodyAppServer
> Accesses the currently running instance of Rhapsody.

getActiveRhapsodyApplicationByID(String) - Static method in class
com.telelogic.rhapsody.core.RhapsodyAppServer
> Accesses the instance of Rhapsody that is registered in the ROT (Running Object Table) with the
> specified ID.

getActiveRhapsodyApplicationIDList() - Static method in class
com.telelogic.rhapsody.core.RhapsodyAppServer
> Returns a list of the strings representing the Rhapsody instances currently registered in the ROT
> (Running Object Table).

getActiveView() - Method in class com.telelogic.rhapsody.core.RPExternalIDEManager
> method GetActiveView

getActivityDiagram() - Method in interface com.telelogic.rhapsody.core.IRPClassifier
> This method should no longer be used because Rational Rhapsody now allows you to define more
> than one statechart and activity diagram for a class.

getActors() - Method in interface com.telelogic.rhapsody.core.IRPPackage
> Returns a collection of all the actors in the package.

getActualParameterList() - Method in interface com.telelogic.rhapsody.core.IRPMessage
> get property actualParameterList

getAdditionalSources() - Method in interface com.telelogic.rhapsody.core.IRPComponent
> Returns the additional sources defined for the component.

getAdditionalSources() - Method in interface com.telelogic.rhapsody.core.IRPConfiguration
> get property additionalSources

getAddToModelMode() - Method in interface com.telelogic.rhapsody.core.IRPUnit
> Returns an indication of how the unit was added to the model.

getAllElementsInInstrumentationScope() - Method in interface
com.telelogic.rhapsody.core.IRPConfiguration
> Checks whether the instrumentation mode selected for the configuration applies to all elements or just
> selected elements.

getAllGraphicalProperties() - Method in interface com.telelogic.rhapsody.core.IRPGraphElement
> method getAllGraphicalProperties

getAllNestedElements() - Method in interface com.telelogic.rhapsody.core.IRPPackage
> Returns a collection of all the model elements that are directly under the current package, including
> functions, global variables, and global objects.

getAllProperties() - Method in interface com.telelogic.rhapsody.core.IRPGraphElement
> method getAllProperties

getAllStereotypes() - Method in interface com.telelogic.rhapsody.core.IRPProject
> Returns a collection of all the stereotypes in the project.

getAllTags() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Returns a collection of all the element's tags.

getAllTriggers() - Method in interface com.telelogic.rhapsody.core.IRPStatechart
> Returns a collection of all the triggers in the statechart

getAnchoredByMe() - Method in interface com.telelogic.rhapsody.core.IRPAnnotation
> Gets the list of model elements that are anchored to the annotation.

getAnnotations() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Returns all of the element's annotations.

getApplication() - Method in interface com.telelogic.rhapsody.core.IRPRhapsodyServer
> getApplication

getApplicationConnectionString() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> getApplicationConnectionString

getApplicationName(String, String) - Method in interface com.telelogic.rhapsody.core.IRPApplication

Get application name

getApplicationStatus() - Method in interface com.telelogic.rhapsody.core.IRPApplication
  getApplicationStatus

getArgumentDirection() - Method in interface com.telelogic.rhapsody.core.IRPArgument
  Returns the direction of the argument (In, Out, or InOut).

getArguments() - Method in interface com.telelogic.rhapsody.core.IRPInterfaceItem
  Returns a collection of all the arguments for the operation (collection of IRPArgument objects).

getArgVals() - Method in interface com.telelogic.rhapsody.core.IRPSendAction
  Returns a collection of the argument values that were set for the event associated with the Send Action element.

getArgValue() - Method in interface com.telelogic.rhapsody.core.IRPTemplateInstantiationParameter
  get property declaration

getAssociatedImage() - Method in interface com.telelogic.rhapsody.core.IRPGraphElement
  get associatedImage

getAssociationClass() - Method in interface com.telelogic.rhapsody.core.IRPRelation
  method getAssociationClass

getAssociationClasses() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
  Returns a collection of all the association classes directly beneath this model element.

getAssociations() - Method in interface com.telelogic.rhapsody.core.IRPCollaboration
  get property associations

getAttributes() - Method in interface com.telelogic.rhapsody.core.IRPClassifier
  Returns a collection of all the classifier's attributes.

getAttributesIncludingBases() - Method in interface com.telelogic.rhapsody.core.IRPClassifier
  Returns a collection of all the classifier's attributes, including those it inherits from its base classifiers.

getAttributeValue(String) - Method in interface com.telelogic.rhapsody.core.IRPInstance
  method getAttributeValue

getBase() - Method in interface com.telelogic.rhapsody.core.IRPTag
  Returns the base tag on which the local copy of the tag is based.

getBaseClass() - Method in interface com.telelogic.rhapsody.core.IRPGeneralization
  get method baseClass

getBaseClassifiers() - Method in interface com.telelogic.rhapsody.core.IRPClassifier
  Returns a collection of the classifiers that server as base classifiers for this classifier.

getBaseEvent() - Method in interface com.telelogic.rhapsody.core.IRPEvent
  get property baseEvent

getBehavioralDiagrams() - Method in interface com.telelogic.rhapsody.core.IRPClassifier
  Returns a collection of all the statecharts and activities defined for the classifier.

getBehavioralDiagrams() - Method in interface com.telelogic.rhapsody.core.IRPPackage
  Returns a collection of all the activity diagrams in the package.

getBinaryID() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
  Returns the GUID of the model element as an array of bytes, as opposed to the method getGUID, which returns the GUID as a string.

getBody() - Method in interface com.telelogic.rhapsody.core.IRPAction
  Gets the code defined as the action for the transition.

getBody() - Method in interface com.telelogic.rhapsody.core.IRPAnnotation
  Gets the text of the specification for the annotation.

getBody() - Method in interface com.telelogic.rhapsody.core.IRPGuard
  get property body

getBody() - Method in interface com.telelogic.rhapsody.core.IRPOperation
  Returns the body of the operation.

getBody() - Method in interface com.telelogic.rhapsody.core.IRPTrigger
  get property body

getBuildNo() - Method in interface com.telelogic.rhapsody.core.IRPApplication
>   get property BuildNo

getBuildSet() - Method in interface com.telelogic.rhapsody.core.IRPConfiguration
>   get property buildSet

getBuildType() - Method in interface com.telelogic.rhapsody.core.IRPComponent
>   Returns the build type of the component - Library, Executable, or Analysis.

getCellElementTypes() - Method in interface com.telelogic.rhapsody.core.IRPMatrixLayout
>   Returns a collection of the element types that were specified to be displayed in the cells of the matrix.

getCgSimplifiedModelPackage() - Method in interface com.telelogic.rhapsody.core.IRPProject
>   Returns the package that contains the simplified model.

getClasses() - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Returns a collection of all the classes in the package.

getClassifier() - Method in interface com.telelogic.rhapsody.core.IRPCollaboration
>   Returns a collection of all the instance lines in the sequence diagram.

getClassifier() - Method in interface com.telelogic.rhapsody.core.IRPInstanceSpecification
>   get property classifier

getClassifierRole() - Method in interface com.telelogic.rhapsody.core.IRPMessagePoint
>   method getClassifierRole

getClassifierRoles() - Method in interface com.telelogic.rhapsody.core.IRPAssociationRole
>   Returns a collection of the classifier roles that are linked by the association role.

getCMHeader() - Method in interface com.telelogic.rhapsody.core.IRPUnit
>   Returns the header used by the Configuration Management tool for the unit.

getCMState() - Method in interface com.telelogic.rhapsody.core.IRPUnit
>   Returns the configuration management state of the unit.

getCodeAnnotations(IRPModelElement, int) - Method in interface com.telelogic.rhapsody.core.IRPCodeGenerator
>   method to get generated code file names

getCodeGeneratedFiles() - Method in interface com.telelogic.rhapsody.core.IRPProject
>   Returns a collection of filenames for the code files that will be generated for the current active component if you select the "regenerate" option.

getCodeGenSimplifiersRegistry() - Method in interface com.telelogic.rhapsody.core.IRPApplication
>   get the code generation simplifiers registry

getCollaborationDiagrams() - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Returns a collection of all the collaboration diagrams in the package.

getCollapseFirstColumn() - Method in interface com.telelogic.rhapsody.core.IRPTableLayout
>   Checks whether the first column of the layout includes controls for collapsing and expanding rows that have the same value in the first column.

getColumnContext(int) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout
>   Returns the context pattern label that was specified for the column.

getColumnCount() - Method in interface com.telelogic.rhapsody.core.IRPTableLayout
>   Returns the number of columns in the table layout.

getColumnDefaultWidth(int) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout
>   Returns the default width that was defined for the specified column.

getColumnImplementationAllowNew(int) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout
>   Checks whether the user-defined picker for the specified column includes the New option in its list.

getColumnImplementationAllowSelect(int) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout
>   Checks whether the user-defined picker for the specified column includes the Select option in its list.

getColumnImplementationCellType(int) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout

G

Returns the type of information that is displayed in the column's cells - string, model element, or list of model elements.

getColumnImplementationDisplayProperty(int) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout

Returns the type of element information that is displayed when the cell value type is set to model element or list of model elements.

getColumnImplementationGetterCode(int) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout

Returns the Java code for the getter for the cells in the specified column.

getColumnImplementationImports(int) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout

For columns that use customized cell behavior, this method returns the list of imports specified for the column.

getColumnImplementationPickerCode(int) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout

Returns the Java code for the picker for the cells in the specified column.

getColumnImplementationSetterCode(int) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout

Returns the Java code for the setter for the cells in the specified column.

getColumnName(int) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout

Returns the name of the specified column.

getColumnProperty(int) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout

Returns the Property of the specified column.

getColumns() - Method in interface com.telelogic.rhapsody.core.IRPTableLayout

getColumnType(int) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout

Returns the type of the specified table column.

getCommunicationConnection() - Method in interface com.telelogic.rhapsody.core.IRPMessage

get property communicationConnection

getCompilerSwitches() - Method in interface com.telelogic.rhapsody.core.IRPConfiguration

get property compilerSwitches

getCompleteness() - Method in class com.telelogic.rhapsody.core.RPExternalCheck

Return true if this is a check for completeness or false if this is a check for correctness

getComponentDiagrams() - Method in interface com.telelogic.rhapsody.core.IRPPackage

Returns a collection of all the component diagrams in the package.

getComponentInstances() - Method in interface com.telelogic.rhapsody.core.IRPNode

get property componentInstances

getComponents() - Method in interface com.telelogic.rhapsody.core.IRPProject

Returns a collection of all the components in the project.

getComponentType() - Method in interface com.telelogic.rhapsody.core.IRPComponentInstance

get property componentType

getConcurrentGroup(IRPMessage) - Method in interface com.telelogic.rhapsody.core.IRPCollaboration

method getConcurrentGroup

getCondition() - Method in interface com.telelogic.rhapsody.core.IRPMessage

get property condition

getConfigByDependency(IRPDependency) - Method in interface com.telelogic.rhapsody.core.IRPComponent

method getConfigByDependency

getConfigurations() - Method in interface com.telelogic.rhapsody.core.IRPComponent

Returns a collection of all the configurations in the component.

getConnectorType() - Method in interface com.telelogic.rhapsody.core.IRPConnector

Returns the type of the connector: Condition, Diagram, EnterExit, Fork, History, Join, Junction, Termination, InPin, OutPin, or InOutPin.

getConstraints() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
    Returns all of the element's constraints.

getConstraintsByHim() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
    For internal use only.

getConstraintsByMe() - Method in interface com.telelogic.rhapsody.core.IRPConstraint
    Returns all of the model elements affected by this constraint.

getContainedMessages() - Method in interface com.telelogic.rhapsody.core.IRPInteractionOperand
    Returns a collection of all the messages contained in the interaction operand.

getContainingArrow() - Method in interface com.telelogic.rhapsody.core.IRPGraphEdge
    method getContainingArrow

getContent(String) - Method in interface com.telelogic.rhapsody.core.IRPMatrixView
    Retrieves the content of the matrix in the specified format.

getContent(String) - Method in interface com.telelogic.rhapsody.core.IRPTableView
    Retrieves the content of the table in the specified format.

getContents() - Method in interface com.telelogic.rhapsody.core.IRPSwimlane
    get property contents

getContract() - Method in interface com.telelogic.rhapsody.core.IRPPort
    get property contract

getControlledFiles() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
    Returns a collection of all the element's controlled files.

getConveyed() - Method in interface com.telelogic.rhapsody.core.IRPFlow
    get property conveyed

getCorrespondingGraphicElements(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPDiagram
    Returns the graphical elements that represent the specified model element in the diagram.

getCount() - Method in interface com.telelogic.rhapsody.core.IRPCollection
    Returns the number of items in a collection.

getCPUtype() - Method in interface com.telelogic.rhapsody.core.IRPNode
    get property CPUtype

getCurrentDirectory() - Method in interface com.telelogic.rhapsody.core.IRPUnit
    Gets the name of the directory that contains the file used to store the unit.

getDeclaration() - Method in interface com.telelogic.rhapsody.core.IRPType
    get property declaration

getDeclaration() - Method in interface com.telelogic.rhapsody.core.IRPVariable
    Returns the type declaration if an on-the-fly type was used for the element rather than an existing type.

getDecorationStyle() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
    Returns the name of the decoration style currently associated with the model element.

getDefaultDirectoryScheme() - Method in interface com.telelogic.rhapsody.core.IRPProject
    Returns the project's default directory scheme with regard to packages.

getDefaultTransition() - Method in interface com.telelogic.rhapsody.core.IRPState
    Returns the default transition within the state.

getDefaultValue() - Method in interface com.telelogic.rhapsody.core.IRPVariable
    Returns the default value that was set for the variable.

getDependencies() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
    Returns all of the element's dependencies.

getDependent() - Method in interface com.telelogic.rhapsody.core.IRPDependency
    Returns the source element in the dependency relation, meaning the element that depends on the other element.

G   845

getDependsOn() - Method in interface com.telelogic.rhapsody.core.IRPDependency
> Returns the target element in the dependency relation, meaning the element on which the first element depends.

getDeploymentDiagrams() - Method in interface com.telelogic.rhapsody.core.IRPPackage
> Returns a collection of all the deployment diagrams in the package.

getDerivedClass() - Method in interface com.telelogic.rhapsody.core.IRPGeneralization
> get method derivedClass

getDerivedClassifiers() - Method in interface com.telelogic.rhapsody.core.IRPClassifier
> Returns a collection of all the classifiers derived from this classifier.

getDerivedInEdges() - Method in interface com.telelogic.rhapsody.core.IRPConnector
> Returns a collection of the transitions coming into the connector.

getDerivedOutEdge() - Method in interface com.telelogic.rhapsody.core.IRPConnector
> Returns the transition exiting the connector.

getDescribingDiagram(String) - Method in interface com.telelogic.rhapsody.core.IRPUseCase
> method getDescribingDiagram

getDescribingDiagrams() - Method in interface com.telelogic.rhapsody.core.IRPUseCase
> get property describingDiagrams

getDescription() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Returns the description defined for the element.

getDescriptionHTML() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Returns HTML representation of the element description.

getDescriptionPlainText() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Returns the description defined for the element in plain text format.

getDescriptionRTF() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Returns the description defined for the element in RTF format.

getDiagram() - Method in interface com.telelogic.rhapsody.core.IRPGraphElement
> method getDiagram

getDiagramOfSelectedElement() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> get diagram of selected element

getDiagSynthAPI(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> for internal use

getDirection() - Method in interface com.telelogic.rhapsody.core.IRPFlow
> get property direction

getDirectory(int, String) - Method in interface com.telelogic.rhapsody.core.IRPConfiguration
> method getDirectory

getDisplayName() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Returns the label of the model element.

getDisplayNameRTF() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Returns the label of the model element as an RTF string.

getDisplayOption(char, String) - Method in interface com.telelogic.rhapsody.core.IRPHyperLink
> getDisplayOption

getDMBoolProperty(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> Returns the value of the specified Design Manager boolean property

getDMModelWorkspaceFolder() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> Returns the path of the workspace folder used for the DM model.

getDMProperty(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> Returns the value of the specified Design Manager property

getDocking() - Method in interface com.telelogic.rhapsody.core.IRPPlugInWindow
> Get docking mode

getDomain() - Method in class com.telelogic.rhapsody.core.RPExternalCheck
> Return the domain of the check which can be user defined or one from predefined list of , or .

getDurationConstraint() - Method in interface com.telelogic.rhapsody.core.IRPMessage
       Gets the text of the Duration Constraint.

getDurationObservation() - Method in interface com.telelogic.rhapsody.core.IRPMessage
       Gets the text of the Duration Observation.

getDurationTime() - Method in interface com.telelogic.rhapsody.core.IRPAcceptTimeEvent
       Returns the duration that was specified for this element.

getElements() - Method in interface com.telelogic.rhapsody.core.IRPFile
       get property elements

getElementsInDiagram() - Method in interface com.telelogic.rhapsody.core.IRPDiagram
       Returns a collection of all the model elements in the diagram.

getElementsInDiagram() - Method in interface com.telelogic.rhapsody.core.IRPStatechart
       Returns a collection of all of the elements in the statechart.

getElementTypes() - Method in interface com.telelogic.rhapsody.core.IRPTableLayout
       Returns a collection of the element types that were specified to be displayed in the table.

getEnd1() - Method in interface com.telelogic.rhapsody.core.IRPAssociationClass
       Gets the relation represented by the first end of the association class.

getEnd1() - Method in interface com.telelogic.rhapsody.core.IRPFlow
       get property end1

getEnd1Multiplicity() - Method in interface com.telelogic.rhapsody.core.IRPLink
       get property end1Multiplicity

getEnd1Name() - Method in interface com.telelogic.rhapsody.core.IRPLink
       get property end1Name

getEnd1Port() - Method in interface com.telelogic.rhapsody.core.IRPFlow
       get property end1Port

getEnd1SysMLPort() - Method in interface com.telelogic.rhapsody.core.IRPFlow
       get property end1SysMLPort

getEnd2() - Method in interface com.telelogic.rhapsody.core.IRPAssociationClass
       Gets the relation represented by the second end of the association class.

getEnd2() - Method in interface com.telelogic.rhapsody.core.IRPFlow
       get property end2

getEnd2Multiplicity() - Method in interface com.telelogic.rhapsody.core.IRPLink
       get property end2Multiplicity

getEnd2Name() - Method in interface com.telelogic.rhapsody.core.IRPLink
       get property end2Name

getEnd2Port() - Method in interface com.telelogic.rhapsody.core.IRPFlow
       get property end2Port

getEnd2SysMLPort() - Method in interface com.telelogic.rhapsody.core.IRPFlow
       get property end2SysMLPort

getEntryAction() - Method in interface com.telelogic.rhapsody.core.IRPState
       Returns the entry action that was defined for the state.

getEntryPoints() - Method in interface com.telelogic.rhapsody.core.IRPUseCase
       get property entryPoints

getEnumerationLiterals() - Method in interface com.telelogic.rhapsody.core.IRPType
       get property enumerationLiterals

getErrorMessage() - Method in interface com.telelogic.rhapsody.core.IRPApplication
       Returns error message for last method called.

getErrorMessage() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
       Returns error message for last method called.

getEvent() - Method in interface com.telelogic.rhapsody.core.IRPAcceptEventAction
       Returns the event that the action waits for.

getEvent() - Method in interface com.telelogic.rhapsody.core.IRPEventReception

method getEvent

getEvent() - Method in interface com.telelogic.rhapsody.core.IRPSendAction

Gets the event sent by the Send Action element.

getEvents() - Method in interface com.telelogic.rhapsody.core.IRPPackage

Returns a collection of all the events in the package.

getEventsBaseId() - Method in interface com.telelogic.rhapsody.core.IRPPackage

Returns the start number used for assigning IDs to events in the package.

getExecutableFolder() - Method in interface com.telelogic.rhapsody.core.IRPApplication

getExecutableFolder

getExecutableName() - Method in interface com.telelogic.rhapsody.core.IRPConfiguration

method getExecutableName

getExecutionOccurrences() - Method in interface com.telelogic.rhapsody.core.IRPCollaboration

Returns a collection of all the Execution Occurrences in the diagram.

getExitAction() - Method in interface com.telelogic.rhapsody.core.IRPState

Returns the exit action that was defined for the state.

getExtendedClass(String) - Method in class com.telelogic.rhapsody.core.RPExtendedRPClassesFactory

getExtensionPoint() - Method in interface com.telelogic.rhapsody.core.IRPGeneralization

get property extensionPoint

getExtensionPoints() - Method in interface com.telelogic.rhapsody.core.IRPUseCase

get property extensionPoints

getExternalCheckerRegistry() - Method in interface com.telelogic.rhapsody.core.IRPApplication

get the External Checker registry

getExternalIDERegistry(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication

get the External IDE registry

getExternalRoundtripInvoker() - Method in interface com.telelogic.rhapsody.core.IRPApplication

getExternalRoundtripInvoker

getFile(IRPClassifier, int) - Method in interface com.telelogic.rhapsody.core.IRPComponent

method getFile

getFileFragments() - Method in interface com.telelogic.rhapsody.core.IRPFile

get property fileFragments

getFileName(IRPClassifier, int, int) - Method in interface com.telelogic.rhapsody.core.IRPComponent

method getFileName

getFilename() - Method in interface com.telelogic.rhapsody.core.IRPUnit

Gets the name of the file used to store the unit.

getFileName(IRPModelElement, IRPModelElement, int, int) - Method in class com.telelogic.rhapsody.core.RPExternalCodeGenerator

method GetFileName

getFiles() - Method in interface com.telelogic.rhapsody.core.IRPComponent

Returns a collection of all the Files in the component.

getFiles() - Method in interface com.telelogic.rhapsody.core.IRPFile

get property files

getFileType() - Method in interface com.telelogic.rhapsody.core.IRPFile

get property fileType

getFileUUID(String) - Method in class com.telelogic.rhapsody.core.RPRTCListener

Called on request to get unit's UUID

getFilterElementTypes() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Returns the element types that are to be searched for the search text.

getFilterReferenceIncludeReferencedElementsInSearchResults() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Checks whether the reference search criterion specified that the referenced elements included in the search criterion should also be displayed in the search results.

getFilterReferenceNameOfReferencedElements() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Returns the model element name that was specified for the reference criterion that was defined.

getFilterReferenceNumberOfReferences() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Returns the number of references that was specified as a search criterion.

getFilterReferenceQuantityOperator() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

When the search criterion includes a specific number of references, this method returns a value that indicates whether the criterion was exactly that number of references, less than that number, or more than that number.

getFilterReferenceRelationKind() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Returns the type of reference used in the search criterion, for example, aggregates or incoming relations.

getFilterReferenceStereotypeOfReferencedElements() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Returns the stereotype that was specified for the reference criterion that was defined.

getFilterReferenceTypeOfReferencedElements() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Returns the model element type that was specified for the reference criterion that was defined.

getFilterSearchInFields() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Returns the list of element fields that the search is to be applied to.

getFilterStereotypes() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Returns the names of the stereotypes that were specified as search criteria.

getFilterSubQueries() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Returns the subqueries that were specified for the search.

getFilterSubQueriesOperator() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Returns indication of how the specified subqueries are to be combined in the search

getFilterSubQueryUseWithNotOperator(IRPTableLayout) - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Checks whether the NOT operator was specified for the specified subquery.

getFilterTagFindAs() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Returns the type of search that was specified for the tag name and tag value search criteria - regular text, wildcard, regular expression, or empty string.

getFilterTagMatchCase() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Checks whether an exact match was specified for the tag name and tag value search criteria, in terms of upper and lower case.

getFilterTagMatchWholeWord() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Checks whether a whole word match was specified for the tag name and tag value search criteria

getFilterTagName() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Returns the tag name specified as a criterion for the search

getFilterTagValue() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Returns the tag value specified as a criterion for the search

getFilterUnitsOnly() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Checks whether the search is limited to model elements that are saved units.

getFilterUnresolvedKind() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Returns the method that was specified for handling unresolved elements in the search.

getFlowchart() - Method in interface com.telelogic.rhapsody.core.IRPActivityDiagram

Returns the IRPFlowchart object underlying the activity diagram.

getFlowchart() - Method in interface com.telelogic.rhapsody.core.IRPOperation

Returns the flowchart or activity defined for the operation.

getFlowchartDiagram() - Method in interface com.telelogic.rhapsody.core.IRPFlowchart

Returns the IRPActivityDiagram object associated with the activity.

getFlowItems() - Method in interface com.telelogic.rhapsody.core.IRPClassifier

Returns a collection of all the classifier's item flows.

getFlowItems() - Method in interface com.telelogic.rhapsody.core.IRPPackage

Returns a collection of all the item flows in the package.

getFlowPort() - Method in interface com.telelogic.rhapsody.core.IRPMessage

get property flowPort

getFlows() - Method in interface com.telelogic.rhapsody.core.IRPClassifier

Returns a collection of the classifier's flows.

getFlows() - Method in interface com.telelogic.rhapsody.core.IRPPackage

Returns a collection of all the flows in the package.

getFormalClassifier() - Method in interface com.telelogic.rhapsody.core.IRPClassifierRole

get property formalClassifier

getFormalInstance() - Method in interface com.telelogic.rhapsody.core.IRPClassifierRole

get property formalInstance

getFormalInterfaceItem() - Method in interface com.telelogic.rhapsody.core.IRPMessage

get property formalInterfaceItem

getFormalRelations() - Method in interface com.telelogic.rhapsody.core.IRPAssociationRole

