Rational Software Corporation®

Rational Suite®

Adapters Reference
Rational Suite Extensibility

VERSION: 2002.05.00

PART NUMBER: 800-025146-000

.

Rational®
the software development company

support@rational.com
http://www.rational.com

Part Number: 800-025146-000

Version Number: 2002.05.00

Contents

Preface

RSE delivers a comprehensive set of application programming interfaces (APIs) that provide a single platform on which to develop client and server capabilities between integrated products in Rational Suite.

This reference manual documents the set of Rational Suite Extensibility (RSE) adapters. For each RSE adapter, this reference provides information for all defined artifact types (classes), including each artifact type's, properties and relationships. Class diagrams are included to illustrate the defined relationships from an artifact type to its target artifacts.

This document was automatically generated and formatted using Rational SoDA. All adapter-specific information (diagrams, names, attributes, descriptions, and so on.) was extracted by SoDA from a Rose model.

Audience

This manual is intended for administrators, project managers, and all members of the software development team, including requirements developers, software architects and developers, and quality engineers.

Other Resources

- Other RSE documentation:
  - COM Client API Reference
  - Programmer's Guide to Application Development
  - Programmer's Guide to Adapter Development
- Rational extensibility API references:
  - ClearCase Reference Manual
  - ClearQuest API Reference
  - RequisitePro Extensibility Interface Online Help
  - RequisitePro extensibility information is documented in the RequisitePro online help for the RequisitePro Extensibility Interface. It is available from the Help menu on the ReqPro tool palette.
  - Rose Extensibility Reference
  - Team Manager Extensibility Reference
- Online Help is available for Rational Suite.
  From a Suite tool, select an option from the **Help** menu.

- All manuals are available online, either in HTML or PDF format. The online manuals are on the Rational Solutions for Windows Online Documentation CD.
- To send feedback about documentation for Rational products, please send e-mail to techpubs@rational.com.
- For more information about Rational Software technical publications, see: `http://www.rational.com/documentation`.
- For more information on training opportunities, see the Rational University Web site: `http://www.rational.com/university`.

Contacting Rational Technical Support

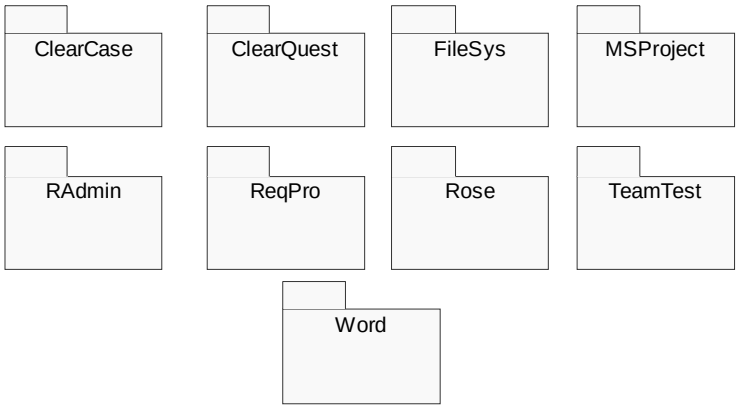
If you have questions about installing, using, or maintaining this product, contact Rational Technical Support as follows:

Your Location	Telephone	Facsimile	E-mail
North America	(800) 433-5444 (toll free) (408) 863-4000 Cupertino, CA	(781) 676-2460 Lexington, MA	support@rational.com
Europe, Middle East, Africa	+31 (0) 20-4546-200 Netherlands	+31 (0) 20-4546-201 Netherlands	support@europe.rational.com
Asia Pacific	+61-2-9419-0111 Australia	+61-2-9419-0123 Australia	support@apac.rational.com

Note: When you contact Rational Technical Support, please be prepared to supply the following information:

- Your name, company name, telephone number, and e-mail address
- Your operating system, version number, and any service packs or patches you have applied
- Product name and release number
- Your case ID number (if you are following up on a previously-reported problem)

RSE Adapters Overview



The collection of RSE adapters.