Returns a collection of IRPRelation objects, representing the association ends of the association role.

getFormalType() - Method in interface com.telelogic.rhapsody.core.IRPMessage

Returns the model element associated with an action block, condition mark, timeout, or canceled timeout, in a sequence diagram.

getFragmentElement() - Method in interface com.telelogic.rhapsody.core.IRPFileFragment

get property fragmentElement

getFragmentText() - Method in interface com.telelogic.rhapsody.core.IRPFileFragment

get property fragmentText

getFragmentType() - Method in interface com.telelogic.rhapsody.core.IRPFileFragment

get property fragmentType

getFrom() - Method in interface com.telelogic.rhapsody.core.IRPLink

get property from

getFromElement() - Method in interface com.telelogic.rhapsody.core.IRPLink

get property fromElement

getFromElementTypes() - Method in interface com.telelogic.rhapsody.core.IRPMatrixLayout

Returns a collection of the "from" element types specified to be displayed in the matrix.

getFromElementTypes() - Method in interface com.telelogic.rhapsody.core.IRPTableLayout

For "relation tables", returns a collection of the element types specified as the "from" element types.

getFromElementTypesQueryToUse() - Method in interface com.telelogic.rhapsody.core.IRPMatrixLayout

Returns the query that was specified to determine the "from" element types.

getFromElementTypesQueryToUse() - Method in interface com.telelogic.rhapsody.core.IRPTableLayout

For "relation tables", returns the query that was specified to determine the "from" element types.

getFromElementTypesUseQueryOrElementsList() - Method in interface com.telelogic.rhapsody.core.IRPMatrixLayout

Checks whether a query or collection of element types was used to specify the "from" element types.

getFromElementTypesUseQueryOrElementsList() - Method in interface com.telelogic.rhapsody.core.IRPTableLayout

For "relation tables", checks whether a query or collection of element types was used to specify the "from" element types.

getFromPort() - Method in interface com.telelogic.rhapsody.core.IRPLink

get property fromPort

G

getFromProfile() - Method in interface com.telelogic.rhapsody.core.<u>IRPTag</u>
> For tags whose source is a profile that was added to the project (as opposed to tags defined locally in the project), this method returns the profile in which the tag was defined.

getFromScope() - Method in interface com.telelogic.rhapsody.core.<u>IRPMatrixView</u>
> method GetFromScope

getFromSysMLPort() - Method in interface com.telelogic.rhapsody.core.<u>IRPLink</u>
> get property fromSysMLPort

getFullNameInStatechart() - Method in interface com.telelogic.rhapsody.core.<u>IRPState</u>
> Returns the full name of the state within the statechart, including information about its hierarchical position within the statechart.

getFullPathFileName() - Method in interface com.telelogic.rhapsody.core.<u>IRPControlledFile</u>
> Returns the full path of the controlled file.

getFullPathName() - Method in interface com.telelogic.rhapsody.core.<u>IRPModelElement</u>
> Returns the full path name of the model element.

getFullPathNameIn() - Method in interface com.telelogic.rhapsody.core.<u>IRPModelElement</u>
> Retrieves the full path name of the element as a string in the following format: (class) in (package).

getGeneralizations() - Method in interface com.telelogic.rhapsody.core.<u>IRPClassifier</u>
> Returns a collection of all the classifier's generalization relationships.

getGenerateCodeForActors() - Method in interface com.telelogic.rhapsody.core.<u>IRPConfiguration</u>
> get property generateCodeForActors

getGeneratedFileNames(IRPModelElement) - Method in interface
com.telelogic.rhapsody.core.<u>IRPCodeGenerator</u>
> method to get generated code file names

getGlobalFunctions() - Method in interface com.telelogic.rhapsody.core.<u>IRPPackage</u>
> Returns a collection of all the global functions in the package.

getGlobalObjects() - Method in interface com.telelogic.rhapsody.core.<u>IRPPackage</u>
> Returns a collection of all the Objects in the package.

getGlobalVariables() - Method in interface com.telelogic.rhapsody.core.<u>IRPPackage</u>
> Returns a collection of all the global variables in the package.

getGraphicalElements() - Method in interface com.telelogic.rhapsody.core.<u>IRPDiagram</u>
> Returns a collection of all the graphical elements in the diagram.

getGraphicalElements() - Method in interface com.telelogic.rhapsody.core.<u>IRPStatechart</u>
> Returns a collection of all the graphical elements in the statechart.

getGraphicalParent() - Method in interface com.telelogic.rhapsody.core.<u>IRPGraphElement</u>
> get property graphicalParent

getGraphicalProperty(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPGraphElement</u>
> method getGraphicalProperty

getGraphicalPropertyOfText(String, String) - Method in interface
com.telelogic.rhapsody.core.<u>IRPGraphElement</u>
> Returns the specified graphical property for a textual element associated with the graphic element.

getGUID() - Method in interface com.telelogic.rhapsody.core.<u>IRPModelElement</u>
> Returns the GUID of the model element.

getHiddenApplication() - Method in interface com.telelogic.rhapsody.core.<u>IRPRhapsodyServer</u>
> getHiddenApplication

getHTMLContent() - Method in interface com.telelogic.rhapsody.core.<u>IRPMatrixView</u>
> Returns the content of the matrix as HTML.

getHTMLContent() - Method in interface com.telelogic.rhapsody.core.<u>IRPTableView</u>
> Returns the content of the table as HTML.

getHyperLinks() - Method in interface com.telelogic.rhapsody.core.<u>IRPModelElement</u>
> Returns a collection of all the hyperlinks associated with the element.

getIcon() - Method in interface com.telelogic.rhapsody.core.<u>IRPStereotype</u>

Gets the full path for the image file that is associated with this stereotype.

getIconFileName() - Method in interface com.telelogic.rhapsody.core.IRPModelElement

Returns the full path of the graphic file used to represent elements of this type in the browser, for example, D:\programs\rhapsody80\Share\PredefinedPictures\Icons\RhapsodyIcons_72.gif.

getId() - Method in class com.telelogic.rhapsody.core.RPApplicationListener

Gets the id of the listener

getId() - Method in class com.telelogic.rhapsody.core.RPCodeGeneratorListener

Gets the id of the listener

getId() - Method in class com.telelogic.rhapsody.core.RPIntegratorListener

Gets the id of the listener

getId() - Method in class com.telelogic.rhapsody.core.RPJavaPluginsManager

Gets the id of the listener

getId() - Method in class com.telelogic.rhapsody.core.RPRoundTripListener

Gets the id of the listener

getId() - Method in class com.telelogic.rhapsody.core.RPRTCListener

Gets the id of the listener

getImageCollection(String, String, String) - Method in interface com.telelogic.rhapsody.core.IRPMatrixView

method GetImageCollection

getImageCollection(String, String, String) - Method in interface com.telelogic.rhapsody.core.IRPTableView

method GetImageCollection

getImageLayout() - Method in interface com.telelogic.rhapsody.core.IRPGraphElement

Returns the image layout specified for the image linked to the graphic element.

getImplementationSignature() - Method in interface com.telelogic.rhapsody.core.IRPOperation

Returns the signature of the operation as it will appear in the generated code.

getImpName(int) - Method in interface com.telelogic.rhapsody.core.IRPFile

method getImpName

getIncludeDescendants() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Checks whether the scope of the search is to include the descendants of the elements specified for the scope.

getIncludeDescendants() - Method in interface com.telelogic.rhapsody.core.IRPTableView

get property includeDescendants

getIncludeDescendantsFromScope() - Method in interface com.telelogic.rhapsody.core.IRPMatrixView

get property includeDescendantsFromScope

getIncludeDescendantsToScope() - Method in interface com.telelogic.rhapsody.core.IRPMatrixView

get property includeDescendantsToScope

getIncludeInNextLoad() - Method in interface com.telelogic.rhapsody.core.IRPUnit

Checks whether the unit is going to be loaded the next time the model is loaded.

getIncludePath() - Method in interface com.telelogic.rhapsody.core.IRPComponent

Returns the include path defined for the component.

getIncludePath() - Method in interface com.telelogic.rhapsody.core.IRPConfiguration

get property includePath

getInheritsFrom() - Method in interface com.telelogic.rhapsody.core.IRPState

Returns the corresponding state from the statechart of the class that this class is derived from.

getInheritsFrom() - Method in interface com.telelogic.rhapsody.core.IRPStatechart

Returns the statechart of the base class of this class.

getInheritsFrom() - Method in interface com.telelogic.rhapsody.core.IRPTransition

method getInheritsFrom

getIniFileParameterValue(String, String) - Method in interface com.telelogic.rhapsody.core.IRPApplication

G   852

getIniFileParameterValue

getInitialInstances() - Method in interface com.telelogic.rhapsody.core.IRPConfiguration
> get method initialInstances

getInitializationCode() - Method in interface com.telelogic.rhapsody.core.IRPConfiguration
> get property initializationCode

getInitializer() - Method in interface com.telelogic.rhapsody.core.IRPOperation
> For constructors, gets the initializer code that was defined for the operation.

getInLinks() - Method in interface com.telelogic.rhapsody.core.IRPInstance
> method getInLinks

getInstanceSlots() - Method in interface com.telelogic.rhapsody.core.IRPInstanceSpecification
> get property instanceSlots

getInstanceSpecifications() - Method in interface com.telelogic.rhapsody.core.IRPPackage
> Returns a collection of all the instance specifications in the package.

getInstantiatedBy() - Method in interface com.telelogic.rhapsody.core.IRPInstance
> get property instantiatedBy

getInstantiates() - Method in interface com.telelogic.rhapsody.core.IRPLink
> get property instantiates

getInState() - Method in interface com.telelogic.rhapsody.core.IRPObjectNode
> **Deprecated.** *Use getInStateList() instead.*

getInStateList() - Method in interface com.telelogic.rhapsody.core.IRPObjectNode
> Returns a collection of the "In State" states for the object node.

getInstrumentationScope() - Method in interface com.telelogic.rhapsody.core.IRPConfiguration
> get property instrumentationScope

getInstrumentationType() - Method in interface com.telelogic.rhapsody.core.IRPConfiguration
> get property instrumentationType

getInteractionConstraint() - Method in interface com.telelogic.rhapsody.core.IRPInteractionOperand
> Returns the constraint (guard condition) that was defined for the interaction operand.

getInteractionOccurrence() - Method in interface com.telelogic.rhapsody.core.IRPMessagePoint
> get property interactionOccurrence

getInteractionOccurrences() - Method in interface com.telelogic.rhapsody.core.IRPCollaboration
> Returns a collection of all the interaction occurrences in the sequence diagram.

getInteractionOperands() - Method in interface com.telelogic.rhapsody.core.IRPInteractionOperator
> get property interactionOperands

getInteractionOperator() - Method in interface com.telelogic.rhapsody.core.IRPMessagePoint
> get property interactionOperator

getInteractionOperators() - Method in interface com.telelogic.rhapsody.core.IRPCollaboration
> Returns a collection of all the interaction operators in the sequence diagram.

getInteractionType() - Method in interface com.telelogic.rhapsody.core.IRPInteractionOperator
> get property interactionType

getInterfaceItems() - Method in interface com.telelogic.rhapsody.core.IRPClassifier
> Returns a collection of the classifier's elements of type IRPInterfaceItem (such as operations, triggered operations, and event receptions).

getInterfaceItemsIncludingBases() - Method in interface com.telelogic.rhapsody.core.IRPClassifier
> Returns a collection of the classifier's elements of type IRPInterfaceItem (such as operations, triggered operations, and event receptions), including those it inherits from its base classifier.

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPASCIIFile
> get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPAXViewCtrl
> get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPCodeGenerator
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPCodeGenSimplifiersRegistry
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPDiagSynthAPI
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPExternalCheckRegistry
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPExternalCodeGeneratorInvoker
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPExternalIDERegistry
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPExternalRoundtripInvoker
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPGraphElement
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPGraphicalProperty
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPImageMap
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPIntegrator
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPJavaPlugins
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
Returns the name of the API interface corresponding to the current element, for example, IRPClass for a class element, IRPOperation for an operation element.

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPPowListListener
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPPowPaneMgr
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPPowTextListener
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPPlugInWindow
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPProgressBar
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPRhapsodyServer
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPRoundTrip
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPSearchManager
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
Returns the name of the interface (IRPSearchQuery).

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPSearchResult
get property interfaceName

getInterfaceName() - Method in interface com.telelogic.rhapsody.core.IRPSelection
get property interfaceName

getInternalTransitions() - Method in interface com.telelogic.rhapsody.core.IRPState
Returns a collection of the state's internal transitions.

getInTransitions() - Method in interface com.telelogic.rhapsody.core.IRPStateVertex
>   Returns all of the transitions that enter the element.

getInvariant() - Method in interface com.telelogic.rhapsody.core.IRPMessage
>   Gets the text of the Invariant field for the state invariant.

getInverse() - Method in interface com.telelogic.rhapsody.core.IRPRelation
>   get property inverse

getInvokedOperation() - Method in interface com.telelogic.rhapsody.core.IRPSendAction
>   Returns the IRPInterfaceItem element that is invoked by the Send Action element.

getIsAbstract() - Method in interface com.telelogic.rhapsody.core.IRPClass
>   Checks whether the class is an abstract class.

getIsAbstract() - Method in interface com.telelogic.rhapsody.core.IRPOperation
>   Checks whether the operation was defined as abstract.

getIsActive() - Method in interface com.telelogic.rhapsody.core.IRPClass
>   Checks whether the class was defined as "active", meaning that during execution it runs on its own
>   thread.

getIsAnalysisOnly() - Method in interface com.telelogic.rhapsody.core.IRPFlowchart
>   Checks whether the activity is defined as analysis-only, meaning that it is used only for modeling
>   purposes and code is not generated for the activity.

getIsBehavioral() - Method in interface com.telelogic.rhapsody.core.IRPPort
>   get property isBehavioral

getIsBehaviorOverriden() - Method in interface com.telelogic.rhapsody.core.IRPActor
>   Checks whether an actor does not inherit the behavior defined in the statechart of its base class.

getIsBehaviorOverriden() - Method in interface com.telelogic.rhapsody.core.IRPClass
>   Checks whether a class does not inherit the behavior defined in the statechart of its base class.

getIsBehaviorOverriden() - Method in interface com.telelogic.rhapsody.core.IRPUseCase
>   get property isBehaviorOverriden

getIsCgDerived() - Method in interface com.telelogic.rhapsody.core.IRPOperation
>   Checks whether the operation is an operation that is automatically generated by Rhapsody.

getIsClass() - Method in interface com.telelogic.rhapsody.core.IRPAssociationClass
>   Checks whether the element is an association class or an association element.

getIsComposite() - Method in interface com.telelogic.rhapsody.core.IRPClass
>   Checks whether the class is a composite class.

getIsConst() - Method in interface com.telelogic.rhapsody.core.IRPOperation
>   For operations in C++ classes, checks whether the operation was defined as a constant member
>   function.

getIsConstant() - Method in interface com.telelogic.rhapsody.core.IRPAttribute
>   Checks whether the attribute was defined as constant.

getIsCtor() - Method in interface com.telelogic.rhapsody.core.IRPOperation
>   Checks whether the operation is a constructor.

getIsDtor() - Method in interface com.telelogic.rhapsody.core.IRPOperation
>   Checks whether the operation is a destructor.

getIsElaborated() - Method in interface com.telelogic.rhapsody.core.IRPTimingDiagram
>   Checks whether the the timing diagram is an elaborated timing diagram.

getIsExternal() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
>   Checks whether the element is an "external" element - corresponds to the value of the property
>   UseAsExternal.

getIsFinal() - Method in interface com.telelogic.rhapsody.core.IRPClass
>   Checks whether the class is a final class.

getIsFinal() - Method in interface com.telelogic.rhapsody.core.IRPOperation
>   For operations in Java classes, checks whether the operation was defined as final.

getIsGUID() - Method in interface com.telelogic.rhapsody.core.IRPImageMap

get property isGUID

getIsHiddenUI() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> get property isHiddenUI

getIsInline() - Method in interface com.telelogic.rhapsody.core.IRPOperation
> Checks whether the code for the operation will be generated inline.

getIsLoadOnDemand() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> get property isLoadOnDemand

getIsMainBehavior() - Method in interface com.telelogic.rhapsody.core.IRPStatechart
> Checks whether the statechart is the main behavior for the class.

getIsNavigable() - Method in interface com.telelogic.rhapsody.core.IRPRelation
> get property isNavigable

getIsNewTerm() - Method in interface com.telelogic.rhapsody.core.IRPStereotype
> Checks whether the stereotype is a "new term" stereotype.

getIsOfMetaClass(String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Indicates whether the model element is based on the metaclass provided as a parameter.

getIsOrdered() - Method in interface com.telelogic.rhapsody.core.IRPAttribute
> For attributes with multiplicity greater than one, checks whether the order of the items was specified as significant.

getIsOverridden() - Method in interface com.telelogic.rhapsody.core.IRPState
> Checks whether there is still an inheritance relationship between this state and the corresponding state from the statechart of the class that this class is derived from.

getIsOverridden() - Method in interface com.telelogic.rhapsody.core.IRPStatechart
> Checks whether the inheritance relationship between this statechart and the statechart of the base class was overridden.

getIsOverridden() - Method in interface com.telelogic.rhapsody.core.IRPTransition
> get isOverridden

getIsPanelWidget() - Method in interface com.telelogic.rhapsody.core.IRPGraphNode
> get property isPanelWidget

getIsParameter() - Method in interface com.telelogic.rhapsody.core.IRPPin
> Checks whether the element is an activity parameter or an action pin.

getIsPredefined() - Method in interface com.telelogic.rhapsody.core.IRPType
> get property isPredefined

getIsReactive() - Method in interface com.telelogic.rhapsody.core.IRPClass
> Checks whether the class is a reactive class, meaning that a statechart or an activity diagram has been created for the class so that it reacts to events.

getIsReference() - Method in interface com.telelogic.rhapsody.core.IRPAttribute
> Checks whether the attribute was defined as a pointer.

getIsReferenceActivity() - Method in interface com.telelogic.rhapsody.core.IRPState
> Checks whether this element is a call behavior element.

getIsReversed() - Method in interface com.telelogic.rhapsody.core.IRPPort
> get property isReversed

getIsReversed() - Method in interface com.telelogic.rhapsody.core.IRPSysMLPort
> Checks whether the flowport was specified as conjugated.

getIsShowDisplayName() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Checks whether the model element is configured to have its label displayed instead of its name whenever it is included in a diagram.

getIsStatic() - Method in interface com.telelogic.rhapsody.core.IRPAttribute
> Checks whether the attribute was defined as static.

getIsStatic() - Method in interface com.telelogic.rhapsody.core.IRPOperation
> Checks whether the operation was defined as static.

getIsStub() - Method in interface com.telelogic.rhapsody.core.IRPUnit

Checks whether the unit is currently unloaded.

getIsSymmetric() - Method in interface com.telelogic.rhapsody.core.IRPRelation
get property isSymmetric

getIsTrigger() - Method in interface com.telelogic.rhapsody.core.IRPOperation
Checks whether the operation was defined as a triggered operation.

getIsTypedef() - Method in interface com.telelogic.rhapsody.core.IRPType
get property isTypedef

getIsTypedefConstant() - Method in interface com.telelogic.rhapsody.core.IRPType
get property isTypedefConstant

getIsTypedefOrdered() - Method in interface com.telelogic.rhapsody.core.IRPType
get property isTypedefOrdered

getIsTypedefReference() - Method in interface com.telelogic.rhapsody.core.IRPType
get property isTypedefReference

getIsUnresolved() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
Checks if the element is an element that can't be resolved by Rhapsody.

getIsVirtual() - Method in interface com.telelogic.rhapsody.core.IRPGeneralization
get property is virtual

getIsVirtual() - Method in interface com.telelogic.rhapsody.core.IRPOperation
For operations in C++ or C# classes, checks whether the operation was defined as virtual.

getItem(int) - Method in interface com.telelogic.rhapsody.core.IRPCollection
Retrieves an item from a collection, using the index specified.

getItsAction() - Method in interface com.telelogic.rhapsody.core.IRPTransition
method getItsAction

getItsClass() - Method in interface com.telelogic.rhapsody.core.IRPStatechart
Returns the class that the statechart is associated with.

getItsComponent() - Method in interface com.telelogic.rhapsody.core.IRPConfiguration
method getItsComponent

getItsGuard() - Method in interface com.telelogic.rhapsody.core.IRPTransition
method getItsGuard

getItsLabel() - Method in interface com.telelogic.rhapsody.core.IRPTransition
get property itsLabel

getItsMatrixLayout() - Method in interface com.telelogic.rhapsody.core.IRPMatrixView
method GetItsMatrixLayout

getItsOperation() - Method in interface com.telelogic.rhapsody.core.IRPTrigger
method getItsOperation

getItsOwner() - Method in interface com.telelogic.rhapsody.core.IRPFlowchart
Deprecated. *Use IRPModelElement.getOwner instead.*

getItsSource() - Method in interface com.telelogic.rhapsody.core.IRPTransition
get property itsSource

getItsStatechart() - Method in interface com.telelogic.rhapsody.core.IRPState
Returns the statechart that this state belongs to.

getItsStatechart() - Method in interface com.telelogic.rhapsody.core.IRPTransition
get property itsStatechart

getItsSwimlane() - Method in interface com.telelogic.rhapsody.core.IRPConnector
For connectors in a swimlane, returns the swimlane that contains the connector.

getItsSwimlane() - Method in interface com.telelogic.rhapsody.core.IRPState
Returns the swimlane that the action is located in.

getItsTableLayout() - Method in interface com.telelogic.rhapsody.core.IRPTableView
method GetItsTableLayout

getItsTarget() - Method in interface com.telelogic.rhapsody.core.IRPTransition
get property itsTarget

getItsTrigger() - Method in interface com.telelogic.rhapsody.core.IRPTransition
>   method getItsTrigger

getJavaPluginManager() - Static method in class com.telelogic.rhapsody.core.RhapsodyAppServer
>   For internal use only.

getKey() - Method in interface com.telelogic.rhapsody.core.IRPGraphicalProperty
>   get property key

getKind() - Method in interface com.telelogic.rhapsody.core.IRPType
>   get property kind

getLanguage() - Method in interface com.telelogic.rhapsody.core.IRPApplication
>   get property Language

getLanguage() - Method in interface com.telelogic.rhapsody.core.IRPUnit
>   Gets the language of the unit.

getLastModifiedTime() - Method in interface com.telelogic.rhapsody.core.IRPUnit
>   Returns the time at which the file representing the unit was last modified.

getLastVisualizationModifiedTime() - Method in interface com.telelogic.rhapsody.core.IRPDiagram
>   Returns the time at which the visual representation of the diagram was last changed.

getLibraries() - Method in interface com.telelogic.rhapsody.core.IRPComponent
>   get property libraries

getLibraries() - Method in interface com.telelogic.rhapsody.core.IRPConfiguration
>   get property libraries

getLinks() - Method in interface com.telelogic.rhapsody.core.IRPClassifier
>   Returns a collection of all the classifier's link relationships.

getLinks() - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Returns a collection of all the Links in the package.

getLinkSwitches() - Method in interface com.telelogic.rhapsody.core.IRPConfiguration
>   get property linkSwitches

getListOfFactoryProperties() - Method in interface com.telelogic.rhapsody.core.IRPApplication
>   get list of factory properties

getListOfInitializerArguments() - Method in interface com.telelogic.rhapsody.core.IRPInstance
>   method getListOfInitializerArguments

getListOfSelectedElements() - Method in interface com.telelogic.rhapsody.core.IRPApplication
>   get list of selected elements

getListOfSiteProperties() - Method in interface com.telelogic.rhapsody.core.IRPApplication
>   get list of site properties

getLocaleName() - Method in interface com.telelogic.rhapsody.core.IRPApplication
>   Returns the locale for the version of Rhapsody running.

getLocalProperties() - Method in interface com.telelogic.rhapsody.core.IRPGraphElement
>   method getLocalProperties

getLocalTags() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
>   Returns a collection of the tags that were created locally for this model element.

getLogicalCollaboration() - Method in interface com.telelogic.rhapsody.core.IRPCollaborationDiagram
>   Returns the IRPCollaboration object underlying the collaboration diagram.

getLogicalCollaboration() - Method in interface com.telelogic.rhapsody.core.IRPSequenceDiagram
>   Returns the IRPCollaboration object underlying the sequence diagram.

getLogicalStates() - Method in interface com.telelogic.rhapsody.core.IRPState
>   Returns a collection of all the substates of the current state and all the first-level substates of those
>   states, meaning down to the second level.

getMainDiagram() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
>   Returns the "main" diagram for the element.

getMainFileName(IRPModelElement, int, int) - Method in class
com.telelogic.rhapsody.core.RPExternalCodeGenerator

G   858

method GetMainFileName

getMainName() - Method in interface com.telelogic.rhapsody.core.IRPConfiguration

method getMainName

getMakefileName(int) - Method in interface com.telelogic.rhapsody.core.IRPConfiguration

method getMakefileName

getMakefileName(IRPModelElement, int, int) - Method in class
com.telelogic.rhapsody.core.RPExternalCodeGenerator

method GetMakefileName

getMatchCase() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Checks whether an exact match was specified for the query in terms of upper and lower case.

getMatchedField() - Method in interface com.telelogic.rhapsody.core.IRPSearchResult

get property matchedField

getMatchedFields() - Method in interface com.telelogic.rhapsody.core.IRPSearchResult

get property matchedFields

getMatchedObject() - Method in interface com.telelogic.rhapsody.core.IRPSearchResult

get property matchedObject

getMatchSpecifiedCriteria() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Checks whether the query is to return the model elements that match the criteria specified, or the
model elements that do not match the criteria specified.

getMatchWholeWord() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery

Checks whether a whole word match was specified for the search.

getMessage() - Method in interface com.telelogic.rhapsody.core.IRPExecutionOccurrence

get property message

getMessage() - Method in interface com.telelogic.rhapsody.core.IRPMessagePoint

get property message

getMessagePoints(IRPClassifierRole) - Method in interface com.telelogic.rhapsody.core.IRPCollaboration

Returns all the message points along the specified instance line.

getMessagePoints() - Method in interface com.telelogic.rhapsody.core.IRPCollaboration

Returns all the message points along the specified instance line.

getMessagePoints() - Method in interface com.telelogic.rhapsody.core.IRPInteractionOccurrence

get property messagePoints

getMessages() - Method in interface com.telelogic.rhapsody.core.IRPCollaboration

Returns a collection of all the messages in the sequence diagram.

getMessageType() - Method in interface com.telelogic.rhapsody.core.IRPMessage

get property messageType

getMetaClass() - Method in interface com.telelogic.rhapsody.core.IRPModelElement

Gets the name of the metaclass on which the model element is based.

getMetaclasses() - Method in class com.telelogic.rhapsody.core.RPExternalCheck

Return a comma separated list of metaclasses or new terms - Rhapsody will call check for all
elements in scope of check that are of the metaclass type in the list

getMode() - Method in interface com.telelogic.rhapsody.core.IRPCollaboration

get property mode

getModelElementFileName(IRPModelElement, int, int) - Method in interface
com.telelogic.rhapsody.core.IRPComponent

method getModelElementFileName

getModelElementFromSource(String, int, int) - Method in interface
com.telelogic.rhapsody.core.IRPApplication

Find model element from source code

getModelObject() - Method in interface com.telelogic.rhapsody.core.IRPGraphElement

get property modelObject

getModules() - Method in interface com.telelogic.rhapsody.core.IRPPackage

Returns a collection of all the File elements in the package.

getMultiplicities() - Method in interface com.telelogic.rhapsody.core.IRPContextSpecification
    Returns the collection of the relevant indices for each of the model elements in the "value" collection.

getMultiplicity() - Method in interface com.telelogic.rhapsody.core.IRPAttribute
    Gets the multiplicity specified for the attribute.

getMultiplicity() - Method in interface com.telelogic.rhapsody.core.IRPRelation
    get property multiplicity

getMultiplicity() - Method in interface com.telelogic.rhapsody.core.IRPTag
    Returns the multiplicity that was specified for the tag.

getName() - Method in interface com.telelogic.rhapsody.core.IRPImageMap
    get property name

getName() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
    Returns the name of the element.

getName() - Method in interface com.telelogic.rhapsody.core.IRPSearchResult
    get property name

getName() - Method in class com.telelogic.rhapsody.core.RPExternalCheck
    Return the name of the check (also used as its error message)

getNamespace() - Method in interface com.telelogic.rhapsody.core.IRPPackage
    getNamespace

getNestedClassifiers() - Method in interface com.telelogic.rhapsody.core.IRPClassifier
    Returns a collection of all the classifiers nested below the current classifier.

getNestedClassifiers() - Method in interface com.telelogic.rhapsody.core.IRPPackage
    Returns a collection of all the classifiers in the package.

getNestedComponents() - Method in interface com.telelogic.rhapsody.core.IRPComponent
    Returns a collection of all the nested components in the component.

getNestedComponents() - Method in interface com.telelogic.rhapsody.core.IRPPackage
    Returns a collection of all the Components in the package.

getNestedElements() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
    Gets a collection of all the model elements that are directly under the current element.

getNestedElementsByMetaClass(String, int) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
    Retrieves all of the model elements of the specified type below the current element.

getNestedElementsRecursive() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
    Returns a collection that consists of the current element and all of the model elements below it.

getNestedSaveUnits() - Method in interface com.telelogic.rhapsody.core.IRPUnit
    Returns a collection of any sub-elements of the unit that were saved as individual files.

getNestedSaveUnitsCount() - Method in interface com.telelogic.rhapsody.core.IRPUnit
    Returns the number of sub-elements of the unit that were saved as individual files.

getNestedStatechart() - Method in interface com.telelogic.rhapsody.core.IRPState
    Returns the state's sub-statechart.

getNewCollaboration() - Method in interface com.telelogic.rhapsody.core.IRPProject
    Creates a new IRPCollaboration object that can be used to create a sequence diagram.

getNewProgressBar(int, String) - Method in interface com.telelogic.rhapsody.core.IRPProject
    method getNewProgressBar

getNewTermStereotype() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
    If a "new term" stereotype has been applied to the element, returns the stereotype.

getNode() - Method in interface com.telelogic.rhapsody.core.IRPComponentInstance
    get property node

getNodes() - Method in interface com.telelogic.rhapsody.core.IRPPackage
    Returns a collection of all the Node elements in the package.

getNotifyPluginOnElementsChanged() - Method in interface com.telelogic.rhapsody.core.IRPProject

G   860

Checks whether plugins will be notified when model elements are modified.

getObjectAsObjectType() - Method in interface com.telelogic.rhapsody.core.IRPRelation
  get property ObjectAsObjectType

getObjectModelDiagrams() - Method in interface com.telelogic.rhapsody.core.IRPPackage
  Returns a collection of all the object model diagrams in the package.

getOfClass() - Method in interface com.telelogic.rhapsody.core.IRPRelation
  get property ofClass

getOfMetaClass() - Method in interface com.telelogic.rhapsody.core.IRPStereotype
  Gets the names of the metaclasses that the stereotype can be applied to.

getOfState() - Method in interface com.telelogic.rhapsody.core.IRPConnector
  For history connectors, returns the state that the history connector belongs to.

getOfState() - Method in interface com.telelogic.rhapsody.core.IRPTransition
  method getOfState

getOfTemplate() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
  If the element is an instantiation of a template, this method returns the template that it instantiates.

getOMROOT() - Method in interface com.telelogic.rhapsody.core.IRPApplication
  get property OMROOT

getOperation() - Method in interface com.telelogic.rhapsody.core.IRPCallOperation
  Returns the operation specified for this call operation element.

getOperations() - Method in interface com.telelogic.rhapsody.core.IRPClassifier
  Returns a collection of all the classifier's operations.

getOSLCLinks() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
  Returns a collection of all the element's OSLC links.

getOther() - Method in interface com.telelogic.rhapsody.core.IRPLink
  get property other

getOtherClass() - Method in interface com.telelogic.rhapsody.core.IRPRelation
  Gets the class that this class is related to via this relation.

getOutLinks() - Method in interface com.telelogic.rhapsody.core.IRPInstance
  method getOutLinks

getOutputWindowText() - Method in interface com.telelogic.rhapsody.core.IRPApplication
  Returns the text displayed in the output window.

getOutTransitions() - Method in interface com.telelogic.rhapsody.core.IRPStateVertex
  Returns all of the transitions that exit the element.

getOverlayIconFileName() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
  Returns the full path of the graphic file that is used as an overlay on this specific model element, on top of the regular icon that represent elements of this type in the browser.

getOverriddenProperties(int) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
  Returns a collection of all the properties whose value was overridden for this model element.

getOverriddenPropertiesByPattern(String, int, int) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
  method getOverriddenPropertiesByPattern

getOWListListener(String) - Method in interface com.telelogic.rhapsody.core.IRPowPaneMgr
  get list listener

getOwnedDependencies() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
  Returns all of the dependencies that are owned by the element.

getOwner() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
  Returns the model element that owns this model element.

getOWPaneMgr(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
  For internal use only.

getOWTextListener(String) - Method in interface com.telelogic.rhapsody.core.IRPowPaneMgr
  get text listener

getPackageFile(IRPPackage, int) - Method in interface com.telelogic.rhapsody.core.IRPComponent
>   method getPackageFile

getPackages() - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Returns a collection of all the nested packages in the package.

getPanelDiagrams() - Method in interface com.telelogic.rhapsody.core.IRPComponent
>   Returns a collection of all the panel diagrams in the component.

getPanelDiagrams() - Method in interface com.telelogic.rhapsody.core.IRPPackage
>   Returns a collection of all the panel diagrams in the package.

getPanelWidgetInstancePath() - Method in interface com.telelogic.rhapsody.core.IRPGraphNode
>   get property panelWidgetInstancePath

getParameterKind() - Method in interface com.telelogic.rhapsody.core.IRPTemplateParameter
>   Returns the type of the template parameter.

getParent() - Method in interface com.telelogic.rhapsody.core.IRPStateVertex
>   Returns the element's parent.

getPath(int) - Method in interface com.telelogic.rhapsody.core.IRPComponent
>   get property path

getPath(int) - Method in interface com.telelogic.rhapsody.core.IRPConfiguration
>   get property path

getPath(int) - Method in interface com.telelogic.rhapsody.core.IRPFile
>   get property path

getPicture(String) - Method in interface com.telelogic.rhapsody.core.IRPDiagram
>   Saves the diagram as an emf format file, using the path and filename provided as a parameter.

getPicture(String) - Method in interface com.telelogic.rhapsody.core.IRPStatechart
>   Saves the statechart as an emf format file, using the path and filename provided as a parameter.

getPictureAs(String, String, int, IRPCollection) - Method in interface
com.telelogic.rhapsody.core.IRPDiagram
>   Saves the diagram in the specified graphic format, breaking the diagram into a number of files if
>   necessary.

getPictureAs(String, String, int, IRPCollection) - Method in interface
com.telelogic.rhapsody.core.IRPStatechart
>   Saves the statechart in the specified graphic format, breaking the diagram into a number of files if
>   necessary.

getPictureAsDividedMetafiles(String) - Method in interface com.telelogic.rhapsody.core.IRPDiagram
>   Saves the diagram as an emf format file, breaking the diagram into a number of such files if
>   necessary.

getPictureAsDividedMetafiles(String) - Method in interface com.telelogic.rhapsody.core.IRPStatechart
>   Saves the statechart as an emf format file, breaking the diagram into a number of such files if
>   necessary.

getPictureEx(String, String, int) - Method in interface com.telelogic.rhapsody.core.IRPDiagram
>   method getPictureEx

getPictureFileName() - Method in interface com.telelogic.rhapsody.core.IRPImageMap
>   get property pictureFileName

getPicturesWithImageMap(String, IRPCollection) - Method in interface
com.telelogic.rhapsody.core.IRPDiagram
>   Saves the diagram as an emf format file, breaking the diagram into a number of files if necessary.

getPicturesWithImageMap(String, IRPCollection) - Method in interface
com.telelogic.rhapsody.core.IRPStatechart
>   Saves the statechart as an emf format file, breaking the diagram into a number of files if necessary.

getPinDirection() - Method in interface com.telelogic.rhapsody.core.IRPPin
>   Returns the direction of the pin/parameter: In, Out, or InOut.

getPinType() - Method in interface com.telelogic.rhapsody.core.IRPPin

Returns the type of the value held by the pin/parameter.

getPlugInWindow(int, int, int) - Method in interface com.telelogic.rhapsody.core.IRPApplication
      PlugIn window factory

getPoints() - Method in interface com.telelogic.rhapsody.core.IRPImageMap
      get property points

getPort() - Method in interface com.telelogic.rhapsody.core.IRPMessage
      get property Port

getPortDirection() - Method in interface com.telelogic.rhapsody.core.IRPSysMLPort
      Returns the direction that was specified for the flowport.

getPorts() - Method in interface com.telelogic.rhapsody.core.IRPClassifier
      Returns a collection of all the classifier's ports.

getPossibleVariants(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPComponent
      method getPossibleVariants

getPosString() - Method in interface com.telelogic.rhapsody.core.IRPPlugInWindow
      Get position string

getPredecessor(IRPMessage) - Method in interface com.telelogic.rhapsody.core.IRPCollaboration
      Returns the message that precedes the specified message.

getProfiles() - Method in interface com.telelogic.rhapsody.core.IRPProject
      Returns a collection of all the profiles in the project.

getProject() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
      Returns the project that the current element belongs to.

getProjects() - Method in interface com.telelogic.rhapsody.core.IRPApplication
      get property projects

getPropertyValue(String) - Method in interface com.telelogic.rhapsody.core.IRPGraphElement
      method getPropertyValue

getPropertyValue(String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
      Returns the value of the specified property for the model element.

getPropertyValueConditional(String, IRPCollection, IRPCollection) - Method in interface
com.telelogic.rhapsody.core.IRPModelElement
      Returns the value of the specified property for the model element, taking into account the collection
      of tokens specified and the collection of token values specified.

getPropertyValueConditionalExplicit(String, IRPCollection, IRPCollection) - Method in interface
com.telelogic.rhapsody.core.IRPModelElement
      Returns the value of the specified property for the model element, if the default value was overridden,
      taking into account the collection of tokens specified and the collection of token values specified.

getPropertyValueExplicit(String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
      Returns the value of the specified property for the model element if the default value was overridden.

getProvidedInterfaces() - Method in interface com.telelogic.rhapsody.core.IRPPort
      get property providedInterfaces

getQualifier() - Method in interface com.telelogic.rhapsody.core.IRPRelation
      get property qualifier

getQualifiers() - Method in interface com.telelogic.rhapsody.core.IRPRelation
      method getQualifiers

getQualifierType() - Method in interface com.telelogic.rhapsody.core.IRPRelation
      For associations that use qualifiers, returns the type of the qualifier.

getRedefines() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
      method getRedefines

getReferencedSequenceDiagram() - Method in interface com.telelogic.rhapsody.core.IRPClassifierRole
      get property referencedSequenceDiagram

getReferences() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
      Returns a collection of all the model elements that point to this model element.

G

getReferenceSequenceDiagram() - Method in interface
com.telelogic.rhapsody.core.IRPInteractionOccurrence
       get property referenceSequenceDiagram

getReferenceToActivity() - Method in interface com.telelogic.rhapsody.core.IRPState
       For call behavior elements, returns the activity that is referenced.

getReferencingClassifierRolesRecursively() - Method in interface
com.telelogic.rhapsody.core.IRPClassifierRole
       method getReferencingClassifierRolesRecursively

getRelatedUseCases() - Method in interface com.telelogic.rhapsody.core.IRPSequenceDiagram
       For internal use only.

getRelationLabel() - Method in interface com.telelogic.rhapsody.core.IRPRelation
       get property relationLabel

getRelationLinkName() - Method in interface com.telelogic.rhapsody.core.IRPRelation
       get property relationLinkName

getRelationRoleName() - Method in interface com.telelogic.rhapsody.core.IRPRelation
       get property relationRoleName

getRelations() - Method in interface com.telelogic.rhapsody.core.IRPClassifier
       Returns a collection of all the classifier's associations.

getRelationsIncludingBases() - Method in interface com.telelogic.rhapsody.core.IRPClassifier
       Returns a collection of all the classifier's associations, including those it inherits from its base
       classifier.

getRelationTable() - Method in interface com.telelogic.rhapsody.core.IRPTableLayout
       Checks whether the table was defined as a "relation table".

getRelationType() - Method in interface com.telelogic.rhapsody.core.IRPRelation
       get property relationType

getRemoteDependencies() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
       For Design Manager projects, returns a collection of all the dependencies that the model element has
       on remote elements.

getRemoteRequirementsPopulateMode() - Method in interface com.telelogic.rhapsody.core.IRPPackage
       Returns the mode that was selected for loading remote requirements in the collection.

getRemoteResourcePackages() - Method in interface com.telelogic.rhapsody.core.IRPProject
       For Design Manager projects, returns the packages of remote requirements (the "Remote Resource
       Packages").

getRemoteURI() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
       For elements that are remote resources, returns the URI of the resource.

getRepresentative() - Method in interface com.telelogic.rhapsody.core.IRPTemplateParameter
       For internal use only.

getRepresented() - Method in interface com.telelogic.rhapsody.core.IRPFlowItem
       Returns a collection of all the information elements that are represented by the item flow.

getRepresents() - Method in interface com.telelogic.rhapsody.core.IRPObjectNode
       Returns the class/type that this object node represents.

getRepresents() - Method in interface com.telelogic.rhapsody.core.IRPSwimlane
       get property represents

getRequiredInterfaces() - Method in interface com.telelogic.rhapsody.core.IRPPort
       get property requiredInterfaces

getRequirementID() - Method in interface com.telelogic.rhapsody.core.IRPRequirement
       Returns the ID that was set for the requirement.

getRequirementsByID(String, int) - Method in interface com.telelogic.rhapsody.core.IRPProject
       Returns all of the requirements that have the specified ID.

getRequirementTraceabilityHandle() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
       Returns the ID used by DOORS to refer to this requirement.

getResultList(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout
  method GetResultList
getReturns() - Method in interface com.telelogic.rhapsody.core.IRPOperation
  Gets the return type of the operation.
getReturnTypeDeclaration() - Method in interface com.telelogic.rhapsody.core.IRPOperation
  If an on-the-fly type is used as the return type of an operation, this method returns the declaration for the type.
getReturnValue() - Method in interface com.telelogic.rhapsody.core.IRPMessage
  get property returnValue
getRhapsodyHandleErrorFunction() - Method in interface com.telelogic.rhapsody.core.IRPApplication
  getRhapsodyHandleErrorFunction
getRhapsodyHandleErrorFunctionLong() - Method in interface com.telelogic.rhapsody.core.IRPApplication
  getRhapsodyHandleErrorFunctionLong
getRoleType() - Method in interface com.telelogic.rhapsody.core.IRPAssociationRole
  For internal use only.
getRoleType() - Method in interface com.telelogic.rhapsody.core.IRPClassifierRole
  get property roleType
getRootState() - Method in interface com.telelogic.rhapsody.core.IRPStatechart
  Returns the root state of the statechart.
getRoundtripShadowModel(int) - Method in interface com.telelogic.rhapsody.core.IRPProject
  For internal use only.
getSavedInSeperateDirectory() - Method in interface com.telelogic.rhapsody.core.IRPPackage
  Checks whether the package is configured to be saved in a separate directory.
getSaveUnit() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
  Returns the unit that the model element is saved in.
getScope() - Method in interface com.telelogic.rhapsody.core.IRPTableView
  method GetScope
getScopeBySelectedElements() - Method in interface com.telelogic.rhapsody.core.IRPComponent
  Checks whether the scope of the component has been set to include all elements or only specific elements.
getScopeElements() - Method in interface com.telelogic.rhapsody.core.IRPComponent
  Returns a collection of all the model elements in the scope of the component.
getScopeElementsByCategory(String) - Method in interface com.telelogic.rhapsody.core.IRPComponent
  method getScopeElementsByCategory
getScopeType() - Method in interface com.telelogic.rhapsody.core.IRPConfiguration
  get property scopeType
getSearchFindAsOption() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
  Returns the type of search that was specified for the search text - regular text, wildcard, regular expression, or empty string.
getSearchManager() - Method in interface com.telelogic.rhapsody.core.IRPApplication
  get Rhapsody search manager
getSearchScopeElements() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
  Returns a collection of the model elements that constitute the scope for the search.
getSearchScopeObject() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
  Deprecated. *This method, used to return the scope specified for the search, was introduced when Rhapsody only allowed you to specify a single element as the scope. Now that Rhapsody allows you to specify a list of such elements, you should use the method* `IRPSearchQuery.getSearchScopeElements()`*.*
getSearchText() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
  Returns the text that was specified as the text to search for.

Returns the standard headers defined for the component.

getStandardHeaders() - Method in interface com.telelogic.rhapsody.core.IRPConfiguration
> get property standardHeaders

getStatechart() - Method in interface com.telelogic.rhapsody.core.IRPClassifier
> This method should no longer be used because Rational Rhapsody now allows you to define more than one statechart and activity diagram for a class.

getStatechart() - Method in interface com.telelogic.rhapsody.core.IRPStatechartDiagram
> Returns the IRPStatechart object underlying the statechart.

getStatechartDiagram() - Method in interface com.telelogic.rhapsody.core.IRPStatechart
> Returns the IRPStatechartDiagram object associated with the statechart.

getStatechartImplementation() - Method in interface com.telelogic.rhapsody.core.IRPConfiguration
> get property statechartImplementation

getStateType() - Method in interface com.telelogic.rhapsody.core.IRPState
> Returns the type of the state, for example, an And state or a Termination state.

getStaticReactions() - Method in interface com.telelogic.rhapsody.core.IRPState
> Returns a collection of the state's internal transitions.

getStereotype() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> **Deprecated.** *Since Rhapsody now allows multiple stereotypes to be applied to a model element, the getStereotypes() method should be used instead.*

getStereotypes() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Returns a collection of the stereotypes that have been applied to the element.

getStructureDiagrams() - Method in interface com.telelogic.rhapsody.core.IRPUnit
> Returns a collection of any structure diagrams that are sub-elements of the unit.

getSubStates() - Method in interface com.telelogic.rhapsody.core.IRPState
> Returns a collection of the substates contained in this state.

getSubStateVertices() - Method in interface com.telelogic.rhapsody.core.IRPState
> Returns a collection of all the first-level elements contained in this state - this includes both node elements and connector elements.

getSuccessor(IRPMessage) - Method in interface com.telelogic.rhapsody.core.IRPCollaboration
> Returns the message that follows the specified message.

getSuperEvent() - Method in interface com.telelogic.rhapsody.core.IRPEvent
> get property baseEvent

getSwimlanes() - Method in interface com.telelogic.rhapsody.core.IRPFlowchart
> Returns a collection of all the swimlanes in the activity.

getSwimlanes() - Method in interface com.telelogic.rhapsody.core.IRPSwimlane
> get property swimlanes

getTag(String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Returns the tag specified.

getTagMetaClass() - Method in interface com.telelogic.rhapsody.core.IRPTag
> Returns the name of the metaclass to which the tag is applicable.

getTarget() - Method in interface com.telelogic.rhapsody.core.IRPCallOperation
> Returns the target specified for this call operation element.

getTarget() - Method in interface com.telelogic.rhapsody.core.IRPGraphEdge
> get property target

getTarget() - Method in interface com.telelogic.rhapsody.core.IRPHyperLink
> Returns the target model element if the hyperlink points to a model element.

getTarget() - Method in interface com.telelogic.rhapsody.core.IRPImageMap
> get property target

getTarget() - Method in interface com.telelogic.rhapsody.core.IRPMessage
> get property target

getTarget() - Method in interface com.telelogic.rhapsody.core.IRPSendAction

Gets the event target of the Send Action element.

getTargetExecutionOccurrence() - Method in interface com.telelogic.rhapsody.core.<u>IRPMessage</u>
        get property targetExecutionOccurrence

getTargetfileName(IRPModelElement, int, int) - Method in class
com.telelogic.rhapsody.core.<u>RPExternalCodeGenerator</u>
        method GetTargetfileName

getTargetName(int) - Method in interface com.telelogic.rhapsody.core.<u>IRPConfiguration</u>
        method getTargetName

getTemplateInstantiationParameters() - Method in interface
com.telelogic.rhapsody.core.<u>IRPTemplateInstantiation</u>
        get property templateInstantiationParameters

getTemplateParameters() - Method in interface com.telelogic.rhapsody.core.<u>IRPModelElement</u>
        For model elements that are templates, returns the template parameters.

getTheCodeGeneratorInterface() - Method in interface com.telelogic.rhapsody.core.<u>IRPApplication</u>
        get codegeneration interface

getTheEntryAction() - Method in interface com.telelogic.rhapsody.core.<u>IRPState</u>
        method getTheEntryAction

getTheExitAction() - Method in interface com.telelogic.rhapsody.core.<u>IRPState</u>
        method getTheExitAction

getTheExternalCodeGeneratorInvoker() - Method in interface com.telelogic.rhapsody.core.<u>IRPApplication</u>
        get external code generator invoker

getTheIntegratorInterface() - Method in interface com.telelogic.rhapsody.core.<u>IRPApplication</u>
        get integrator interface

getTheJavaPluginsInterface() - Method in interface com.telelogic.rhapsody.core.<u>IRPApplication</u>
        getTheJavaPluginsInterface

getTheRoundtripInterface() - Method in interface com.telelogic.rhapsody.core.<u>IRPApplication</u>
        get roundtrip interface

getTi() - Method in interface com.telelogic.rhapsody.core.<u>IRPModelElement</u>
        For model elements that are template instantiations, returns an object that contains the template
        instantiation parameters.

getTimeConstraint() - Method in interface com.telelogic.rhapsody.core.<u>IRPMessage</u>
        Gets the text for the Time Constraint that was applied to this state variant.

getTimeModel() - Method in interface com.telelogic.rhapsody.core.<u>IRPConfiguration</u>
        get property timeModel

getTimeObservation() - Method in interface com.telelogic.rhapsody.core.<u>IRPMessage</u>
        Gets the text of the Time Observation.

getTimerValue() - Method in interface com.telelogic.rhapsody.core.<u>IRPMessage</u>
        get property timerValue

getTimingDiagrams() - Method in interface com.telelogic.rhapsody.core.<u>IRPPackage</u>
        Returns a collection of all the timing diagrams in the package.

getTo() - Method in interface com.telelogic.rhapsody.core.<u>IRPLink</u>
        Returns the target of a link.

getToElement() - Method in interface com.telelogic.rhapsody.core.<u>IRPLink</u>
        get property toElement

getToElementTypes() - Method in interface com.telelogic.rhapsody.core.<u>IRPMatrixLayout</u>
        Returns a collection of the "to" element types specified to be displayed in the matrix.

getToElementTypes() - Method in interface com.telelogic.rhapsody.core.<u>IRPTableLayout</u>
        For "relation tables", returns a collection of the element types specified as the "to" element types.

getToElementTypesQueryToUse() - Method in interface com.telelogic.rhapsody.core.<u>IRPMatrixLayout</u>
        Returns the query that was specified to determine the "to" element types.

getToElementTypesQueryToUse() - Method in interface com.telelogic.rhapsody.core.<u>IRPTableLayout</u>

For "relation tables", returns the query that was specified to determine the "to" element types.

getToElementTypesUseQueryOrElementsList() - Method in interface com.telelogic.rhapsody.core.IRPMatrixLayout

Checks whether a query or collection of element types was used to specify the "to" element types.

getToElementTypesUseQueryOrElementsList() - Method in interface com.telelogic.rhapsody.core.IRPTableLayout

For "relation tables", checks whether a query or collection of element types was used to specify the "to" element types.

getToolSet() - Method in interface com.telelogic.rhapsody.core.IRPApplication

get property ToolSet

getToolTipHTML() - Method in interface com.telelogic.rhapsody.core.IRPModelElement

Returns the HTML that would be used to display the tooltip for the element in the user interface.

getToPort() - Method in interface com.telelogic.rhapsody.core.IRPLink

Returns the port through which a link reaches a target object.

getToScope() - Method in interface com.telelogic.rhapsody.core.IRPMatrixView

method GetToScope

getToSysMLPort() - Method in interface com.telelogic.rhapsody.core.IRPLink

get property toSysMLPort

getType() - Method in interface com.telelogic.rhapsody.core.IRPMessagePoint

get property type

getType() - Method in interface com.telelogic.rhapsody.core.IRPSysMLPort

Returns the type that was specified for the flowport.

getType() - Method in interface com.telelogic.rhapsody.core.IRPTemplateInstantiationParameter

get property type

getType() - Method in interface com.telelogic.rhapsody.core.IRPVariable

Returns the type of the variable.

getTypedefBaseType() - Method in interface com.telelogic.rhapsody.core.IRPType

get property typedefBaseType

getTypedefMultiplicity() - Method in interface com.telelogic.rhapsody.core.IRPType

get property typedefMultiplicity

getTypes() - Method in interface com.telelogic.rhapsody.core.IRPPackage

Returns a collection of all the types in the package.

getUninitializedApplication() - Method in interface com.telelogic.rhapsody.core.IRPRhapsodyServer

getUninitializedApplication

getURL() - Method in interface com.telelogic.rhapsody.core.IRPHyperLink

Returns the target URL if the hyperlink points to a URL.

getUseCaseDiagrams() - Method in interface com.telelogic.rhapsody.core.IRPPackage

Returns a collection of all the use case diagrams in the package.

getUseCases() - Method in interface com.telelogic.rhapsody.core.IRPPackage

Returns a collection of all the use cases in the package.

getUserDefinedMetaClass() - Method in interface com.telelogic.rhapsody.core.IRPModelElement

Gets the name of the New Term on which the model element is based.

getUserDefinedStereotypes() - Method in interface com.telelogic.rhapsody.core.IRPPackage

Returns a collection of all the user-defined stereotypes in the package.

getValue() - Method in interface com.telelogic.rhapsody.core.IRPContextSpecification

Returns the collection of strings that represents the model elements that constitute the full path to the element.

getValue() - Method in interface com.telelogic.rhapsody.core.IRPEnumerationLiteral

get property value

getValue() - Method in interface com.telelogic.rhapsody.core.IRPGraphicalProperty

get property value

G

getValue() - Method in interface com.telelogic.rhapsody.core.IRPInstanceValue
> Returns the stored value.

getValue() - Method in interface com.telelogic.rhapsody.core.IRPLiteralSpecification
> Returns the stored value.

getValue() - Method in interface com.telelogic.rhapsody.core.IRPTag
> Returns the value of the tag.

getValues() - Method in interface com.telelogic.rhapsody.core.IRPInstanceSlot
> get property values

getValueSpecifications() - Method in interface com.telelogic.rhapsody.core.IRPVariable
> Returns a collection of the initial values that were declared for elements where the multiplicity is greater than one.

getVariant(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPComponent
> method getVariant

getVariationPoints() - Method in interface com.telelogic.rhapsody.core.IRPComponent
> Returns a collection of the variation points that are included in the scope of the component.

getView(int) - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
> Retrieves the specified item from the list of tables, matrices, and diagrams that are to be searched.

getViewCount() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
> Returns the number of views in the list of views that are to be searched.

getViewIncludeModelElements() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
> Checks whether the query specifies that the search results should also include model elements that were found by the search but are not referenced in any of the views that you specified.

getViewsToSearch() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
> Returns indication of which views (diagrams, tables, and matrices) are supposed to be searched.

getVisibility() - Method in interface com.telelogic.rhapsody.core.IRPAttribute
> Gets the visibility specified for the attribute.

getVisibility() - Method in interface com.telelogic.rhapsody.core.IRPGeneralization
> get property visibility

getVisibility() - Method in interface com.telelogic.rhapsody.core.IRPOperation
> Gets the visibility specified for the operation.

getVisibility() - Method in interface com.telelogic.rhapsody.core.IRPRelation
> get property visibility

getWindowHandle() - Method in interface com.telelogic.rhapsody.core.IRPPlugInWindow
> Get window handle

GROUP_ALL - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

GROUP_CODE - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

GROUP_ELEMENT_NAME - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

GROUP_OTHER_TEXT - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

---

Package Class Use **Tree** **Serialized** **Deprecated** Index **Help**
PREV LETTER **NEXT LETTER**   **FRAMES** **NO FRAMES** **All Classes**
A B C D E F G H I L M N O P Q R S T U V W X

G   870

H

hasNestedElements() - Method in interface com.telelogic.rhapsody.core.IRPModelElement

Checks whether the model element contains other elements.

hasPanelWidget() - Method in interface com.telelogic.rhapsody.core.IRPModelElement

Checks whether the model element is bound to a panel diagram widget.

hasPluginWithMethod(String, String) - Method in class com.telelogic.rhapsody.core.RPJavaPluginsManager

Check if a method exists on a plugin

hasPluginWithMethodArgs(String, String, String) - Method in class com.telelogic.rhapsody.core.RPJavaPluginsManager

Check if a method with arguments exists on a plugin

hideAllPorts() - Method in interface com.telelogic.rhapsody.core.IRPGraphNode

method hideAllPorts

highlightByHandle(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication

highlight by handle

highLightElement(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPApplication

highlight element

highLightElement() - Method in interface com.telelogic.rhapsody.core.IRPModelElement

Locates the element in the Rhapsody browser, and highlights the element in the diagram where it appears.

highlightFromCode(String, int) - Method in interface com.telelogic.rhapsody.core.IRPProject

Highlights in the Rhapsody browser the model element associated with the specified line of code.

HTML - Static variable in class com.telelogic.rhapsody.core.IRPMatrixView.ContentFormat

Export in HTML format.

HTML - Static variable in class com.telelogic.rhapsody.core.IRPTableView.ContentFormat

Export in HTML format.

HYPNameType - Class in com.telelogic.rhapsody.core

HYPNameType() - Constructor for class com.telelogic.rhapsody.core.HYPNameType

I

ID - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column.AnnotationAttribute

> Value to be used for Property parameter of addColumn method, when AnnotationAttribute is selected for the Type parameter of addColumn method.

ID - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column.RequirementAttribute

> Value to be used for Property parameter of addColumn method, when RequirementAttribute is selected for the Type parameter of addColumn method.

IGNORE_UNRESOLVED - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.UnresolvedKind

IMAGE_ONLY_SHOW_NAME - Static variable in class com.telelogic.rhapsody.core.IRPGraphElement.ImageLayout

> Set image layout as show image only with name

IMAGE_ONLY_WITHOUT_NAME - Static variable in class com.telelogic.rhapsody.core.IRPGraphElement.ImageLayout

> Set image layout as show image only without name

Implementation - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column.UserDefinedMethod

> Use this value to declare that a dynamic java code was set to be executed for this column.

importClasses() - Method in interface com.telelogic.rhapsody.core.IRPApplication

> import Classes

importPackageFromRose(String, String, String) - Method in interface com.telelogic.rhapsody.core.IRPProject

> Imports the specified Rose package.

importProjectFromRose(String, String) - Method in interface com.telelogic.rhapsody.core.IRPProject

> Imports the specified Rose project.

importSyncSimulinkBlock2(IRPModelElement, String, String, String, String) - Method in interface com.telelogic.rhapsody.core.IRPApplication

> Imports a Simulink model into a Rhapsody model.

importTlb(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication

> import tlb

INCOMING_RELATION - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.References.RelationKind

INITIAL_VALUE - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

initializeApplication(IRPApplication) - Method in interface com.telelogic.rhapsody.core.IRPRhapsodyServer

> initializeApplication

initializeRhapsodyApplicationDllServer(IRPApplication) - Static method in class com.telelogic.rhapsody.core.RhapsodyAppServer

> For internal use only.

I

insertProject(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
Insert existing project into current workspace

insertProjectFromDesignManager(String, String, String, String, String, String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
insertProjectFromDesignManager

invokePluginsMethod(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
invoke Plugins Method

invokeRPE() - Method in interface com.telelogic.rhapsody.core.IRPApplication
Opens the IBM Rational Rhapsody Report Generator wizard.

IRPAcceptEventAction - Interface in com.telelogic.rhapsody.core
The IRPAcceptEventAction interface represents Accept Event Action elements in a statechart or activity diagram.

IRPAcceptTimeEvent - Interface in com.telelogic.rhapsody.core
The IRPAcceptTimeEvent interface represents Accept Time Event elements in activity diagrams and statecharts.

IRPAction - Interface in com.telelogic.rhapsody.core
The IRPAction interface represents the action defined for a transition in a statechart.

IRPActionBlock - Interface in com.telelogic.rhapsody.core
The IRPActionBlock interface represents action blocks in sequence diagrams.

IRPActivityDiagram - Interface in com.telelogic.rhapsody.core
The IRPActivityDiagram interface represents activity diagrams in Rational Rhapsody models.

IRPActor - Interface in com.telelogic.rhapsody.core
The IRPActor interface represents actors in Rhapsody models.

IRPAnnotation - Interface in com.telelogic.rhapsody.core
The IRPAnnotation interface represents the different types of annotations you can add to your model - notes, comments, constraints, and requirements.

IRPApplication - Interface in com.telelogic.rhapsody.core
The IRPApplication interface represents the Rhapsody application, and its methods reflect many of the commands that you can access from the Rhapsody menu bar.

IRPApplication.AddToModel_Mode - Class in com.telelogic.rhapsody.core
This class holds constant values to be used with addToModelEx method.

IRPApplication.AddToModel_Mode() - Constructor for class com.telelogic.rhapsody.core.IRPApplication.AddToModel_Mode

IRPArgument - Interface in com.telelogic.rhapsody.core
The IRPArgument interface represents an argument of an operation or an event.

IRPASCIIFile - Interface in com.telelogic.rhapsody.core

IRPAssociationClass - Interface in com.telelogic.rhapsody.core
The IRPAssociationClass interface represents association classes in Rational Rhapsody models.

IRPAssociationRole - Interface in com.telelogic.rhapsody.core
The IRPAssociationRole interface represents the association roles that link objects in communication diagrams.

IRPAttribute - Interface in com.telelogic.rhapsody.core
The IRPAttribute interface represents attributes of a class, and global variables.

IRPAXViewCtrl - Interface in com.telelogic.rhapsody.core

IRPBaseExternalCodeGeneratorTool - Interface in com.telelogic.rhapsody.core

IRPCallOperation - Interface in com.telelogic.rhapsody.core
The IRPCallOperation interface represents call operation elements in activity diagrams.

IRPClass - Interface in [com.telelogic.rhapsody.core](#)
    The IRPClass interface represents classes in Rational Rhapsody models.
IRPClassifier - Interface in [com.telelogic.rhapsody.core](#)
    Represents the features shared by elements such as classes, actors, use cases, and types.
IRPClassifierRole - Interface in [com.telelogic.rhapsody.core](#)

IRPCodeGenerator - Interface in [com.telelogic.rhapsody.core](#)

IRPCodeGenSimplifiersRegistry - Interface in [com.telelogic.rhapsody.core](#)

IRPCollaboration - Interface in [com.telelogic.rhapsody.core](#)
    The IRPCollaboration interface represents the capabilities included in sequence diagrams and
    communications diagrams.
IRPCollaborationDiagram - Interface in [com.telelogic.rhapsody.core](#)
    The IRPCollaborationDiagram interface represents collaboration diagrams in a Rational Rhapsody
    model.
IRPCollection - Interface in [com.telelogic.rhapsody.core](#)
    The IRPCollection interface contains methods used to store and manipulate collections of various
    types of elements that you may have in your Rational Rhapsody model.
IRPComment - Interface in [com.telelogic.rhapsody.core](#)
    The IRPComment interface represents comments in a Rational Rhapsody model.
IRPComponent - Interface in [com.telelogic.rhapsody.core](#)
    The IRPComponent interface represents a code generation component defined in a Rhapsody model.
IRPComponentDiagram - Interface in [com.telelogic.rhapsody.core](#)
    The IRPComponentDiagram interface represents component diagrams in Rational Rhapsody models.
IRPComponentInstance - Interface in [com.telelogic.rhapsody.core](#)

IRPConditionMark - Interface in [com.telelogic.rhapsody.core](#)
    The IRPConditionMark interface represents condition marks in sequence diagrams.
IRPConfiguration - Interface in [com.telelogic.rhapsody.core](#)
    The IRPConfiguration interface represents a code generation configuration within a specific
    component.
IRPConnector - Interface in [com.telelogic.rhapsody.core](#)
    The IRPConnector interface represents the characteristics shared by the various types of "connector"
    elements that can be included in a statechart, such as condition connectors, history connectors, join
    sync bar connectors, and fork sync bar connectors.
IRPConstraint - Interface in [com.telelogic.rhapsody.core](#)
    The IRPConstraint interface represents constraints in a Rational Rhapsody model.
IRPContextSpecification - Interface in [com.telelogic.rhapsody.core](#)
    The IRPContextSpecification interface represents the exact context of an object in a hierarchy.
IRPControlledFile - Interface in [com.telelogic.rhapsody.core](#)
    The IRPControlledFile interface represents a controlled file in a Rhapsody model.
IRPDependency - Interface in [com.telelogic.rhapsody.core](#)
    The IRPDependency interface represents dependencies in a Rational Rhapsody model.
IRPDeploymentDiagram - Interface in [com.telelogic.rhapsody.core](#)
    The IRPDeploymentDiagram interface represents deployment diagrams in Rational Rhapsody
    models.
IRPDestructionEvent - Interface in [com.telelogic.rhapsody.core](#)
    The IRPDestructionEvent interface represents destruction events in sequence diagrams.
IRPDiagram - Interface in [com.telelogic.rhapsody.core](#)

The IRPDiagram interface contains the methods shared by all the interfaces that represent specific types of diagrams.

IRPDiagSynthAPI - Interface in com.telelogic.rhapsody.core

IRPEnumerationLiteral - Interface in com.telelogic.rhapsody.core

IRPEvent - Interface in com.telelogic.rhapsody.core
The IRPEvent interface represents events in Rational Rhapsody models.

IRPEventReception - Interface in com.telelogic.rhapsody.core

IRPExecutionOccurrence - Interface in com.telelogic.rhapsody.core

IRPExternalCheckRegistry - Interface in com.telelogic.rhapsody.core

IRPExternalCodeGeneratorInvoker - Interface in com.telelogic.rhapsody.core

IRPExternalIDERegistry - Interface in com.telelogic.rhapsody.core

IRPExternalRoundtripInvoker - Interface in com.telelogic.rhapsody.core

IRPFile - Interface in com.telelogic.rhapsody.core
The IRPFile interface represents a file or folder to be generated during code generation.

IRPFileFragment - Interface in com.telelogic.rhapsody.core

IRPFlow - Interface in com.telelogic.rhapsody.core

IRPFlowchart - Interface in com.telelogic.rhapsody.core
The IRPFlowchart interface represents activities in Rational Rhapsody models.

IRPFlowItem - Interface in com.telelogic.rhapsody.core
The IRPFlowItem interface represents item flows in Rational Rhapsody models.

IRPGeneralization - Interface in com.telelogic.rhapsody.core

IRPGraphEdge - Interface in com.telelogic.rhapsody.core

IRPGraphElement - Interface in com.telelogic.rhapsody.core

IRPGraphElement.ImageLayout - Class in com.telelogic.rhapsody.core
This class contains constant values for use with the method setImageLayout

IRPGraphElement.ImageLayout() - Constructor for class com.telelogic.rhapsody.core.IRPGraphElement.ImageLayout

IRPGraphicalProperty - Interface in com.telelogic.rhapsody.core

IRPGraphNode - Interface in com.telelogic.rhapsody.core

IRPGuard - Interface in com.telelogic.rhapsody.core

IRPHyperLink - Interface in com.telelogic.rhapsody.core
The IRPHyperLink interface represents hyperlinks in Rational Rhapsody models.

IRPImageMap - Interface in com.telelogic.rhapsody.core

IRPInstance - Interface in com.telelogic.rhapsody.core

IRPInstanceSlot - Interface in com.telelogic.rhapsody.core

IRPInstanceSpecification - Interface in com.telelogic.rhapsody.core

IRPInstanceValue - Interface in com.telelogic.rhapsody.core
> The IRPInstanceValue interface is used in contexts where a single model element must be stored.

IRPIntegrator - Interface in com.telelogic.rhapsody.core

IRPInteractionOccurrence - Interface in com.telelogic.rhapsody.core

IRPInteractionOperand - Interface in com.telelogic.rhapsody.core
> The IRPInteractionOperand interface represents interaction operands in Rational Rhapsody models.

IRPInteractionOperator - Interface in com.telelogic.rhapsody.core

IRPInterfaceItem - Interface in com.telelogic.rhapsody.core
> The IRPInterfaceItem interface represents the features shared by operations, events, and event receptions in Rational Rhapsody models.

IRPInternalOEMPlugin - Interface in com.telelogic.rhapsody.core

IRPJavaPlugins - Interface in com.telelogic.rhapsody.core

IRPLink - Interface in com.telelogic.rhapsody.core
> The IRPLink interface represents links in Rational Rhapsody models.

IRPLiteralSpecification - Interface in com.telelogic.rhapsody.core
> The IRPLiteralSpecification interface is used in contexts where a single value must be stored.

IRPMatrixLayout - Interface in com.telelogic.rhapsody.core

IRPMatrixLayout.QueryOrElementsList - Class in com.telelogic.rhapsody.core
> This class contains constant values for use with the methods setFromElementTypesUseQueryOrElementsList and setToElementTypesUseQueryOrElementsList.

IRPMatrixLayout.QueryOrElementsList() - Constructor for class com.telelogic.rhapsody.core.IRPMatrixLayout.QueryOrElementsList

IRPMatrixView - Interface in com.telelogic.rhapsody.core
> The IRPMatrixView interface represents Matrix View elements in Rhapsody models.

IRPMatrixView.ContentFormat - Class in com.telelogic.rhapsody.core
> This class contains values that specify export format

IRPMatrixView.ContentFormat() - Constructor for class com.telelogic.rhapsody.core.IRPMatrixView.ContentFormat

IRPMessage - Interface in com.telelogic.rhapsody.core

IRPMessagePoint - Interface in com.telelogic.rhapsody.core

IRPModelElement - Interface in com.telelogic.rhapsody.core
> The IRPModelElement interface represents an element in a Rational Rhapsody model, and its methods reflect the behavior shared by the various types of model elements.

IRPModelElement.OSLCLink - Class in com.telelogic.rhapsody.core
> Constant values used with elements of this type

I

IRPStatechartDiagram - Interface in com.telelogic.rhapsody.core
>The IRPStatechartDiagram interface represents statecharts in a Rational Rhapsody model.

IRPStateVertex - Interface in com.telelogic.rhapsody.core
>The IRPStateVertex interface represents the characteristics that are shared by various statechart elements such as states, join/fork connectors, and condition connectors.

IRPStereotype - Interface in com.telelogic.rhapsody.core
>The IRPStereotype interface represents sterotypes in Rational Rhapsody models.

IRPStructureDiagram - Interface in com.telelogic.rhapsody.core
>The IRPStructureDiagram interface represents structure diagrams in a Rational Rhapsody model.

IRPSwimlane - Interface in com.telelogic.rhapsody.core
>The IRPSwimlane interface represents swimlanes in an activity diagram.

IRPSysMLPort - Interface in com.telelogic.rhapsody.core
>The IRPSysMLPort interface represents flowport elements in Rhapsody models.

IRPTableLayout - Interface in com.telelogic.rhapsody.core

IRPTableLayout.Column - Class in com.telelogic.rhapsody.core
>This class holds constant values to be used with addColumn method.

IRPTableLayout.Column() - Constructor for class com.telelogic.rhapsody.core.IRPTableLayout.Column

IRPTableLayout.Column.AnnotationAttribute - Class in com.telelogic.rhapsody.core
>Contains values to be used for Property parameter of addColumn method, when AnnotationAttribute is selected for the Type parameter of addColumn method.

IRPTableLayout.Column.AnnotationAttribute() - Constructor for class
com.telelogic.rhapsody.core.IRPTableLayout.Column.AnnotationAttribute

IRPTableLayout.Column.DependsOn - Class in com.telelogic.rhapsody.core
>Contains the pre-defined values to be used for Property parameter of addColumn method, when DependsOn is selected for the Type parameter of addColumn method.

IRPTableLayout.Column.DependsOn() - Constructor for class
com.telelogic.rhapsody.core.IRPTableLayout.Column.DependsOn

IRPTableLayout.Column.FlowAttribute - Class in com.telelogic.rhapsody.core
>Contains values to be used for Property parameter of addColumn method, when FlowAttribute is selected for the Type parameter of addColumn method.

IRPTableLayout.Column.FlowAttribute() - Constructor for class
com.telelogic.rhapsody.core.IRPTableLayout.Column.FlowAttribute

IRPTableLayout.Column.GeneralAttribute - Class in com.telelogic.rhapsody.core
>Contains values to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.

IRPTableLayout.Column.GeneralAttribute() - Constructor for class
com.telelogic.rhapsody.core.IRPTableLayout.Column.GeneralAttribute

IRPTableLayout.Column.ImplementationCellType - Class in com.telelogic.rhapsody.core

IRPTableLayout.Column.ImplementationCellType() - Constructor for class
com.telelogic.rhapsody.core.IRPTableLayout.Column.ImplementationCellType

IRPTableLayout.Column.RelationAttributeFrom - Class in com.telelogic.rhapsody.core
>Contains values to be used for Property parameter of addColumn method, when RelationAttributeFrom is selected for the Type parameter of addColumn method.

IRPUnit - Interface in com.telelogic.rhapsody.core

> The IRPUnit interface represents model elements that can be saved as separate files.

IRPUseCase - Interface in com.telelogic.rhapsody.core

IRPUseCaseDiagram - Interface in com.telelogic.rhapsody.core

> The IRPUseCaseDiagram interface represents use case diagrams in a Rational Rhapsody model.

IRPValueSpecification - Interface in com.telelogic.rhapsody.core

> The interface IRPValueSpecification represents the UML concept of "value specification" and serves as the base interface for IRPContextSpecification, IRPInstanceValue, and IRPLiteralSpecification.

IRPVariable - Interface in com.telelogic.rhapsody.core

> The IRPVariable interface represents the characteristics shared by model elements such as attributes, variables, and arguments.

isActivelyManaged() - Method in interface com.telelogic.rhapsody.core.IRPProject

> Checks whether the project is an actively-managed Design Manager project.

isAnd() - Method in interface com.telelogic.rhapsody.core.IRPState

> Checks whether the state contains one or more And Lines.

isArray() - Method in interface com.telelogic.rhapsody.core.IRPType

> method isArray

isATemplate() - Method in interface com.telelogic.rhapsody.core.IRPModelElement

> Checks whether the model element is a template.

isCompound() - Method in interface com.telelogic.rhapsody.core.IRPState

> Checks whether the state is a compound state, meaning a state that contains one or more substates.

isConditionConnector() - Method in interface com.telelogic.rhapsody.core.IRPConnector

> Checks whether the connector is a condition connector.

isDefaultTransition() - Method in interface com.telelogic.rhapsody.core.IRPTransition

> method isDefaultTransition

isDescriptionRTF() - Method in interface com.telelogic.rhapsody.core.IRPModelElement

> Checks whether the description for the element is in RTF format.

isDiagramConnector() - Method in interface com.telelogic.rhapsody.core.IRPConnector

> Checks whether the connector is a diagram connector.

isDirectoryPerModelComponent(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPComponent

> method isDirectoryPerModelComponent

isDisplayNameRTF() - Method in interface com.telelogic.rhapsody.core.IRPModelElement

> Checks whether the label of the element is in RTF format.

isEmpty() - Method in interface com.telelogic.rhapsody.core.IRPFile

> method isEmpty

isEnum() - Method in interface com.telelogic.rhapsody.core.IRPType

> For types whose "kind" was set to Language, parses the declaration to see if the type is actually an enum.

isEqualTo() - Method in interface com.telelogic.rhapsody.core.IRPType

> method isEqualTo

isForkConnector() - Method in interface com.telelogic.rhapsody.core.IRPConnector

> Checks whether the connector is a fork sync bar connector.

isHistoryConnector() - Method in interface com.telelogic.rhapsody.core.IRPConnector

> Checks whether the connector is a history connector.

isImplicit() - Method in interface com.telelogic.rhapsody.core.IRPType

> method isImplicit

isJoinConnector() - Method in interface com.telelogic.rhapsody.core.IRPConnector

> Checks whether the connector is a join sync bar connector.

isJunctionConnector() - Method in interface com.telelogic.rhapsody.core.IRPConnector

I

isSpecificationRTF() - Method in interface com.telelogic.rhapsody.core.IRPAnnotation
> Checks whether the specification is in RTF format

isStaticReaction() - Method in interface com.telelogic.rhapsody.core.IRPTransition
> method isStaticReaction

isStruct() - Method in interface com.telelogic.rhapsody.core.IRPType
> For types whose "kind" was set to Language, parses the declaration to see if the type is actually a struct.

isStubConnector() - Method in interface com.telelogic.rhapsody.core.IRPConnector
> Checks whether the connector is an EnterExit point.

isTemplate() - Method in interface com.telelogic.rhapsody.core.IRPType
> method isTemplate

isTerminationConnector() - Method in interface com.telelogic.rhapsody.core.IRPConnector
> Checks whether the connector is a termination connector.

isTimeout() - Method in interface com.telelogic.rhapsody.core.IRPTrigger
> method isTimeout

isTypelessObject() - Method in interface com.telelogic.rhapsody.core.IRPRelation
> method isTypelessObject

isUnion() - Method in interface com.telelogic.rhapsody.core.IRPType
> For types whose "kind" was set to Language, parses the declaration to see if the type is actually a union.

ITEM_FLOWS - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.FlowAttribute
> Value to be used for Property parameter of addColumn method, when FlowAttribute is selected for the Type parameter of addColumn method.

itsCompoundSource() - Method in interface com.telelogic.rhapsody.core.IRPTransition
> method itsCompoundSource

L

LABEL - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

LABEL - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column.GeneralAttribute
> Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.

LESS_THAN - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.References.QuantityOperator

LINK_FROM - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column.RequirementAttribute

LINK_FROM_FULLNAME - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column.RequirementAttribute

LINK_SUSPECT - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column.RequirementAttribute

LINK_TYPE - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column.RequirementAttribute

LIST_OF_MODEL_ELEMENTS - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column.ImplementationCellType
> Value to be used for cellType parameter of SetColumnImplementationCellType method.

load(int) - Method in interface com.telelogic.rhapsody.core.IRPUnit
> Loads the unit.

loadFromQuery(IRPTableLayout) - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
> Loads the settings from the specified query into the search query object.

loadPlugin(String, IRPCollection, IRPCollection) - Method in class com.telelogic.rhapsody.core.RPJavaPluginsManager
> Loads the Java plugin main class

LOCALLY_OVERRIDDEN_PROPERTY - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

locateInBrowser() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Locates the model element in the Rhapsody browser.

locateInIDE(IRPConfiguration, String, int) - Method in interface com.telelogic.rhapsody.core.IRPProject
> For projects that use Rhapsody's integration with Eclipse or Visual Studio, you can use the locateInIDE method to have the IDE highlight a specific line in a specific source file.

lockOnDesignManager() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Locks the model element on the Design Manager server.

loginToDesignManagerWithAlias(String, String) - Method in interface
com.telelogic.rhapsody.core.IRPApplication
      loginToDesignManagerWithAlias
loginToDesignManagerWithCertificate(String, String, String) - Method in interface
com.telelogic.rhapsody.core.IRPApplication
      loginToDesignManagerWithCertificate
loginToDesignManagerWithUsername(String, String, String) - Method in interface
com.telelogic.rhapsody.core.IRPApplication
      loginToDesignManagerWithUsername

M

m_javaPluginsManager - Static variable in class com.telelogic.rhapsody.core.RhapsodyAppServer
    For internal use only.
make() - Method in interface com.telelogic.rhapsody.core.IRPApplication
    make
makeUnidirect() - Method in interface com.telelogic.rhapsody.core.IRPRelation
    method makeUnidirect
matchOnSignature(IRPInterfaceItem) - Method in interface com.telelogic.rhapsody.core.IRPInterfaceItem
    Compares the signature of the operation with the signature of the operation that was provided as an
    argument.
mergeElements(IRPModelElement, IRPModelElement) - Method in interface
com.telelogic.rhapsody.core.IRPApplication
    mergeElements
MODEL_ELEMENT - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.ImplementationCellType
    Value to be used for cellType parameter of SetColumnImplementationCellType method.
MORE_THAN - Static variable in class
com.telelogic.rhapsody.core.IRPSearchQuery.References.QuantityOperator

moveFragmentInOwner(int) - Method in interface com.telelogic.rhapsody.core.IRPFileFragment
    method moveFragmentInOwner
moveToAnotherProjectLeaveAReference(IRPModelElement) - Method in interface
com.telelogic.rhapsody.core.IRPUnit
    Moves the unit to a different project, and adds a reference to it in the original project.
moveToDesignManager(String, String, String, String, String) - Method in interface
com.telelogic.rhapsody.core.IRPProject
    Moves the model to a Rhapsody Design Manager server.
moveToDesignManagerAfterLogin(String, String, String) - Method in interface
com.telelogic.rhapsody.core.IRPProject
    Moves the model to a Rhapsody Design Manager server.
MULTIPLICITY - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

N

NAME - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

NAME - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column.GeneralAttribute
> Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for the Type parameter of addColumn method.

needsCodeGeneration() - Method in interface com.telelogic.rhapsody.core.IRPConfiguration
> method needsCodeGeneration checks is code generation is needed

newProjectOnDesignManager(String, String, String, String, String, String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> Creates a new project on the Design Manager server.

NONE - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.ViewsToSearch

NOTES_AND_TEXT - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

notifyFileChanged(String, int) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> File change notification

notifyGenerationDone() - Method in interface com.telelogic.rhapsody.core.IRPExternalCodeGeneratorInvoker
> method notifyGenerationDone

notifySimplificationDone() - Method in interface com.telelogic.rhapsody.core.IRPCodeGenSimplifiersRegistry
> method notifySimplificationDone

O

okToAddAggregate(IRPModelElement, IRPModelElement) - Method in class
com.telelogic.rhapsody.core.RPExternalRoundtrip
       method OkToAddAggregate
okToMakeAction(IRPModelElement, String) - Method in class
com.telelogic.rhapsody.core.RPExternalRoundtrip
       method OkToMakeAction
onCodeGenerationCompleted() - Method in class com.telelogic.rhapsody.core.RPCodeGeneratorListener
       Called after code generation is completed
onDiagramOpen(IRPDiagram) - Method in class com.telelogic.rhapsody.core.RPApplicationListener
       Called when diagram is opened
onDoubleClick(IRPModelElement) - Method in class com.telelogic.rhapsody.core.RPApplicationListener
       Called on double click
onElementsChanged(String) - Method in class com.telelogic.rhapsody.core.RPApplicationListener

onFeaturesOpen(IRPModelElement) - Method in class com.telelogic.rhapsody.core.RPApplicationListener
       Called when element features dialog is opened
onGetStatus(IRPUnit) - Method in class com.telelogic.rhapsody.core.RPRTCListener
       Called on request to get unit's status
onIDETextMessage(String) - Method in class com.telelogic.rhapsody.core.RPExternalIDEManager
       method OnIDETextMessage
onInvokeSearch(IRPModelElement) - Method in class
com.telelogic.rhapsody.core.RPExternalIDEManager
       method OnSearchRequest
onLocateInPendingChanges(IRPUnit) - Method in class com.telelogic.rhapsody.core.RPRTCListener
       Called on request to locate unit in pending changes view
onLocateInRepositoryFiles(IRPUnit) - Method in class com.telelogic.rhapsody.core.RPRTCListener
       Called on request to locate unit in repository files' view
onLock(IRPUnit) - Method in class com.telelogic.rhapsody.core.RPRTCListener
       Called on request to lock unit
ONLY_UNRESOLVED_OR_UNLOADED - Static variable in class
com.telelogic.rhapsody.core.IRPSearchQuery.UnresolvedKind

onMenuItemSelect(String) - Method in interface com.telelogic.rhapsody.core.IRPInternalOEMPlugin
       Selects a given menu item
OnMenuItemSelect(String) - Method in class com.telelogic.rhapsody.core.RPUserPlugin

onMenuItemSelectWithParameters(String, String) - Method in interface
com.telelogic.rhapsody.core.IRPInternalOEMPlugin
       For internal use only.
onNewSearchResult(IRPSearchResult) - Method in class com.telelogic.rhapsody.core.RPSearchListener
       Called during search

onNotifyMessage(String, IRPCollection, IRPCollection) - Method in class com.telelogic.rhapsody.core.RPExternalIDEManager
> method OnNotifyMessage

onPerspectiveChange(String, String) - Method in class com.telelogic.rhapsody.core.RPApplicationListener

onRefreshStatus() - Method in class com.telelogic.rhapsody.core.RPRTCListener
> Called to refresh status cache in RTC

onSelectionChanged() - Method in class com.telelogic.rhapsody.core.RPApplicationListener

onShowHistory(IRPUnit) - Method in class com.telelogic.rhapsody.core.RPRTCListener
> Called on request to show CM revisions' history of a unit

onShowInUnitView(IRPModelElement) - Method in class com.telelogic.rhapsody.core.RPExternalIDEManager
> method OnShowInUnitView

OnTrigger(String) - Method in class com.telelogic.rhapsody.core.RPUserPlugin

onUnlock(IRPUnit) - Method in class com.telelogic.rhapsody.core.RPRTCListener
> Called on request to un-lock unit

open(String) - Method in interface com.telelogic.rhapsody.core.IRPASCIIFile
> open file

open() - Method in interface com.telelogic.rhapsody.core.IRPControlledFile
> Opens the controlled file, using the associated program.

open() - Method in interface com.telelogic.rhapsody.core.IRPMatrixView
> method open

OPEN - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.ViewsToSearch

open() - Method in interface com.telelogic.rhapsody.core.IRPTableView
> method open

openActiveXView(String, IRPCollection, IRPCollection) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> Open ActivexView

openAdvancedSearchAndReplaceDialog() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> Open Advanced Search and Replace dialog

openCSVFile(String, int) - Method in interface com.telelogic.rhapsody.core.IRPProject
> Displays the content of the specified csv file in a new tab in the Output window.

openDiagram() - Method in interface com.telelogic.rhapsody.core.IRPDiagram
> Opens the diagram.

openDiagram(IRPDiagram) - Method in class com.telelogic.rhapsody.core.RPExternalIDEManager
> method OpenDiagram

openDiagramView(IRPDiagram) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> method OpenDiagramView

openDiagramView() - Method in interface com.telelogic.rhapsody.core.IRPDiagram
> Used internally by Rational Rhapsody to display diagrams within Eclipse (when using the Rhapsody-Eclipse platform integration).

openDiagramView() - Method in interface com.telelogic.rhapsody.core.IRPStatechart
> Used internally by Rational Rhapsody to display diagrams within Eclipse (when using the Rhapsody-Eclipse platform integration).

openFeaturesDialog(int) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Displays the information for the element in the Features window.

openFile(String) - Method in class com.telelogic.rhapsody.core.RPExternalIDEManager

Opens

openFileAndSelectLine(String, int) - Method in class com.telelogic.rhapsody.core.RPExternalIDEManager
>   Opens file and selects line

openFileList(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
>   method openFileList

openHotFeatures() - Method in class com.telelogic.rhapsody.core.RPExternalIDEManager
>   method OpenHotFeatures

openNewFeatures(IRPModelElement) - Method in class
com.telelogic.rhapsody.core.RPExternalIDEManager
>   method OpenNewFeatures

openProject(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
>   Opens an existing Rhapsody project

openProjectFromDesignManager(String, String, String, String, String, String) - Method in interface
com.telelogic.rhapsody.core.IRPApplication
>   openProjectFromDesignManager

openProjectFromDesignManagerAfterLogin(String, String, String, String) - Method in interface
com.telelogic.rhapsody.core.IRPApplication
>   openProjectFromDesignManagerAfterLogin

openProjectFromURL(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
>   open Project From URL

openProjectWithLastSession(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
>   open project with last session

openProjectWithoutSubUnits(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
>   open project without subunits

openYesNoCancelQuestion(String, String, String, int) - Method in class
com.telelogic.rhapsody.core.RPExternalIDEManager
>   Display YES OCancel message box with check-box to remmember the chosen reply

OPERATION_BODIES - Static variable in class
com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

OR - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.SubQueriesOperator

OUTGOING_RELATION - Static variable in class
com.telelogic.rhapsody.core.IRPSearchQuery.References.RelationKind

overrideInheritance() - Method in interface com.telelogic.rhapsody.core.IRPState
>   Breaks the inheritance relationship between this state and the corresponding state from the statechart
>   of the class that this class is derived from.

overrideInheritance() - Method in interface com.telelogic.rhapsody.core.IRPStatechart
>   Breaks the inheritance relationship between this statechart and the statechart of the base class.

overrideInheritance() - Method in interface com.telelogic.rhapsody.core.IRPTransition
>   method overrideInheritance

OWNER - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column.GeneralAttribute
>   Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for
>   the Type parameter of addColumn method.

P

pasteSelected() - Method in interface com.telelogic.rhapsody.core.IRPSelection

      Perform PASTE operation under current selection

populateDiagram(IRPCollection, IRPCollection, String) - Method in interface
com.telelogic.rhapsody.core.IRPDiagram

      Populates the diagram with the elements and types of relations specified.

populateDiagram(IRPCollection, IRPCollection, String) - Method in interface
com.telelogic.rhapsody.core.IRPStatechart

      Populates the statechart with the elements and types of relations specified.

populateRemoteRequirements() - Method in interface com.telelogic.rhapsody.core.IRPPackage

      For Design Manager projects, populates the package with the remote requirements that model
      elements do not yet have dependencies upon.

populateSlots() - Method in interface com.telelogic.rhapsody.core.IRPInstanceSpecification

      method populateSlots

PORT_PROVIDED_INERFACE - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.RelationAttributeFrom

      Value to be used for Property parameter of addColumn method, when RelationAttributeFrom is
      selected for the Type parameter of addColumn method.

PORT_PROVIDED_INERFACE - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.RelationAttributeTo

      Value to be used for Property parameter of addColumn method, when RelationAttributeTo is selected
      for the Type parameter of addColumn method.

PORT_REQUIRED_INERFACE - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.RelationAttributeFrom

      Value to be used for Property parameter of addColumn method, when RelationAttributeFrom is
      selected for the Type parameter of addColumn method.

PORT_REQUIRED_INERFACE - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.RelationAttributeTo

      Value to be used for Property parameter of addColumn method, when RelationAttributeTo is selected
      for the Type parameter of addColumn method.

postSimplify(IRPModelElement, IRPModelElement, String) - Method in class
com.telelogic.rhapsody.core.RPCodeGenSimplifier

      post element simplification

progressTaskAsynchCallback(int, int) - Method in interface
com.telelogic.rhapsody.core.IRPExternalIDERegistry

      Initiate Progress Task execution

progressTaskAsynchEliminate(int, int) - Method in interface
com.telelogic.rhapsody.core.IRPExternalIDERegistry

      Initiate Progress Task execution

progressTaskStep(int, int, int) - Method in class com.telelogic.rhapsody.core.RPExternalIDEManager

      Indicate a Progress Task step performed

PROVIDED_INERFACE_OPERATIONS - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.RelationAttributeFrom
> Value to be used for Property parameter of addColumn method, when RelationAttributeFrom is
> selected for the Type parameter of addColumn method.

PROVIDED_INERFACE_OPERATIONS - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.RelationAttributeTo
> Value to be used for Property parameter of addColumn method, when RelationAttributeTo is selected
> for the Type parameter of addColumn method.

Q

QUERY - Static variable in class com.telelogic.rhapsody.core.IRPMatrixLayout.QueryOrElementsList
> When QUERY is used as the parameter for the methods setFromElementTypesUseQueryOrElementsList and setToElementTypesUseQueryOrElementsList, it indicates that a query will be used to determine the "from" element types or "to" element types for the matrix.

QUERY - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.QueryOrElementsList
> When QUERY is used as the parameter for the methods setFromElementTypesUseQueryOrElementsList and setToElementTypesUseQueryOrElementsList, it indicates that a query is going to be used to determine the "from" element types or "to" element types for the relation table.

quit() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> quit application

R

rebuild() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> rebuild

rebuildEntireProject() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> rebuildEntireProject

rebuildWithDependencies() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> rebuildWithDependencies

reCalculateEventsBaseId() - Method in interface com.telelogic.rhapsody.core.IRPPackage
> If you are using Rational Rhapsody's default numbering scheme for event IDs, then a certain amount of IDs are reserved for each package.

recalculateEventsBaseIds() - Method in interface com.telelogic.rhapsody.core.IRPProject
> If you are using Rational Rhapsody's default numbering scheme for event IDs, then a certain amount of IDs are reserved for each package.

receiveMessage(long, long) - Method in interface com.telelogic.rhapsody.core.IRPDiagSynthAPI
> DiagSynthAPI : recieve sequence diagram message

redo() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> Perform Redo

refactorSelectedOperation(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> Changes the name of the currently-selected operation and updates any references to the operation accordingly.

REFERENCE - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.References.RelationKind

referenceToAnotherProject(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPUnit
> Creates a reference to the unit in a different project.

REFINE - Static variable in class com.telelogic.rhapsody.core.IRPModelElement.OSLCLink.Types
> OSLC link type: Refine

refreshAllViews() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> refresh all views

refreshRequest() - Method in class com.telelogic.rhapsody.core.RPExternalIDEManager
> Refresh Rhapsody project/workspace contents

regenerate() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> regenerate

regenerateElements(IRPCollection) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> method regenerateElements

regenerateEntireProject() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> regenerateEntireProject

regenerateWithDependencies() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> regenerateWithDependencies

registerAsActiveObject(IRPApplication) - Static method in class com.telelogic.rhapsody.core.RhapsodyAppServer

For internal use only.

registerCOMClient(int, String, int) - Method in interface com.telelogic.rhapsody.core.IRPApplication
register COM client

RELAION_ATTRIBUTE_FROM - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column
Deprecated.

RELAION_ATTRIBUTE_TO - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column
Deprecated.

RELATION_ATTRIBUTE_FROM - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column
Value used for Type parameter of addColumn method

RELATION_ATTRIBUTE_TO - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column
Value used for Type parameter of addColumn method

reloadCSVFile(String) - Method in interface com.telelogic.rhapsody.core.IRPProject
Reloads the content of the specified csv file in a tab in the Output window.

remove(int) - Method in interface com.telelogic.rhapsody.core.IRPCollection
Removes an element from a collection.

remove() - Method in interface com.telelogic.rhapsody.core.IRPProject
Removes the project from the Rhapsody workspace.

removeAnchor(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPAnnotation
Removes the anchor to the specified model element.

removeColumn(int) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout
Removes the specified column from the table layout.

removeConveyed(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPFlow
method removeConveyed

removeCustomViewOnBrowser(IRPPackage) - Method in interface com.telelogic.rhapsody.core.IRPProject
Removes the specified custom view from the model browser.

removeCustomViewOnDiagram(IRPDiagram, IRPPackage) - Method in interface com.telelogic.rhapsody.core.IRPProject
Removes the specified custom view from the specified diagram.

removeFilterElementTypes() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
Removes any element type filters that you defined to limit the search to certain element types.

removeFilterReferences() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
Removes reference search criterion that was defined for the search query.

removeFilterSearchInFields() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
Removes any element field filters that you defined to limit the search to certain element fields, for example, model element descriptions.

removeFilterStereotypes() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
Removes any stereotype filter that was defined to limit the search to model elements that have certain stereotypes applied to them.

removeFilterSubQueries() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
Removes the subquery criteria that were specified for the search.

removeFilterSubQuery(IRPTableLayout) - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
Removes the specified subquery from the search.

removeFilterTag() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
Removes the tag name and tag value criteria that were defined for the search query.

R

removeFromInstrumentationScope(IRPClassifier) - Method in interface
com.telelogic.rhapsody.core.IRPConfiguration
> method removeFromInstrumentationScope

removeGraphElements(IRPCollection) - Method in interface com.telelogic.rhapsody.core.IRPDiagram
> Removes the specified graphic elements from the diagram.

removeInState(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPObjectNode
> Removes the specified state from the list of "In State" states for the object node.

removeMetaClass(String) - Method in interface com.telelogic.rhapsody.core.IRPStereotype
> Removes a metaclass from the list of metaclasses that the stereotype can be applied to.

removePackageFromInstrumentationScope(IRPPackage) - Method in interface
com.telelogic.rhapsody.core.IRPConfiguration
> method removePackageFromInstrumentationScope

removeProperty(String) - Method in interface com.telelogic.rhapsody.core.IRPGraphElement
> method removeProperty

removeProperty(String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Removes the value that was set for the specified property.

removeProvidedInterface(IRPClass) - Method in interface com.telelogic.rhapsody.core.IRPPort
> method removeProvidedInterface

removeQualifier(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPRelation
> method removeQualifier

removeRedefines(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> method removeRedefines

removeRepresented(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPFlowItem
> Removes the specified element from the collection of information elements that are represented by the
> item flow.

removeRequiredInterface(IRPClass) - Method in interface com.telelogic.rhapsody.core.IRPPort
> method removeRequiredInterface

removeScopeElement(IRPModelElement) - Method in interface
com.telelogic.rhapsody.core.IRPComponent
> Removes the specified model element from the scope of the component.

removeSearchScopeElement(IRPModelElement) - Method in interface
com.telelogic.rhapsody.core.IRPSearchQuery
> Removes the specified model element from the scope for the search.

removeStereotype(IRPStereotype) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
> Removes the specified stereotype from the element.

removeSynthSDToModel2(IRPSequenceDiagram) - Method in interface
com.telelogic.rhapsody.core.IRPDiagSynthAPI
> DiagSynthAPI : remove synth sequence diagarm to model

removeView(int) - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
> Removes the specified view from the list of views to be searched for the search text.

report(String, String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
> report

REQUIRED_INERFACE_OPERATIONS - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.RelationAttributeFrom
> Value to be used for Property parameter of addColumn method, when RelationAttributeFrom is
> selected for the Type parameter of addColumn method.

REQUIRED_INERFACE_OPERATIONS - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.RelationAttributeTo
> Value to be used for Property parameter of addColumn method, when RelationAttributeTo is selected
> for the Type parameter of addColumn method.

REQUIREMENT_ATTRIBUTE - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column
       Value used for Type parameter of addColumn method
REQUIREMENT_ID - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

REQUIREMENT_SPECIFICATION - Static variable in class
com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

reroute() - Method in interface com.telelogic.rhapsody.core.IRPMessage
       method reroute
reset() - Method in interface com.telelogic.rhapsody.core.IRPProgressBar
       method reset
resetCurrentContextClassFactory() - Static method in class
com.telelogic.rhapsody.core.RhapsodyAppServer
       For internal use only.
resetEntryActionInheritance() - Method in interface com.telelogic.rhapsody.core.IRPState
       Restores the inheritance relationship between this state and the corresponding state from the statechart
       of the class that this class is derived from, for the entry action.
resetExitActionInheritance() - Method in interface com.telelogic.rhapsody.core.IRPState
       Restores the inheritance relationship between this state and the corresponding state from the statechart
       of the class that this class is derived from, for the exit action.
resetLabelInheritance() - Method in interface com.telelogic.rhapsody.core.IRPTransition
       method resetLabelInheritance
resetSearchScope() - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
       Resets the search scope to include the entire project, or all projects if multiple projects are open.
rhapPluginAnimationStopped() - Method in interface com.telelogic.rhapsody.core.IRPInternalOEMPlugin
       Notify the Plugin upon RhapPluginAnimationStopped
RhapsodyAppServer - Class in com.telelogic.rhapsody.core
       The RhapsodyAppServer class contains methods relating to accessing an instance of Rhapsody.
RhapsodyAppServer() - Constructor for class com.telelogic.rhapsody.core.RhapsodyAppServer

RhapsodyRuntimeException - Exception in com.telelogic.rhapsody.core

RhapsodyRuntimeException(String) - Constructor for exception
com.telelogic.rhapsody.core.RhapsodyRuntimeException

rhpCheckinLicense(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
       checkin license
rhpCheckoutLicense(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
       checkout license
RhpClassLoader - Class in com.telelogic.rhapsody.core

RhpClassLoader(URL[]) - Constructor for class com.telelogic.rhapsody.core.RhpClassLoader

rhpPluginAnimationStarted() - Method in interface com.telelogic.rhapsody.core.IRPInternalOEMPlugin
       Notify the Plugin upon RhpPluginAnimationStarted
rhpPluginCleanup() - Method in interface com.telelogic.rhapsody.core.IRPInternalOEMPlugin
       Performs cleanup of the Plugin
RhpPluginCleanup() - Method in class com.telelogic.rhapsody.core.RPUserPlugin

rhpPluginDoCommand(String) - Method in interface com.telelogic.rhapsody.core.IRPInternalOEMPlugin

Notify the Plugin to executes a command

rhpPluginFinalCleanup() - Method in interface com.telelogic.rhapsody.core.IRPInternalOEMPlugin
Performs final cleanup of the plugin

RhpPluginFinalCleanup() - Method in class com.telelogic.rhapsody.core.RPUserPlugin

rhpPluginInit() - Method in interface com.telelogic.rhapsody.core.IRPInternalOEMPlugin
Initializes the plugin

RhpPluginInit(IRPApplication) - Method in class com.telelogic.rhapsody.core.RPUserPlugin

rhpPluginInvokeItem() - Method in interface com.telelogic.rhapsody.core.IRPInternalOEMPlugin
Invoke an item of the Plugin

RhpPluginInvokeItem() - Method in class com.telelogic.rhapsody.core.RPUserPlugin

RhpPluginInvokeItem(String) - Method in class com.telelogic.rhapsody.core.RPUserPlugin

rhpPluginOnIDEBuildDone(String) - Method in interface
com.telelogic.rhapsody.core.IRPInternalOEMPlugin
Notify the Plugin upon build done

rhpPluginSetApplication(IRPApplication) - Method in interface
com.telelogic.rhapsody.core.IRPInternalOEMPlugin
Sets the IRPApplication of the plugin

rhpSavingProject() - Method in interface com.telelogic.rhapsody.core.IRPInternalOEMPlugin
Notify the Plugin upon Rhapsody save

RhpUtils - Class in com.telelogic.rhapsody.core

RhpUtils() - Constructor for class com.telelogic.rhapsody.core.RhpUtils

roundtrip() - Method in interface com.telelogic.rhapsody.core.IRPApplication
roundtrip

roundtripElements(IRPCollection) - Method in interface com.telelogic.rhapsody.core.IRPApplication
method roundtripElements

roundtripFile(String, int) - Method in interface com.telelogic.rhapsody.core.IRPRoundTrip
roundtrip file

RP_HYP_FREETEXT - Static variable in class com.telelogic.rhapsody.core.HYPNameType
show user defined name

RP_HYP_LABELTEXT - Static variable in class com.telelogic.rhapsody.core.HYPNameType
show hyperlink target label

RP_HYP_NAMETEXT - Static variable in class com.telelogic.rhapsody.core.HYPNameType
show hyperlink target name

RP_HYP_TAGVALUETEXT - Static variable in class com.telelogic.rhapsody.core.HYPNameType
show hyperlink target tag value

RP_SEARCH_EMPTY_ONLY - Static variable in class com.telelogic.rhapsody.core.SearchFindAsEnum
search for empty string only

RP_SEARCH_EXACT - Static variable in class com.telelogic.rhapsody.core.SearchFindAsEnum
search for exact string

RP_SEARCH_REGEX - Static variable in class com.telelogic.rhapsody.core.SearchFindAsEnum
search as regular expression

RP_SEARCH_WILDCARD - Static variable in class com.telelogic.rhapsody.core.SearchFindAsEnum
search as wildcard

RPApplicationListener - Class in com.telelogic.rhapsody.core

RPApplicationListener() - Constructor for class com.telelogic.rhapsody.core.RPApplicationListener

RPCodeGeneratorListener - Class in com.telelogic.rhapsody.core

RPCodeGeneratorListener() - Constructor for class com.telelogic.rhapsody.core.RPCodeGeneratorListener

RPCodeGenSimplifier - Class in com.telelogic.rhapsody.core

RPCodeGenSimplifier() - Constructor for class com.telelogic.rhapsody.core.RPCodeGenSimplifier

RPExtendedRPClassesFactory - Class in com.telelogic.rhapsody.core

RPExtendedRPClassesFactory() - Constructor for class com.telelogic.rhapsody.core.RPExtendedRPClassesFactory

RPExternalCheck - Class in com.telelogic.rhapsody.core

RPExternalCheck() - Constructor for class com.telelogic.rhapsody.core.RPExternalCheck

RPExternalCodeGenerator - Class in com.telelogic.rhapsody.core

RPExternalCodeGenerator() - Constructor for class com.telelogic.rhapsody.core.RPExternalCodeGenerator

RPExternalIDEManager - Class in com.telelogic.rhapsody.core

RPExternalIDEManager() - Constructor for class com.telelogic.rhapsody.core.RPExternalIDEManager

RPExternalRoundtrip - Class in com.telelogic.rhapsody.core

RPExternalRoundtrip() - Constructor for class com.telelogic.rhapsody.core.RPExternalRoundtrip

RPIntegratorListener - Class in com.telelogic.rhapsody.core

RPIntegratorListener() - Constructor for class com.telelogic.rhapsody.core.RPIntegratorListener

RPJavaPluginsManager - Class in com.telelogic.rhapsody.core

RPJavaPluginsManager() - Constructor for class com.telelogic.rhapsody.core.RPJavaPluginsManager

RPowPaneMgrEvents - Class in com.telelogic.rhapsody.core

RPowPaneMgrEvents() - Constructor for class com.telelogic.rhapsody.core.RPowPaneMgrEvents

RPRoundTripListener - Class in com.telelogic.rhapsody.core

RPRoundTripListener() - Constructor for class com.telelogic.rhapsody.core.RPRoundTripListener

RPRTCListener - Class in com.telelogic.rhapsody.core

RPRTCListener() - Constructor for class com.telelogic.rhapsody.core.RPRTCListener

RPSearchListener - Class in com.telelogic.rhapsody.core

RPSearchListener() - Constructor for class com.telelogic.rhapsody.core.RPSearchListener

RPUserPlugin - Class in com.telelogic.rhapsody.core

RPUserPlugin() - Constructor for class com.telelogic.rhapsody.core.RPUserPlugin

runApplication() - Method in interface com.telelogic.rhapsody.core.IRPApplication
     Runs the application that was built for the project

runHelper(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
     runHelper

runHelperWithParameters(String, String) - Method in interface
com.telelogic.rhapsody.core.IRPApplication
     runHelperWithParameters

S

SATISFY - Static variable in class com.telelogic.rhapsody.core.IRPModelElement.OSLCLink.Types
        OSLC Link Type: Satisfy
save() - Method in interface com.telelogic.rhapsody.core.IRPProject
        Saves the project.
save(int) - Method in interface com.telelogic.rhapsody.core.IRPUnit
        Saves the unit.
saveAll() - Method in interface com.telelogic.rhapsody.core.IRPApplication
        method saveAll
saveAs(String) - Method in interface com.telelogic.rhapsody.core.IRPProject
        Saves the project using the specified path.
saveAsPrevVersion(String, String) - Method in interface com.telelogic.rhapsody.core.IRPProject
        Saves the project, using the format of a previous version of Rhapsody.
saveAsQuery(IRPPackage) - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
        Saves the search query object that you defined as a query in your model.
sDAddConditionMark(long, String, String, String) - Method in interface
com.telelogic.rhapsody.core.IRPDiagSynthAPI
        DiagSynthAPI : send condition mark to instance
search(IRPSearchQuery) - Method in interface com.telelogic.rhapsody.core.IRPSearchManager
        method Search
searchAndShowResults(IRPSearchQuery) - Method in interface
com.telelogic.rhapsody.core.IRPSearchManager
        method SearchAndShowResults
searchAsync(IRPSearchQuery) - Method in interface com.telelogic.rhapsody.core.IRPSearchManager
        method SearchAsync
searchEnded(IRPSearchQuery) - Method in class com.telelogic.rhapsody.core.RPSearchListener
        Called after search ends
SearchFindAsEnum - Class in com.telelogic.rhapsody.core

SearchFindAsEnum() - Constructor for class com.telelogic.rhapsody.core.SearchFindAsEnum

searchStarted(IRPSearchQuery) - Method in class com.telelogic.rhapsody.core.RPSearchListener
        Called before search starts
selectGraphElements(IRPCollection) - Method in interface com.telelogic.rhapsody.core.IRPApplication
        Selects multiple elements in the most recently opened diagram.
selectModelElements(IRPCollection) - Method in interface com.telelogic.rhapsody.core.IRPApplication
        Selects multiple items in the model browser.
sendIDETextMessage(String) - Method in interface com.telelogic.rhapsody.core.IRPExternalIDERegistry
        method SendIDETextMessage
sendMessage(long, String, String, String, String, String) - Method in interface
com.telelogic.rhapsody.core.IRPDiagSynthAPI
        DiagSynthAPI : send sequence diagram message

S

setClassFactory(RPExtendedRPClassesFactory, boolean) - Static method in class
com.telelogic.rhapsody.core.RhapsodyAppServer
>> For internal use only.

setClassifier(IRPClassifier) - Method in interface com.telelogic.rhapsody.core.IRPInstanceSpecification
>> set property classifier

setClassType() - Method in interface com.telelogic.rhapsody.core.IRPTemplateParameter
>> Sets the type of the parameter to "class".

setCMHeader(String) - Method in interface com.telelogic.rhapsody.core.IRPUnit
>> Sets the Configuration Management tool header for the unit.

setCollapseFirstColumn(int) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout
>> Specifies whether or not the fist column should include controls for collapsing and expanding rows that have the same value in the first column.

setCollectionCachingMode(boolean) - Static method in class
com.telelogic.rhapsody.core.RhapsodyAppServer
>> For internal use only.

setCollectionCashingMode(boolean) - Static method in class
com.telelogic.rhapsody.core.RhapsodyAppServer
>> For internal use only.

setColumnContext(int, String) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout
>> If you have defined a context pattern, this method can be used to specify a label from the context pattern, for the specified column.

setColumnDefaultWidth(int, int) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout
>> Sets the default width of the specified column.

setColumnImplementationAllowNew(int, int) - Method in interface
com.telelogic.rhapsody.core.IRPTableLayout
>> For columns that use customized cell behavior, this method can be used to include the New option in the list provided by the picker.

setColumnImplementationAllowSelect(int, int) - Method in interface
com.telelogic.rhapsody.core.IRPTableLayout
>> For columns that use customized cell behavior, this method can be used to include the Select option in the list provided by the picker.

setColumnImplementationCellType(int, String) - Method in interface
com.telelogic.rhapsody.core.IRPTableLayout
>> For columns that use customized cell behavior, this method is used to specify the type of information that will be displayed in the column's cells - string, model element, or list of model elements.

setColumnImplementationDisplayProperty(int, String) - Method in interface
com.telelogic.rhapsody.core.IRPTableLayout
>> For columns that use customized cell behavior, this method is used to specify the type of element information that should be displayed when the cell value type is set to model element or list of model elements, for example, the name or value of the element.

setColumnImplementationGetterCode(int, String) - Method in interface
com.telelogic.rhapsody.core.IRPTableLayout
>> For columns that use customized cell behavior, this method is used to specify the Java code for the getter for the cells in the column.

setColumnImplementationImports(int, String) - Method in interface
com.telelogic.rhapsody.core.IRPTableLayout
>> For columns that use customized cell behavior, this method can be used to specify classes required by your code.

setColumnImplementationPickerCode(int, String) - Method in interface
com.telelogic.rhapsody.core.IRPTableLayout

S

For columns that use customized cell behavior, this method is used to specify the Java code for the picker for the cells in the column.

setColumnImplementationSetterCode(int, String) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout

For columns that use customized cell behavior, this method is used to specify the Java code for the setter for the cells in the column.

setColumnName(int, String) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout

Sets the name of the specified column.

setColumnProperty(int, String) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout

Sets the Property of the specified column.

setColumnType(int, String) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout

Sets the type of the specified table column.

setCompilerSwitches(String) - Method in interface com.telelogic.rhapsody.core.IRPConfiguration

set property compilerSwitches

setComponent(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication

set Component

setComponentType(IRPComponent) - Method in interface com.telelogic.rhapsody.core.IRPComponentInstance

set property componentType

setConfiguration(String) - Method in interface com.telelogic.rhapsody.core.IRPApplication

set Configuration

setContract(IRPClass) - Method in interface com.telelogic.rhapsody.core.IRPPort

set property contract

setCPUtype(String) - Method in interface com.telelogic.rhapsody.core.IRPNode

set property CPUtype

setCurrentContextClassFactory(RPExtendedRPClassesFactory) - Static method in class com.telelogic.rhapsody.core.RhapsodyAppServer

For internal use only.

setDeclaration(String) - Method in interface com.telelogic.rhapsody.core.IRPType

set property declaration

setDeclaration(String) - Method in interface com.telelogic.rhapsody.core.IRPVariable

Specifies an "on-the-fly" declaration for the type of the element instead of using an existing type.

setDecorationStyle(String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement

Used to specify the decoration style that should now be associated with the model element.

setDefaultDirectoryScheme(String) - Method in interface com.telelogic.rhapsody.core.IRPProject

Set's the project's default directory scheme with regard to packages.

setDefaultValue(String) - Method in interface com.telelogic.rhapsody.core.IRPVariable

Sets a new default value for the variable.

setDefultClassFactory(RPExtendedRPClassesFactory) - Static method in class com.telelogic.rhapsody.core.RhapsodyAppServer

For internal use only.

setDelayedReleaseInterfacesMode(boolean) - Static method in class com.telelogic.rhapsody.core.RhapsodyAppServer

For internal use only.

setDependent(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPDependency

Sets the source element in the dependency relation, meaning the element that depends on the other element.

setDependsOn(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPDependency

Sets the target element in the dependency relation, meaning the element on which the first element depends

setDerivedClass(IRPClassifier) - Method in interface com.telelogic.rhapsody.core.IRPGeneralization

set method derivedClass

setDescription(String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement

Sets the specified string as the description of the element.

setDescriptionAndHyperlinks(String, IRPCollection) - Method in interface com.telelogic.rhapsody.core.IRPModelElement

Specifies an RTF string to use as the description for the element, and a collection of elements to which hyperlinks should be created.

setDescriptionHTML(String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement

Specifies the HTML text to use for the description of the model element.

setDescriptionRTF(String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement

Specifies the RTF string to use for the description of the model element.

setDirection(String) - Method in interface com.telelogic.rhapsody.core.IRPFlow

Specifies the direction to use for the flow.

setDirectory(int, String) - Method in interface com.telelogic.rhapsody.core.IRPConfiguration

method setDirectory

setDisplayName(String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement

Specifies the text to use for the label of the model element.

setDisplayNameRTF(String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement

Specifies the RTF string to use for the label of the model element.

setDisplayOption(char, String) - Method in interface com.telelogic.rhapsody.core.IRPHyperLink

setDisplayOption

setDMBoolProperty(String, int) - Method in interface com.telelogic.rhapsody.core.IRPApplication

Sets the value of the specified boolean Design Manager property

setDMProperty(String, String) - Method in interface com.telelogic.rhapsody.core.IRPApplication

Sets the value of the specified Design Manager property

setDocking(int) - Method in interface com.telelogic.rhapsody.core.IRPPlugInWindow

Set docking mode.

setDurationConstraint(String) - Method in interface com.telelogic.rhapsody.core.IRPMessage

Modifies the text of this Duration Constraint.

setDurationObservation(String) - Method in interface com.telelogic.rhapsody.core.IRPMessage

Modifies the text of this Duration Observation.

setDurationTime(String) - Method in interface com.telelogic.rhapsody.core.IRPAcceptTimeEvent

Specifies the duration that should be used for this element.

setElementTypes(IRPCollection) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout

Specifies the list of element types that should be displayed in the table.

setEnd1(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPFlow

set property end1

setEnd1Multiplicity(String) - Method in interface com.telelogic.rhapsody.core.IRPLink

set property end1Multiplicity

setEnd1Name(String) - Method in interface com.telelogic.rhapsody.core.IRPLink

set property end1Name

setEnd1ViaPort(IRPInstance, IRPPort) - Method in interface com.telelogic.rhapsody.core.IRPFlow

method setEnd1ViaPort

setEnd1ViaSysMLPort(IRPInstance, IRPSysMLPort) - Method in interface com.telelogic.rhapsody.core.IRPFlow

method setEnd1ViaSysMLPort

setEnd2(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPFlow

set property end2

setEnd2Multiplicity(String) - Method in interface com.telelogic.rhapsody.core.IRPLink

set property end2Multiplicity

setEnd2Name(String) - Method in interface com.telelogic.rhapsody.core.IRPLink

Used to specify the model element that should be associated with an action block, condition mark, timeout, or canceled timeout, in a sequence diagram.

setFragmentText(String) - Method in interface com.telelogic.rhapsody.core.IRPFileFragment
    set property fragmentText

setFromElementTypes(IRPCollection) - Method in interface com.telelogic.rhapsody.core.IRPMatrixLayout
    Specifies the "from" element types that should be displayed in the matrix.

setFromElementTypes(IRPCollection) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout
    For "relation tables", specifies the list of element types to use as the "from" element types.

setFromElementTypesQueryToUse(IRPTableLayout) - Method in interface com.telelogic.rhapsody.core.IRPMatrixLayout
    Specifies the query to use to determine the "from" element types for the matrix layout.

setFromElementTypesQueryToUse(IRPTableLayout) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout
    For "relation tables", specifies the query to use to determine the "from" element types for the table layout.

setFromElementTypesUseQueryOrElementsList(int) - Method in interface com.telelogic.rhapsody.core.IRPMatrixLayout
    Specifies whether a query or collection of element types should be used to determine the "from" element types for the matrix layout.

setFromElementTypesUseQueryOrElementsList(int) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout
    For "relation tables", specifies whether a query or collection of element types should be used to determine the "from" element types for the table layout.

setFromScope(IRPCollection) - Method in interface com.telelogic.rhapsody.core.IRPMatrixView
    Specifies the "from" scope to use for this matrix view.

setGenerateCodeForActors(int) - Method in interface com.telelogic.rhapsody.core.IRPConfiguration
    set property generateCodeForActors

setGraphicalProperty(String, String) - Method in interface com.telelogic.rhapsody.core.IRPGraphElement
    Sets a new value for a graphical property.

setGraphicalPropertyOfText(String, String, String) - Method in interface com.telelogic.rhapsody.core.IRPGraphElement
    Sets a new value for a graphical property for the specified textual element associated with the graphic element.

setGUID(String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
    Sets a new GUID for the model element.

setHiddenUI(boolean) - Method in interface com.telelogic.rhapsody.core.IRPApplication
    set property hiddenUI

setImageLayout(String) - Method in interface com.telelogic.rhapsody.core.IRPGraphElement
    Used to specify the image layout that should be used for the image linked to the graphic element.

setImplicit() - Method in interface com.telelogic.rhapsody.core.IRPInstance
    method setImplicit

setIncludeDescendants(int) - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
    Specifies whether the scope for the search should include the descendants of the elements specified for the scope, for example, the subpackages of a package that was added to the scope.

setIncludeDescendants(int) - Method in interface com.telelogic.rhapsody.core.IRPTableView
    set property includeDescendants

setIncludeDescendantsFromScope(int) - Method in interface com.telelogic.rhapsody.core.IRPMatrixView
    set property includeDescendantsFromScope

setIncludeDescendantsToScope(int) - Method in interface com.telelogic.rhapsody.core.IRPMatrixView
    set property includeDescendantsToScope

setIncludeInNextLoad(int) - Method in interface com.telelogic.rhapsody.core.IRPUnit

Toggles whether the unit is going to be loaded the next time the model is loaded.

setIncludePath(String) - Method in interface com.telelogic.rhapsody.core.IRPComponent

Specifies the include path to use for the component.

setIncludePath(String) - Method in interface com.telelogic.rhapsody.core.IRPConfiguration

set property includePath

setInitializationCode(String) - Method in interface com.telelogic.rhapsody.core.IRPConfiguration

set property initializationCode

setInitializer(String) - Method in interface com.telelogic.rhapsody.core.IRPOperation

For constructors, used to specify code for the initializer of the operation.

setInitializerArgumentValue(String, String) - Method in interface com.telelogic.rhapsody.core.IRPInstance

method setInitializerArgumentValue

setInstantiatedBy(IRPOperation) - Method in interface com.telelogic.rhapsody.core.IRPInstance

set property instantiatedBy

setInstantiates(IRPRelation) - Method in interface com.telelogic.rhapsody.core.IRPLink

method setInstantiates

setInState(String) - Method in interface com.telelogic.rhapsody.core.IRPObjectNode

Deprecated. *Use addInState instead.*

setInstrumentationType(String) - Method in interface com.telelogic.rhapsody.core.IRPConfiguration

set property instrumentationType

setInteger(int, int) - Method in interface com.telelogic.rhapsody.core.IRPCollection

Used to place an integer in a specific place in a collection.

setInteractionConstraint(String) - Method in interface com.telelogic.rhapsody.core.IRPInteractionOperand

Sets the constraint (guard condition) for the interaction operand.

setInteractionType(String) - Method in interface com.telelogic.rhapsody.core.IRPInteractionOperator

set property interactionType

setInternalTransition(String, String, String) - Method in interface com.telelogic.rhapsody.core.IRPState

method setInternalTransition

setInvariant(String) - Method in interface com.telelogic.rhapsody.core.IRPMessage

Modifies the text of the Invariant field for the state invariant.

setInverse(String, String) - Method in interface com.telelogic.rhapsody.core.IRPRelation

property setInverse

setInvokedOperation(IRPInterfaceItem) - Method in interface com.telelogic.rhapsody.core.IRPSendAction

set property invokedOperation

setIsAbstract(int) - Method in interface com.telelogic.rhapsody.core.IRPClass

Specifies that the class should be abstract.

setIsAbstract(int) - Method in interface com.telelogic.rhapsody.core.IRPOperation

Specifies whether an operation should be defined as abstract.

setIsActive(int) - Method in interface com.telelogic.rhapsody.core.IRPClass

Specifies that the class should be defined as "active", meaning that during execution it runs on its own thread.

setIsAnalysisOnly(int) - Method in interface com.telelogic.rhapsody.core.IRPFlowchart

Specifies whether the activity should be defined as analysis-only.

setIsBehavioral(int) - Method in interface com.telelogic.rhapsody.core.IRPPort

set property isBehavioral

setIsBehaviorOverriden(int) - Method in interface com.telelogic.rhapsody.core.IRPActor

Specifies whether an actor should inherit the behavior defined in the statechart of its base class.

setIsBehaviorOverriden(int) - Method in interface com.telelogic.rhapsody.core.IRPClass

Specifies whether a class should inherit the behavior defined in the statechart of its base class.

setIsBehaviorOverriden(int) - Method in interface com.telelogic.rhapsody.core.IRPUseCase

set property isBehaviorOverriden

setIsClass(int) - Method in interface com.telelogic.rhapsody.core.IRPAssociationClass

Specifies whether the element should be an association class or an association element.

setIsConst(int) - Method in interface com.telelogic.rhapsody.core.IRPOperation

For operations in C++ classes, used to specify whether an operation should be defined as a constant member function.

setIsConstant(int) - Method in interface com.telelogic.rhapsody.core.IRPAttribute

Specifies whether an attribute should be defined as constant.

setIsElaborated(int) - Method in interface com.telelogic.rhapsody.core.IRPTimingDiagram

Specifies whether the diagram should be an elaborated timing diagram or a compact timing diagram.

setIsFinal(int) - Method in interface com.telelogic.rhapsody.core.IRPClass

Specifies that the class should be a final class.

setIsFinal(int) - Method in interface com.telelogic.rhapsody.core.IRPOperation

For operations in Java classes, used to specify whether an operation should be defined as final.

setIsLoadOnDemand(int) - Method in interface com.telelogic.rhapsody.core.IRPApplication

set property isLoadOnDemand

setIsNavigable(int) - Method in interface com.telelogic.rhapsody.core.IRPRelation

set property isNavigable

setIsNewTerm(int) - Method in interface com.telelogic.rhapsody.core.IRPStereotype

Used to change a stereotype to a "new term" stereotype, or change a "new term" stereotype to an ordinary stereotype.

setIsOrdered(int) - Method in interface com.telelogic.rhapsody.core.IRPAttribute

For attributes with multiplicity greater than one, this method is used to specify whether the attribute should be defined as ordered, meaning that the order of the items is significant.

setIsParameter(int) - Method in interface com.telelogic.rhapsody.core.IRPPin

Specifies whether the element should be an activity parameter or an action pin.

setIsReference(int) - Method in interface com.telelogic.rhapsody.core.IRPAttribute

Specifies whether an attribute should be defined as a pointer.

setIsReversed(int) - Method in interface com.telelogic.rhapsody.core.IRPPort

set property isReversed

setIsReversed(int) - Method in interface com.telelogic.rhapsody.core.IRPSysMLPort

Specifies whether the flowport should be conjugated

setIsShowDisplayName(int) - Method in interface com.telelogic.rhapsody.core.IRPModelElement

Specifies whether the label of the element should be displayed instead of the element name whenever the element is used in a diagram.

setIsStatic(int) - Method in interface com.telelogic.rhapsody.core.IRPAttribute

Specifies whether an attribute should be defined as static.

setIsStatic(int) - Method in interface com.telelogic.rhapsody.core.IRPOperation

Specifies whether an operation should be defined as static.

setIsTypedefConstant(int) - Method in interface com.telelogic.rhapsody.core.IRPType

set property isTypedefConstant

setIsTypedefOrdered(int) - Method in interface com.telelogic.rhapsody.core.IRPType

set property isTypedefOrdered

setIsTypedefReference(int) - Method in interface com.telelogic.rhapsody.core.IRPType

set property isTypedefReference

setIsVirtual(int) - Method in interface com.telelogic.rhapsody.core.IRPGeneralization

set property is virtual

setIsVirtual(int) - Method in interface com.telelogic.rhapsody.core.IRPOperation

For operations in C++ or C# classes, used to specify whether an operation should be defined as virtual.

setItsAction(String) - Method in interface com.telelogic.rhapsody.core.IRPTransition

method setItsAction

setItsComponent(IRPComponent) - Method in interface com.telelogic.rhapsody.core.IRPConfiguration

method setItsComponent

setItsGuard(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPTransition</u>
method setItsGuard

setItsLabel(String, String, String) - Method in interface com.telelogic.rhapsody.core.<u>IRPTransition</u>
property setItsLabel

setItsMatrixLayout(IRPMatrixLayout) - Method in interface com.telelogic.rhapsody.core.<u>IRPMatrixView</u>
Specifies the matrix layout to use for this matrix view.

setItsOwner(IRPOperation) - Method in interface com.telelogic.rhapsody.core.<u>IRPFlowchart</u>
Deprecated. *Use IRPModelElement.setOwner instead.*

setItsSource(IRPStateVertex) - Method in interface com.telelogic.rhapsody.core.<u>IRPTransition</u>
set property itsSource

setItsStatechart(IRPStatechart) - Method in interface com.telelogic.rhapsody.core.<u>IRPTransition</u>
set property itsStatechart

setItsSwimlane(IRPSwimlane) - Method in interface com.telelogic.rhapsody.core.<u>IRPConnector</u>
Specifies the swimlane that should contain this connector.

setItsSwimlane(IRPSwimlane) - Method in interface com.telelogic.rhapsody.core.<u>IRPState</u>
Specifies the swimlane that the action should be in

setItsTableLayout(IRPTableLayout) - Method in interface com.telelogic.rhapsody.core.<u>IRPTableView</u>
Specifies the table layout to use for this table view.

setItsTarget(IRPStateVertex) - Method in interface com.telelogic.rhapsody.core.<u>IRPTransition</u>
set property itsTarget

setItsTrigger(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPTransition</u>
method setItsTrigger

setKind(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPType</u>
set property kind

setLanguage(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPApplication</u>
set property Language

setLanguage(String, int) - Method in interface com.telelogic.rhapsody.core.<u>IRPUnit</u>
Specifies the programming language that should be used when code is generated for the unit.

setLibraries(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPComponent</u>
set property libraries

setLibraries(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPConfiguration</u>
set property libraries

setLinkSwitches(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPConfiguration</u>
set property linkSwitches

setLocalLibPath(String) - Method in class com.telelogic.rhapsody.core.<u>RhpClassLoader</u>

setLog(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPApplication</u>
set log file

setLogFile(String) - Static method in class com.telelogic.rhapsody.core.<u>RhapsodyAppServer</u>
Specifies a log file to use for recording API actions.

setMainDiagram(IRPDiagram) - Method in interface com.telelogic.rhapsody.core.<u>IRPModelElement</u>
Specifies the "main" diagram for the element.

setMatchCase(int) - Method in interface com.telelogic.rhapsody.core.<u>IRPSearchQuery</u>
Specifies whether the search should require an exact match in terms of upper and lower case.

setMatchSpecifiedCriteria(int) - Method in interface com.telelogic.rhapsody.core.<u>IRPSearchQuery</u>
Specifies whether the query should return the model elements that match the criteria specified, or the model elements that do not match the criteria specified.

setMatchWholeWord(int) - Method in interface com.telelogic.rhapsody.core.<u>IRPSearchQuery</u>
Specifies whether the search should require whole word matches.

setModelElement(int, IRPModelElement) - Method in interface com.telelogic.rhapsody.core.<u>IRPCollection</u>

Places an item in a specific place in a collection.

setMultiplicities(IRPCollection) - Method in interface com.telelogic.rhapsody.core.IRPContextSpecification
Specifies the collection of indices to use for the model elements in the "value" collection.

setMultiplicity(String) - Method in interface com.telelogic.rhapsody.core.IRPAttribute
Specifies the multiplicity for the attribute.

setMultiplicity(String) - Method in interface com.telelogic.rhapsody.core.IRPRelation
set property multiplicity

setMultiplicity(String) - Method in interface com.telelogic.rhapsody.core.IRPTag
Specifies the multiplicity for the tag.

setName(String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
Sets the specified string as the name of the element.

setNotifyPluginOnElementsChanged(int) - Method in interface com.telelogic.rhapsody.core.IRPProject
For plugins that use the callback API, you must call the method setNotifyPluginOnElementsChanged if you want the plugin to be notified when model elements are modified.

setObjectExplicit(IRPInstance) - Method in interface com.telelogic.rhapsody.core.IRPProject
Changes the specified object to an explicit object.

setObjectImplicit(IRPInstance) - Method in interface com.telelogic.rhapsody.core.IRPProject
Changes the specified object to an implicit object.

setObjID(String) - Method in interface com.telelogic.rhapsody.core.IRPPowListener
method SetObjID

setObjID(String) - Method in interface com.telelogic.rhapsody.core.IRPPowTextListener
method SetObjID

setOfClass(IRPClassifier) - Method in interface com.telelogic.rhapsody.core.IRPRelation
set property ofClass

setOfState(IRPState) - Method in interface com.telelogic.rhapsody.core.IRPConnector
For history connectors, specifies the state for which the connector should maintain historical state information.

setOfTemplate(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
Makes the current model element a template instantiation of the specified template.

setOperation(IRPInterfaceItem) - Method in interface com.telelogic.rhapsody.core.IRPCallOperation
Specifies the operation to use for this call operation element.

setOtherClass(IRPClassifier) - Method in interface com.telelogic.rhapsody.core.IRPRelation
set property otherClass

setOwner(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
Specifies the model element that should be the owner of this element.

setOwnerWithoutChangingDependent(IRPModelElement) - Method in interface
com.telelogic.rhapsody.core.IRPDependency
Specifies a new owner for the dependency, without changing the dependent model element.

setPanelWidgetInstancePath(IRPCollection) - Method in interface
com.telelogic.rhapsody.core.IRPGraphNode
set property panelWidgetInstancePath

setPaneWndTitle(String, String) - Method in class com.telelogic.rhapsody.core.RPowPaneMgrEvents
method SetPaneWndTitle

setParameterKind(String) - Method in interface com.telelogic.rhapsody.core.IRPTemplateParameter
Used to specify the type of the template parameter.

setParent(IRPState) - Method in interface com.telelogic.rhapsody.core.IRPStateVertex
Sets the parent state of the element.

setPath(String) - Method in interface com.telelogic.rhapsody.core.IRPComponent
method setPath

setPath(String) - Method in interface com.telelogic.rhapsody.core.IRPFile
property setPath

S   912

setPinDirection(String) - Method in interface com.telelogic.rhapsody.core.IRPPin
>   Specifies the direction of the pin/parameter.

setPinType(IRPClassifier) - Method in interface com.telelogic.rhapsody.core.IRPPin
>   Specifies the type to use for the value held by the pin/parameter.

setPort(IRPPort) - Method in interface com.telelogic.rhapsody.core.IRPMessage
>   set property Port

setPortDirection(String) - Method in interface com.telelogic.rhapsody.core.IRPSysMLPort
>   Sets the direction of the flowport.

setPosString(String) - Method in interface com.telelogic.rhapsody.core.IRPPlugInWindow
>   Set position string

setProcessSubTaskName(int, int, String) - Method in class
com.telelogic.rhapsody.core.RPExternalIDEManager
>   Set a Progress Task subtask's name

setPropertyValue(String, String) - Method in interface com.telelogic.rhapsody.core.IRPGraphElement
>   method setPropertyValue

setPropertyValue(String, String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement
>   Sets the value of a property for the model element.

setQualifier(String) - Method in interface com.telelogic.rhapsody.core.IRPRelation
>   set property qualifier

setQualifierType(IRPClassifier) - Method in interface com.telelogic.rhapsody.core.IRPRelation
>   Sets the type to use for the qualifier for the association.

setReadOnly(int) - Method in interface com.telelogic.rhapsody.core.IRPUnit
>   Toggles the read-only status of the file used to store the unit.

setReferencedSequenceDiagram(IRPSequenceDiagram) - Method in interface
com.telelogic.rhapsody.core.IRPClassifierRole
>   set property referencedSequenceDiagram

setReferenceSequenceDiagram(IRPSequenceDiagram) - Method in interface
com.telelogic.rhapsody.core.IRPInteractionOccurrence
>   set property referenceSequenceDiagram

setReferenceToActivity(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPState
>   For call behavior elements, sets the activity that is referenced by the element.

setRelationLabel(String) - Method in interface com.telelogic.rhapsody.core.IRPRelation
>   set property relationLabel

setRelationLinkName(String) - Method in interface com.telelogic.rhapsody.core.IRPRelation
>   set property relationLinkName

setRelationRoleName(String) - Method in interface com.telelogic.rhapsody.core.IRPRelation
>   set property relationRoleName

setRelationTable(int) - Method in interface com.telelogic.rhapsody.core.IRPTableLayout
>   Specifies whether the table should be defined as a "relation table".

setRelationType(String) - Method in interface com.telelogic.rhapsody.core.IRPRelation
>   set property relationType

setReleaseInterfacesOnGBMode(boolean) - Static method in class
com.telelogic.rhapsody.core.RhapsodyAppServer
>   For internal use only.

setRemoteRequirementsPopulateMode(String) - Method in interface
com.telelogic.rhapsody.core.IRPPackage
>   For collections of remote requirements, you can use setRemoteRequirementsPopulateMode to specify
>   which requirements in the collection should be loaded when you open the model - all the
>   requirements, only the requirements that have OSLC links to model elements, or none of the
>   requirements.

S

setRepresentative(IRPModelElement) - Method in interface
com.telelogic.rhapsody.core.IRPTemplateParameter
  For internal use only.

setRepresents(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPObjectNode
  Specifies the class/type that this object node should represent.

setRepresents(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPSwimlane
  set property represents

setRequirementID(String) - Method in interface com.telelogic.rhapsody.core.IRPRequirement
  Sets the ID for the requirement.

setRequirementTraceabilityHandle(int) - Method in interface
com.telelogic.rhapsody.core.IRPModelElement
  Sets a new ID to be used to reference this requirement

setReturns(IRPClassifier) - Method in interface com.telelogic.rhapsody.core.IRPOperation
  Specifies the return type of the operation.

setReturnTypeDeclaration(String) - Method in interface com.telelogic.rhapsody.core.IRPOperation
  Creates an on-the-fly type to use as the return type of the operation, using the declaration that you
  provide as a parameter.

setReturnValue(String) - Method in interface com.telelogic.rhapsody.core.IRPMessage
  set property returnValue

setSavedInSeperateDirectory(int) - Method in interface com.telelogic.rhapsody.core.IRPPackage
  Specifies whether the package should be saved in a separate directory.

setScope(IRPCollection) - Method in interface com.telelogic.rhapsody.core.IRPTableView
  Specifies the scope to use for this table view.

setScopeBySelectedElements(int) - Method in interface com.telelogic.rhapsody.core.IRPComponent
  set toggle the scope between selected and all-elements

setScopeType(String) - Method in interface com.telelogic.rhapsody.core.IRPConfiguration
  set property scopeType

setSearchFindAsOption(char) - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
  Sets the type of search that should be used for the search text - regular text, wildcard, regular
  expression, or empty string.

setSearchScopeObject(IRPModelElement) - Method in interface
com.telelogic.rhapsody.core.IRPSearchQuery
  Deprecated. *This method, used to set the scope for the search, was introduced when Rhapsody only*
  allowed you to specify a single element as the scope. Now that Rhapsody allows you to specify a list of
  such elements, you should use the method
  IRPSearchQuery.addSearchScope(com.telelogic.rhapsody.core.IRPModelElement).

setSearchText(String) - Method in interface com.telelogic.rhapsody.core.IRPSearchQuery
  Specifies the text that should be searched for.

setSelectedImage(String) - Method in interface com.telelogic.rhapsody.core.IRPGraphElement
  Links the graphic element to the image represented by the path specified.

setSeparateSaveUnit(int) - Method in interface com.telelogic.rhapsody.core.IRPUnit
  Specifies whether the current IRPUnit object should be saved in its own file.

setShowDiagramFrame(int) - Method in interface com.telelogic.rhapsody.core.IRPDiagram
  Shows/hides the diagram frame.

setShowDiagramFrame(int) - Method in interface com.telelogic.rhapsody.core.IRPStatechart
  Shows/hides the diagram frame.

setSize(int) - Method in interface com.telelogic.rhapsody.core.IRPCollection
  Sets the size of a collection.

setSlotProperty(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPInstanceSlot
  set property slotProperty

setSpecification(String) - Method in interface com.telelogic.rhapsody.core.IRPAnnotation

Adds a specification to the annotation.

setSpecificationRTF(String) - Method in interface com.telelogic.rhapsody.core.IRPAnnotation

Specifies RTF string to use for the specification of the annotation.

setStandardHeaders(String) - Method in interface com.telelogic.rhapsody.core.IRPComponent

Specifies the standard headers for the component.

setStandardHeaders(String) - Method in interface com.telelogic.rhapsody.core.IRPConfiguration

set property standardHeaders

setStatechartImplementation(String) - Method in interface com.telelogic.rhapsody.core.IRPConfiguration

set property statechartImplementation

setStateType(String) - Method in interface com.telelogic.rhapsody.core.IRPState

Specifies the type of the state

setStaticReaction(String, String, String) - Method in interface com.telelogic.rhapsody.core.IRPState

Adds a new internal transition to the state.

setStereotype(IRPStereotype) - Method in interface com.telelogic.rhapsody.core.IRPModelElement

Applies the specified stereotype to the element.

setString(int, String) - Method in interface com.telelogic.rhapsody.core.IRPCollection

Used to place a String in a specific place in a collection.

setSuperEvent(IRPEvent) - Method in interface com.telelogic.rhapsody.core.IRPEvent

set property baseEvent

setTagContextValue(IRPTag, IRPCollection, IRPCollection) - Method in interface
com.telelogic.rhapsody.core.IRPModelElement

Applies the specified tag to the model element, and sets the value of the tag to a specific instance of
another model element.

setTagContextValue(IRPCollection, IRPCollection) - Method in interface
com.telelogic.rhapsody.core.IRPTag

Sets the value of the tag to a specific instance of another model element.

setTagElementValue(IRPTag, IRPModelElement) - Method in interface
com.telelogic.rhapsody.core.IRPModelElement

Applies a tag whose type is a model element to the current element with the value specified.

setTagMetaClass(String) - Method in interface com.telelogic.rhapsody.core.IRPTag

Specifies the metaclass to which the tag should be applicable, for example, "Class".

setTagValue(IRPTag, String) - Method in interface com.telelogic.rhapsody.core.IRPModelElement

Applies the specified tag to the model element with the value specified.

setTarget(IRPRelation) - Method in interface com.telelogic.rhapsody.core.IRPCallOperation

Specifies the target to use for this call operation element.

setTarget(String) - Method in interface com.telelogic.rhapsody.core.IRPControlledFile

Specifies a different file to associate with the Controlled File element.

setTarget(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPHyperLink

Sets the specified model element to be the target of the hyperlink.

setTarget(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.IRPSendAction

Sets the specified model element to be the target of the Send Action element.

setTi(IRPTemplateInstantiation) - Method in interface com.telelogic.rhapsody.core.IRPModelElement

For internal use only.

setTimeConstraint(String) - Method in interface com.telelogic.rhapsody.core.IRPMessage

Modifies the text of this Time Constraint.

setTimeModel(String) - Method in interface com.telelogic.rhapsody.core.IRPConfiguration

set property timeModel

setTimeObservation(String) - Method in interface com.telelogic.rhapsody.core.IRPMessage

Modifies the text of this Time Observation.

setTimerValue(String) - Method in interface com.telelogic.rhapsody.core.IRPMessage

set property timerValue

S

Specifies the collection of strings that represents the model elements that constitute the full path to the element.

setValue(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPEnumerationLiteral</u>
> set property value

setValue(IRPModelElement) - Method in interface com.telelogic.rhapsody.core.<u>IRPInstanceValue</u>
> Sets the value to store.

setValue(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPLiteralSpecification</u>
> Sets the value to store.

setValue(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPTag</u>
> Sets the value of the tag.

setVariant(IRPModelElement, IRPModelElement) - Method in interface com.telelogic.rhapsody.core.<u>IRPComponent</u>
> method setVariant

setViewIncludeModelElements(int) - Method in interface com.telelogic.rhapsody.core.<u>IRPSearchQuery</u>
> Specifies whether the search results should also include model elements that were found by the search but are not referenced in any of the views that you specified.

setViewsToSearch(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPSearchQuery</u>
> Specifies which views (tables, matrices, and diagrams) should be searched - all, none, all open, or just the views that were specified with the methods addDiagramToViewsList, addTableToViewsList, and addMatrixToViewsList.

setVisibility(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPAttribute</u>
> Specifies the visibility of the operation.

setVisibility(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPGeneralization</u>
> set property visibility

setVisibility(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPOperation</u>
> set property visibility

setWaitDialogWatchdogValue(String) - Method in interface com.telelogic.rhapsody.core.<u>IRPProject</u>
> The method setWaitDialogWatchdogValue provides a mechanism that allows an external process to inform Rhapsody that the process has ended or crashed.

shouldAbortCodeGeneration() - Method in interface com.telelogic.rhapsody.core.<u>IRPBaseExternalCodeGeneratorTool</u>
> method shouldAbortCodeGeneration

shouldAddAggregate(IRPModelElement, IRPModelElement) - Method in class com.telelogic.rhapsody.core.<u>RPExternalRoundtrip</u>
> method ShouldAddAggregate

shouldMergeAggregate(IRPModelElement, IRPModelElement, IRPModelElement, IRPModelElement) - Method in class com.telelogic.rhapsody.core.<u>RPExternalRoundtrip</u>
> method ShouldMergeAggregate

shouldMergeAssociation(String, IRPModelElement, IRPModelElement, IRPModelElement, IRPModelElement) - Method in class com.telelogic.rhapsody.core.<u>RPExternalRoundtrip</u>
> method ShouldMergeAssociation

shouldMergeAttribute(String, String, String, IRPModelElement, IRPModelElement, String) - Method in class com.telelogic.rhapsody.core.<u>RPExternalRoundtrip</u>
> method ShouldMergeAttribute

shouldRemoveAggregate(IRPModelElement, IRPModelElement, IRPModelElement) - Method in class com.telelogic.rhapsody.core.<u>RPExternalRoundtrip</u>
> method ShouldRemoveAggregate

SHOW_UNRESOLVED - Static variable in class com.telelogic.rhapsody.core.<u>IRPSearchQuery.UnresolvedKind</u>

showAllPorts() - Method in interface com.telelogic.rhapsody.core.<u>IRPGraphNode</u>

method showAllPorts

showBrowser(int) - Method in class com.telelogic.rhapsody.core.RPExternalIDEManager
ShowBrowser

showPaneWnd(String) - Method in class com.telelogic.rhapsody.core.RPowPaneMgrEvents
method ShowPaneWnd

showStatusBarMessage(String) - Method in class com.telelogic.rhapsody.core.RPExternalIDEManager
Display message in status bar

showWindow(int) - Method in interface com.telelogic.rhapsody.core.IRPPlugInWindow
Show or hide window

simplify(IRPModelElement, IRPModelElement, String) - Method in class
com.telelogic.rhapsody.core.RPCodeGenSimplifier
simplify the user element

SPECIFICATION - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.AnnotationAttribute
Value to be used for Property parameter of addColumn method, when AnnotationAttribute is selected
for the Type parameter of addColumn method.

SPECIFICATION - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.RequirementAttribute
Value to be used for Property parameter of addColumn method, when RequirementAttribute is
selected for the Type parameter of addColumn method.

startTransactionOfNoCGInterest() - Method in interface com.telelogic.rhapsody.core.IRPProject
For internal use only.

startUndoTransaction() - Method in interface com.telelogic.rhapsody.core.IRPApplication
start undo transaction

STEREOTYPES - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.GeneralAttribute
Value to be used for Property parameter of addColumn method, when GeneralAttribute is selected for
the Type parameter of addColumn method.

STRING - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.ImplementationCellType
Value to be used for cellType parameter of SetColumnImplementationCellType method.

STRUCTURED - Static variable in class com.telelogic.rhapsody.core.IRPGraphElement.ImageLayout
Show image in structured layout

subscribedTo() - Method in class com.telelogic.rhapsody.core.RPIntegratorListener
Get the application's name that Rhapsody integrates with

syncBuild() - Method in interface com.telelogic.rhapsody.core.IRPApplication
syncBuild

synchronizeTemplateInstantiation() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
After changes are made to a template, this method can be called on each instantiation of the template
in order to update the instantiation to match the changes that were made to the template.

T

TAG - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column
  Value used for Type parameter of addColumn method
TAG_EDIT - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column
  Value used for Type parameter of addColumn method
TAG_EDIT_STRICT - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column
  Value used for Type parameter of addColumn method
TAG_VALUE - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

terminateApplication() - Method in interface com.telelogic.rhapsody.core.IRPApplication
  Terminate the Application
TEXT_FRAGMENT - Static variable in class com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

tick(int) - Method in interface com.telelogic.rhapsody.core.IRPProgressBar
  method tick
TO_ELEMENT - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.RelationAttributeTo
  Value to be used for Property parameter of addColumn method, when RelationAttributeTo is selected
  for the Type parameter of addColumn method.
toList() - Method in interface com.telelogic.rhapsody.core.IRPCollection
  Returns a java.util.List populated with the elements in the collection.
TRACE - Static variable in class com.telelogic.rhapsody.core.IRPModelElement.OSLCLink.Types
  OSLC Link Type: Trace
TRANSITION_LABEL - Static variable in class
com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

TYPE_DECLARATIONS_AND_REFERENCES - Static variable in class
com.telelogic.rhapsody.core.IRPSearchQuery.SearchInField

U

UNDEFINED_RELATION - Static variable in class
com.telelogic.rhapsody.core.IRPSearchQuery.References.RelationKind

undo() - Method in interface com.telelogic.rhapsody.core.IRPApplication
>   Perform Undo

unload() - Method in interface com.telelogic.rhapsody.core.IRPUnit
>   Unloads the unit.

unloadFromTarget() - Method in interface com.telelogic.rhapsody.core.IRPApplication
>   unloadFromTarget

unloadPlugin(String, int) - Method in class com.telelogic.rhapsody.core.RPJavaPluginsManager
>   Unload plugin

unlockOnDesignManager() - Method in interface com.telelogic.rhapsody.core.IRPModelElement
>   Unlocks the model element on the Design Manager server.

unoverrideInheritance() - Method in interface com.telelogic.rhapsody.core.IRPState
>   Restores the inheritance relationship between this state and the corresponding state from the statechart
>   of the class that this class is derived from.

unoverrideInheritance() - Method in interface com.telelogic.rhapsody.core.IRPStatechart
>   Restores the inheritance relationship between this statechart and the statechart of the base class.

unoverrideInheritance() - Method in interface com.telelogic.rhapsody.core.IRPTransition
>   method unoverrideInheritance

unRegisterAsActiveObject(IRPApplication) - Static method in class
com.telelogic.rhapsody.core.RhapsodyAppServer
>   For internal use only.

unregisterCOMClient(int, String, int) - Method in interface com.telelogic.rhapsody.core.IRPApplication
>   unregister COM client

USER_DEFINED_METHOD - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column
>   Value used for Type parameter of addColumn method.

V

VALUE - Static variable in class com.telelogic.rhapsody.core.IRPTableLayout.Column.GeneralAttribute

version() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> Get Rhapsody version

versionNumberLong() - Method in interface com.telelogic.rhapsody.core.IRPApplication
> Get Rhapsody versionNumberLong

VIA_PORT - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.RelationAttributeFrom
> Value to be used for Property parameter of addColumn method, when RelationAttributeFrom is selected for the Type parameter of addColumn method.

VIA_PORT - Static variable in class
com.telelogic.rhapsody.core.IRPTableLayout.Column.RelationAttributeTo
> Value to be used for Property parameter of addColumn method, when RelationAttributeTo is selected for the Type parameter of addColumn method.

W

whoAmI() - Method in class com.telelogic.rhapsody.core.RPExternalCodeGenerator
  method WhoAmI
write(String) - Method in interface com.telelogic.rhapsody.core.IRPASCIIFile
  write to file
writeCodeGenMessage(String) - Method in interface
com.telelogic.rhapsody.core.IRPBaseExternalCodeGeneratorTool
  method writeCodeGenMessage
writeToLog(String) - Static method in class com.telelogic.rhapsody.core.RhapsodyAppServer
  Writes the specified text to the Rhapsody API log file.
writeToOutputWindow(String, String) - Method in interface com.telelogic.rhapsody.core.IRPApplication
  Writes text to Rhapsody's Output window.

X

XML - Static variable in class com.telelogic.rhapsody.core.IRPMatrixView.ContentFormat
     Export in XML format.
XML - Static variable in class com.telelogic.rhapsody.core.IRPTableView.ContentFormat
     Export in XML format.

Deprecated API

Contents

- Deprecated Fields
- Deprecated Methods

Deprecated Fields

com.telelogic.rhapsody.core.IRPTableLayout.Column.RELAION_ATTRIBUTE_FROM
com.telelogic.rhapsody.core.IRPTableLayout.Column.RELAION_ATTRIBUTE_TO

Deprecated Methods

com.telelogic.rhapsody.core.IRPObjectNode.getInState() *Use getInStateList() instead.*
com.telelogic.rhapsody.core.IRPFlowchart.getItsOwner() *Use IRPModelElement.getOwner instead.*
com.telelogic.rhapsody.core.IRPSearchQuery.getSearchScopeObject() *This method, used to return the scope specified for the search, was introduced when Rhapsody only allowed you to specify a single element as the scope. Now that Rhapsody allows you to specify a list of such elements, you should use the method IRPSearchQuery.getSearchScopeElements().*
com.telelogic.rhapsody.core.IRPModelElement.getStereotype() *Since Rhapsody now allows multiple stereotypes to be applied to a model element, the getStereotypes( ) method should be used instead.*
com.telelogic.rhapsody.core.IRPObjectNode.setInState(String) *Use addInState instead.*
com.telelogic.rhapsody.core.IRPFlowchart.setItsOwner(IRPOperation) *Use IRPModelElement.setOwner instead.*
com.telelogic.rhapsody.core.IRPSearchQuery.setSearchScopeObject(IRPModelElement) *This method, used to set the scope for the search, was introduced when Rhapsody only allowed you to specify a single element as the scope. Now that Rhapsody allows you to specify a list of such elements, you should use the method IRPSearchQuery.addSearchScope(com.telelogic.rhapsody.core.IRPModelElement).*

Constant Field Values

Contents

- com.telelogic.*

com.telelogic.*

com.telelogic.rhapsody.core.**HYPNameType**		
public static final char	RP_HYP_FREETEXT	0
public static final char	RP_HYP_LABELTEXT	2
public static final char	RP_HYP_NAMETEXT	1
public static final char	RP_HYP_TAGVALUETEXT	3

com.telelogic.rhapsody.core.**IRPApplication.AddToModel_Mode**		
public static final int	AS_REFERENCE	0
public static final int	AS_UNIT_WITH_COPY	1
public static final int	AS_UNIT_WITHOUT_COPY	2

com.telelogic.rhapsody.core.**IRPGraphElement.ImageLayout**		
public static final java.lang.String	COMPARTMENT	"Compartment"
public static final java.lang.String	IMAGE_ONLY_SHOW_NAME	"Image Only Show Name"
public static final java.lang.String	IMAGE_ONLY_WITHOUT_NAME	"Image Only Without Name"
public static final java.lang.String	STRUCTURED	"Structured"

com.telelogic.rhapsody.core.**IRPMatrixLayout.QueryOrElementsList**		
public static final int	ELEMENTS_LIST	0
public static final int	QUERY	1

com.telelogic.rhapsody.core.**IRPMatrixView.ContentFormat**		
public static final java.lang.String	CSV	"CSV"
public static final java.lang.String	HTML	"HTML"
public static final java.lang.String	XML	"XML"

com.telelogic.rhapsody.core.**IRPModelElement.OSLCLink.Types**		
public static final java.lang.String	DERIVES	"http://jazz.net/ns/dm/linktypes#derives"
public static final java.lang.String	EXTERNAL	"http://jazz.net/ns/dm/linktypes#external"
public static final java.lang.String	REFINE	"http://jazz.net/ns/dm/linktypes#refine"
public static final java.lang.String	SATISFY	"http://jazz.net/ns/dm/linktypes#satisfy"
public static final java.lang.String	TRACE	"http://jazz.net/ns/dm/linktypes#trace"

com.telelogic.rhapsody.core.**IRPSearchQuery.References.QuantityOperator**		
public static final java.lang.String	EXACTLY	"Exactly"
public static final java.lang.String	LESS_THAN	"Less than"
public static final java.lang.String	MORE_THAN	"More than"

com.telelogic.rhapsody.core.**IRPSearchQuery.References.RelationKind**		
public static final java.lang.String	AGGREGATE	"Aggregate"
public static final java.lang.String	DIAGRAM_ELEMENT	"Diagram element"
public static final java.lang.String	INCOMING_RELATION	"Incoming relation"
public static final java.lang.String	OUTGOING_RELATION	"Outgoing relation"
public static final java.lang.String	REFERENCE	"Reference"
public static final java.lang.String	UNDEFINED_RELATION	"Undefined relation"

com.telelogic.rhapsody.core.**IRPSearchQuery.SearchInField**		
public static final java.lang.String	COMMENT_SPECIFICATION	"Comment specification"
public static final java.lang.String	CONFIGURATION_INITIALIZATION	"Configuration initialization"
public static final java.lang.String	CONSTRAINT_SPECIFICATION	"Constraint specification"
public static final java.lang.String	DESCRIPTIONS	"Descriptions"
public static final java.lang.String	ENUMERATION_LITERAL_VALUE	"Enumerationliteral value"
public static final java.lang.String	GROUP_ALL	"<All>"
public static final java.lang.String	GROUP_CODE	"<User code (Operation bodies, actions, etc.)>"
public static final java.lang.String	GROUP_ELEMENT_NAME	"<Element Name>"
public static final java.lang.String	GROUP_OTHER_TEXT	"<Other text (Descreptions, label, specification, etc.)>"
public static final java.lang.String	INITIAL_VALUE	"Initial value(attribute,argument)"
public static final java.lang.String	LABEL	"Label"
public static final java.lang.String	LOCALLY_OVERRIDDEN_PROPERTY	"Locally overridden property"
public static final java.lang.String	MULTIPLICITY	"Multiplicity"
public static final java.lang.String	NAME	"Name"

com.telelogic.rhapsody.core.**IRPSearchQuery.SearchInField**

public static final java.lang.String	NOTES_AND_TEXT	"Notes and Text"
public static final java.lang.String	OPERATION_BODIES	"Operation bodies"
public static final java.lang.String	REQUIREMENT_ID	"Requirement ID"
public static final java.lang.String	REQUIREMENT_SPECIFICATION	"Requirement specification"
public static final java.lang.String	TAG_VALUE	"Tag value"
public static final java.lang.String	TEXT_FRAGMENT	"TextFragment"
public static final java.lang.String	TRANSITION_LABEL	"Transition label(action,guard,trigger)"
public static final java.lang.String	TYPE_DECLARATIONS_AND_REFERENCES	"Type declarations and references"

com.telelogic.rhapsody.core.**IRPSearchQuery.SubQueriesOperator**

public static final java.lang.String	AND	"And"
public static final java.lang.String	OR	"Or"

com.telelogic.rhapsody.core.**IRPSearchQuery.UnresolvedKind**

public static final java.lang.String	IGNORE_UNRESOLVED	"Ignore unresolved"
public static final java.lang.String	ONLY_UNRESOLVED_OR_UNLOADED	"Only unresolved or unloaded"
public static final java.lang.String	SHOW_UNRESOLVED	"Show unresolved"

com.telelogic.rhapsody.core.**IRPSearchQuery.ViewsToSearch**

public static final java.lang.String	ALL	"All"
public static final java.lang.String	DETAILED	"Detailed"
public static final java.lang.String	NONE	"None"
public static final java.lang.String	OPEN	"Open"

com.telelogic.rhapsody.core.**IRPTableLayout.Column**

public static final java.lang.String	ANNOTATION_ATTRIBUTE	"Annotation Attribute"
public static final java.lang.String	DEPENDS_ON	"Depends On"
public static final java.lang.String	FLOW_ATTRIBUTE	"Flow Attribute"
public static final java.lang.String	GENERAL_ATTRIBUTE	"General Attribute"
public static final java.lang.String	RELAION_ATTRIBUTE_FROM	"Relation Attribute (From)"
public static final java.lang.String	RELAION_ATTRIBUTE_TO	"Relation Attribute (To)"
public static final java.lang.String	RELATION_ATTRIBUTE_FROM	"Relation Attribute (From)"
public static final java.lang.String	RELATION_ATTRIBUTE_TO	"Relation Attribute (To)"
public static final java.lang.String	REQUIREMENT_ATTRIBUTE	"Requirement Attribute"
public static final java.lang.String	TAG	"Tag"

com.telelogic.rhapsody.core

com.telelogic.rhapsody.core.**IRPTableLayout.Column**		
public static final java.lang.String	TAG_EDIT	"Tag"
public static final java.lang.String	TAG_EDIT_STRICT	"Tag (Strict)"
public static final java.lang.String	USER_DEFINED_METHOD	"User Defined Method"

com.telelogic.rhapsody.core.**IRPTableLayout.Column.AnnotationAttribute**		
public static final java.lang.String	ID	"ID"
public static final java.lang.String	SPECIFICATION	"Specification"

com.telelogic.rhapsody.core.**IRPTableLayout.Column.DependsOn**		
public static final java.lang.String	DEPENDENCY	"Dependency"

com.telelogic.rhapsody.core.**IRPTableLayout.Column.FlowAttribute**		
public static final java.lang.String	ITEM_FLOWS	"Item Flows"

com.telelogic.rhapsody.core.**IRPTableLayout.Column.GeneralAttribute**		
public static final java.lang.String	CLASSIFIER	"Classifier"
public static final java.lang.String	DESCRIPTION	"Description"
public static final java.lang.String	ELEMENT_TYPE	"Element type"
public static final java.lang.String	FULL_PATH_NAME	"Full path name"
public static final java.lang.String	LABEL	"Label"
public static final java.lang.String	NAME	"Name"
public static final java.lang.String	OWNER	"Owner"
public static final java.lang.String	STEREOTYPES	"Stereotypes"
public static final java.lang.String	VALUE	"Value"

com.telelogic.rhapsody.core.**IRPTableLayout.Column.ImplementationCellType**		
public static final java.lang.String	LIST_OF_MODEL_ELEMENTS	"List of model elements"
public static final java.lang.String	MODEL_ELEMENT	"Model element"
public static final java.lang.String	STRING	"String"

com.telelogic.rhapsody.core.**IRPTableLayout.Column.RelationAttributeFrom**		
public static final java.lang.String	FROM_ELEMENT	"From element"
public static final java.lang.String	PORT_PROVIDED_INERFACE	"Port provided interface"
public static final java.lang.String	PORT_REQUIRED_INERFACE	"Port required interface"
public static final java.lang.String	PROVIDED_INERFACE_OPERATIONS	"Provided interface operations"
public static final java.lang.String	REQUIRED_INERFACE_OPERATIONS	"Required interface operations"

com.telelogic.rhapsody.core.**IRPTableLayout.Column.RelationAttributeFrom**		
public static final java.lang.String	VIA_PORT	"Via port"

com.telelogic.rhapsody.core.**IRPTableLayout.Column.RelationAttributeTo**		
public static final java.lang.String	PORT_PROVIDED_INERFACE	"Port provided interface"
public static final java.lang.String	PORT_REQUIRED_INERFACE	"Port required interface"
public static final java.lang.String	PROVIDED_INERFACE_OPERATIONS	"Provided interface operations"
public static final java.lang.String	REQUIRED_INERFACE_OPERATIONS	"Required interface operations"
public static final java.lang.String	TO_ELEMENT	"To element"
public static final java.lang.String	VIA_PORT	"Via port"

com.telelogic.rhapsody.core.**IRPTableLayout.Column.RequirementAttribute**		
public static final java.lang.String	ID	"ID"
public static final java.lang.String	LINK_FROM	"Link From"
public static final java.lang.String	LINK_FROM_FULLNAME	"Link From FullName"
public static final java.lang.String	LINK_SUSPECT	"Link Suspect"
public static final java.lang.String	LINK_TYPE	"Link Type"
public static final java.lang.String	SPECIFICATION	"Specification"

com.telelogic.rhapsody.core.**IRPTableLayout.Column.UserDefinedMethod**		
public static final java.lang.String	Implementation	"Implementation..."

com.telelogic.rhapsody.core.**IRPTableLayout.QueryOrElementsList**		
public static final int	ELEMENTS_LIST	0
public static final int	QUERY	1

com.telelogic.rhapsody.core.**IRPTableView.ContentFormat**		
public static final java.lang.String	CSV	"CSV"
public static final java.lang.String	HTML	"HTML"
public static final java.lang.String	XML	"XML"

com.telelogic.rhapsody.core.**SearchFindAsEnum**		
public static final char	RP_SEARCH_EMPTY_ONLY	3
public static final char	RP_SEARCH_EXACT	0
public static final char	RP_SEARCH_REGEX	2
public static final char	RP_SEARCH_WILDCARD	1

Hierarchy For Package com.telelogic.rhapsody.core

Class Hierarchy

- java.lang.Object
    - ♦ java.lang.ClassLoader
        - ◊ java.security.SecureClassLoader
            - · java.net.URLClassLoader
                - com.telelogic.rhapsody.core.**RhpClassLoader**
    - ♦ com.telelogic.rhapsody.core.**HYPNameType**
    - ♦ com.telelogic.rhapsody.core.**IRPApplication.AddToModel_Mode**
    - ♦ com.telelogic.rhapsody.core.**IRPGraphElement.ImageLayout**
    - ♦ com.telelogic.rhapsody.core.**IRPMatrixLayout.QueryOrElementsList**
    - ♦ com.telelogic.rhapsody.core.**IRPMatrixView.ContentFormat**
    - ♦ com.telelogic.rhapsody.core.**IRPModelElement.OSLCLink**
    - ♦ com.telelogic.rhapsody.core.**IRPModelElement.OSLCLink.Types**
    - ♦ com.telelogic.rhapsody.core.**IRPSearchQuery.References**
    - ♦ com.telelogic.rhapsody.core.**IRPSearchQuery.References.QuantityOperator**
    - ♦ com.telelogic.rhapsody.core.**IRPSearchQuery.References.RelationKind**
    - ♦ com.telelogic.rhapsody.core.**IRPSearchQuery.SearchInField**
    - ♦ com.telelogic.rhapsody.core.**IRPSearchQuery.SubQueriesOperator**
    - ♦ com.telelogic.rhapsody.core.**IRPSearchQuery.UnresolvedKind**
    - ♦ com.telelogic.rhapsody.core.**IRPSearchQuery.ViewsToSearch**
    - ♦ com.telelogic.rhapsody.core.**IRPTableLayout.Column**
    - ♦ com.telelogic.rhapsody.core.**IRPTableLayout.Column.AnnotationAttribute**
    - ♦ com.telelogic.rhapsody.core.**IRPTableLayout.Column.DependsOn**
    - ♦ com.telelogic.rhapsody.core.**IRPTableLayout.Column.FlowAttribute**
    - ♦ com.telelogic.rhapsody.core.**IRPTableLayout.Column.GeneralAttribute**
    - ♦ com.telelogic.rhapsody.core.**IRPTableLayout.Column.ImplementationCellType**
    - ♦ com.telelogic.rhapsody.core.**IRPTableLayout.Column.RelationAttributeFrom**
    - ♦ com.telelogic.rhapsody.core.**IRPTableLayout.Column.RelationAttributeTo**
    - ♦ com.telelogic.rhapsody.core.**IRPTableLayout.Column.RequirementAttribute**
    - ♦ com.telelogic.rhapsody.core.**IRPTableLayout.Column.UserDefinedMethod**
    - ♦ com.telelogic.rhapsody.core.**IRPTableLayout.QueryOrElementsList**
    - ♦ com.telelogic.rhapsody.core.**IRPTableView.ContentFormat**
    - ♦ com.telelogic.rhapsody.core.**RhapsodyAppServer**
    - ♦ com.telelogic.rhapsody.core.**RhpUtils**
    - ♦ com.telelogic.rhapsody.core.**RPApplicationListener**
    - ♦ com.telelogic.rhapsody.core.**RPCodeGeneratorListener**
    - ♦ com.telelogic.rhapsody.core.**RPCodeGenSimplifier**
    - ♦ com.telelogic.rhapsody.core.**RPExtendedRPClassesFactory**
    - ♦ com.telelogic.rhapsody.core.**RPExternalCheck**
    - ♦ com.telelogic.rhapsody.core.**RPExternalCodeGenerator**
    - ♦ com.telelogic.rhapsody.core.**RPExternalIDEManager**

com.telelogic.rhapsody.core

- ♦ com.telelogic.rhapsody.core.**RPExternalRoundtrip**
- ♦ com.telelogic.rhapsody.core.**RPIntegratorListener**
- ♦ com.telelogic.rhapsody.core.**RPJavaPluginsManager**
- ♦ com.telelogic.rhapsody.core.**RPowPaneMgrEvents**
- ♦ com.telelogic.rhapsody.core.**RPRoundTripListener**
- ♦ com.telelogic.rhapsody.core.**RPRTCListener**
- ♦ com.telelogic.rhapsody.core.**RPSearchListener**
- ♦ com.telelogic.rhapsody.core.**RPUserPlugin**
- ♦ com.telelogic.rhapsody.core.**SearchFindAsEnum**
- ♦ java.lang.Throwable (implements java.io.Serializable)
    - ◊ java.lang.Exception
        - · java.lang.RuntimeException
            - • com.telelogic.rhapsody.core.**RhapsodyRuntimeException**

Interface Hierarchy

- • com.telelogic.rhapsody.core.**IRPApplication**
- • com.telelogic.rhapsody.core.**IRPASCIIFile**
- • com.telelogic.rhapsody.core.**IRPAXViewCtrl**
- • com.telelogic.rhapsody.core.**IRPBaseExternalCodeGeneratorTool**
    - ♦ com.telelogic.rhapsody.core.**IRPCodeGenSimplifiersRegistry**
    - ♦ com.telelogic.rhapsody.core.**IRPExternalCodeGeneratorInvoker**
- • com.telelogic.rhapsody.core.**IRPCodeGenerator**
- • com.telelogic.rhapsody.core.**IRPCollection**
- • com.telelogic.rhapsody.core.**IRPDiagSynthAPI**
- • com.telelogic.rhapsody.core.**IRPExternalCheckRegistry**
- • com.telelogic.rhapsody.core.**IRPExternalIDERegistry**
- • com.telelogic.rhapsody.core.**IRPExternalRoundtripInvoker**
- • com.telelogic.rhapsody.core.**IRPGraphElement**
    - ♦ com.telelogic.rhapsody.core.**IRPGraphEdge**
    - ♦ com.telelogic.rhapsody.core.**IRPGraphNode**
- • com.telelogic.rhapsody.core.**IRPGraphicalProperty**
- • com.telelogic.rhapsody.core.**IRPImageMap**
- • com.telelogic.rhapsody.core.**IRPIntegrator**
- • com.telelogic.rhapsody.core.**IRPInternalOEMPlugin**
- • com.telelogic.rhapsody.core.**IRPJavaPlugins**
- • com.telelogic.rhapsody.core.**IRPModelElement**
    - ♦ com.telelogic.rhapsody.core.**IRPAction**
        - ◊ com.telelogic.rhapsody.core.**IRPSendAction**
    - ♦ com.telelogic.rhapsody.core.**IRPAssociationRole**
    - ♦ com.telelogic.rhapsody.core.**IRPClassifierRole**
    - ♦ com.telelogic.rhapsody.core.**IRPCollaboration**
        - ◊ com.telelogic.rhapsody.core.**IRPInteractionOperand**
    - ♦ com.telelogic.rhapsody.core.**IRPComponentInstance**
    - ♦ com.telelogic.rhapsody.core.**IRPConfiguration**
    - ♦ com.telelogic.rhapsody.core.**IRPDependency**
        - ◊ com.telelogic.rhapsody.core.**IRPHyperLink**
    - ♦ com.telelogic.rhapsody.core.**IRPEnumerationLiteral**
    - ♦ com.telelogic.rhapsody.core.**IRPExecutionOccurrence**
    - ♦ com.telelogic.rhapsody.core.**IRPFileFragment**

com.telelogic.rhapsody.core

♦ com.telelogic.rhapsody.core.**IRPFlow**
♦ com.telelogic.rhapsody.core.**IRPGeneralization**
♦ com.telelogic.rhapsody.core.**IRPGuard**
♦ com.telelogic.rhapsody.core.**IRPInstanceSlot**
♦ com.telelogic.rhapsody.core.**IRPInstanceSpecification**
♦ com.telelogic.rhapsody.core.**IRPInteractionOccurrence**
♦ com.telelogic.rhapsody.core.**IRPInteractionOperator**
♦ com.telelogic.rhapsody.core.**IRPMessage**
 ◊ com.telelogic.rhapsody.core.**IRPActionBlock**
 ◊ com.telelogic.rhapsody.core.**IRPConditionMark**
 ◊ com.telelogic.rhapsody.core.**IRPDestructionEvent**
♦ com.telelogic.rhapsody.core.**IRPMessagePoint**
♦ com.telelogic.rhapsody.core.**IRPStateVertex**
 ◊ com.telelogic.rhapsody.core.**IRPConnector**
  · com.telelogic.rhapsody.core.**IRPPin**
 ◊ com.telelogic.rhapsody.core.**IRPState**
  · com.telelogic.rhapsody.core.**IRPAcceptEventAction**
  · com.telelogic.rhapsody.core.**IRPAcceptTimeEvent**
  · com.telelogic.rhapsody.core.**IRPCallOperation**
  · com.telelogic.rhapsody.core.**IRPObjectNode**
♦ com.telelogic.rhapsody.core.**IRPSwimlane**
♦ com.telelogic.rhapsody.core.**IRPTemplateInstantiation**
♦ com.telelogic.rhapsody.core.**IRPTemplateInstantiationParameter**
♦ com.telelogic.rhapsody.core.**IRPTransition**
♦ com.telelogic.rhapsody.core.**IRPTrigger**
♦ com.telelogic.rhapsody.core.**IRPUnit**
 ◊ com.telelogic.rhapsody.core.**IRPAnnotation**
  · com.telelogic.rhapsody.core.**IRPComment**
  · com.telelogic.rhapsody.core.**IRPConstraint**
  · com.telelogic.rhapsody.core.**IRPRequirement**
 ◊ com.telelogic.rhapsody.core.**IRPClassifier**
  · com.telelogic.rhapsody.core.**IRPActor**
  · com.telelogic.rhapsody.core.**IRPClass**
   • com.telelogic.rhapsody.core.**IRPAssociationClass**
   • com.telelogic.rhapsody.core.**IRPStatechart**
    ♦ com.telelogic.rhapsody.core.**IRPFlowchart**
  · com.telelogic.rhapsody.core.**IRPFlowItem**
  · com.telelogic.rhapsody.core.**IRPInterfaceItem**
   • com.telelogic.rhapsody.core.**IRPEvent**
   • com.telelogic.rhapsody.core.**IRPEventReception**
   • com.telelogic.rhapsody.core.**IRPOperation**
  · com.telelogic.rhapsody.core.**IRPNode**
  · com.telelogic.rhapsody.core.**IRPStereotype**
  · com.telelogic.rhapsody.core.**IRPType**
  · com.telelogic.rhapsody.core.**IRPUseCase**
 ◊ com.telelogic.rhapsody.core.**IRPComponent**
 ◊ com.telelogic.rhapsody.core.**IRPControlledFile**
 ◊ com.telelogic.rhapsody.core.**IRPDiagram**
  · com.telelogic.rhapsody.core.**IRPCollaborationDiagram**
  · com.telelogic.rhapsody.core.**IRPComponentDiagram**
  · com.telelogic.rhapsody.core.**IRPDeploymentDiagram**

· com.telelogic.rhapsody.core.**IRPObjectModelDiagram**
· com.telelogic.rhapsody.core.**IRPPanelDiagram**
· com.telelogic.rhapsody.core.**IRPSequenceDiagram**
     • com.telelogic.rhapsody.core.**IRPTimingDiagram**
· com.telelogic.rhapsody.core.**IRPStatechartDiagram**
     • com.telelogic.rhapsody.core.**IRPActivityDiagram**
· com.telelogic.rhapsody.core.**IRPStructureDiagram**
· com.telelogic.rhapsody.core.**IRPUseCaseDiagram**
◊ com.telelogic.rhapsody.core.**IRPFile**
◊ com.telelogic.rhapsody.core.**IRPLink**
◊ com.telelogic.rhapsody.core.**IRPMatrixLayout**
◊ com.telelogic.rhapsody.core.**IRPMatrixView**
◊ com.telelogic.rhapsody.core.**IRPPackage**
· com.telelogic.rhapsody.core.**IRPProfile**
· com.telelogic.rhapsody.core.**IRPProject**
◊ com.telelogic.rhapsody.core.**IRPRelation**
· com.telelogic.rhapsody.core.**IRPInstance**
     • com.telelogic.rhapsody.core.**IRPModule**
     • com.telelogic.rhapsody.core.**IRPPort**
     • com.telelogic.rhapsody.core.**IRPSysMLPort**
◊ com.telelogic.rhapsody.core.**IRPTableLayout**
◊ com.telelogic.rhapsody.core.**IRPTableView**
◊ com.telelogic.rhapsody.core.**IRPVariable**
· com.telelogic.rhapsody.core.**IRPArgument**
· com.telelogic.rhapsody.core.**IRPAttribute**
· com.telelogic.rhapsody.core.**IRPTag**
· com.telelogic.rhapsody.core.**IRPTemplateParameter**
♦ com.telelogic.rhapsody.core.**IRPValueSpecification**
◊ com.telelogic.rhapsody.core.**IRPContextSpecification**
◊ com.telelogic.rhapsody.core.**IRPInstanceValue**
◊ com.telelogic.rhapsody.core.**IRPLiteralSpecification**
• com.telelogic.rhapsody.core.**IRPowListListener**
• com.telelogic.rhapsody.core.**IRPowPaneMgr**
• com.telelogic.rhapsody.core.**IRPowTextListener**
• com.telelogic.rhapsody.core.**IRPPlugInWindow**
• com.telelogic.rhapsody.core.**IRPProgressBar**
• com.telelogic.rhapsody.core.**IRPRhapsodyServer**
• com.telelogic.rhapsody.core.**IRPRoundTrip**
• com.telelogic.rhapsody.core.**IRPSearchManager**
• com.telelogic.rhapsody.core.**IRPSearchQuery**
• com.telelogic.rhapsody.core.**IRPSearchResult**
• com.telelogic.rhapsody.core.**IRPSelection**

Serialized Form

Package com.telelogic.rhapsody.core

Class com.telelogic.rhapsody.core.RhapsodyRuntimeException extends java.lang.RuntimeException implements Serializable

serialVersionUID: 7803061196391805387L