This RSE Adapter	Is for
ClearCase	Rational ClearCase
ClearQuest	Rational ClearQuest
FileSys	Microsoft File System
MSProject	Microsoft Project
RAdmin	Rational Administrator
ReqPro	Rational RequisitePro
Rose	Rational Rose
TeamTest	Rational Test Manager
Word	Microsoft Word

ClearCase

Rational ClearCase

The following Classes are available through the ClearCase RSE adapter:

- Activity
- Attribute
- AttributeType
- Baseline
- Branch
- BranchType
- CheckedOutFile
- Component
- Element
- File
- Folder
- HistoryRecord
- Hyperlink
- HyperlinkType
- Label
- LabelType
- Lock
- Name
- Project
- ProjectPolicy
- ProjectVOB
- Region
- Stream
- Trigger
- TriggerType
- UCMObject
- Value
- Version
- View
- VOB
- VOBObject

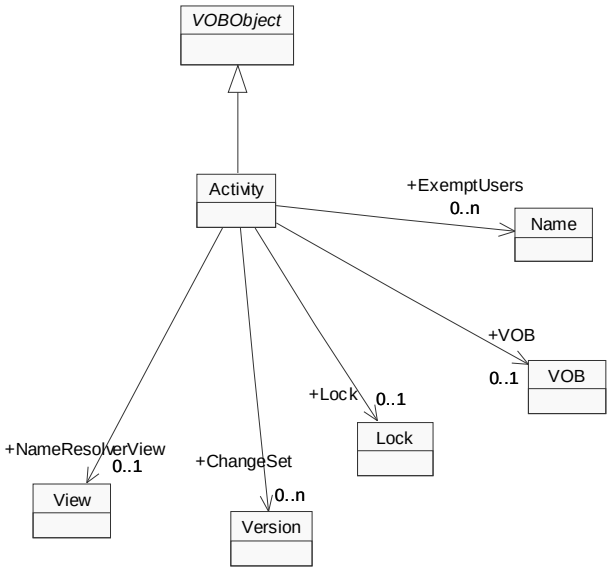
Activity (ClearCase Adapter)

In the UCM model, an activity is a ClearCase object that you use to track the work required to complete a development task. An activity includes a text headline, which describes the task, and a change set, which identifies the versions that you create or modify while working on the activity.

Class Hierarchy: VOBObject>Activity

SubClasses of Activity

Activity has no subclasses.



Properties Specific to Activity

Properties	Inherited From	Description
Comment	VOBObject	Comment associated with the VOB object.
CreatedBy	VOBObject	User who created the object.
CreatedOn	VOBObject	Date the object was created.
Headline		Title of this activity.
LockDescription		Description of the lock on this activity.

LockedBy		Name of the user who locked the activity.
LockedOn		Date the activity was locked.
Master		Master replica for this activity.
Name	VOBObject	Name of the versioned object.
OID	VOBObject	The object identifier for the VOB object.
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
State		State of the lock on this activity.
TypeName	VOBObject	The VOBObject type name.
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to Activity

Name	Kind	Class	Description
ChangeSet	0..n	Version	Versions in this activity's change set.
ExemptUsers	0..n	Name	List of users exempted from the lock.
Lock	0..1	Lock	Lock on this activity.
NameResolverView	0..1	View	A "best guess" view for resolving the names of versions in a change set.
VOB	0..1	VOB	VOB containing the activity.

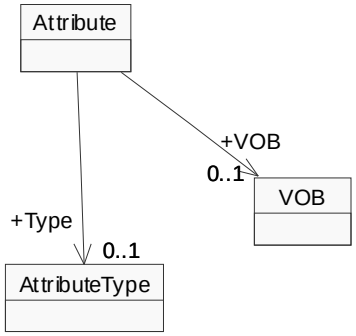
Attribute (ClearCase Adapter)

An attribute is a meta-data annotation attached to a VOB object, in the form of a name/value pair. The names of attributes are specified by user-defined attribute types; values of these attributes can be set by users. For example, a project administrator may create an attribute type whose name is QAed. A user may then attach the attribute QAed with the value "Yes" to a version. An attribute is a VOB object.

Class Hierarchy: Artifact>Attribute

SubClasses of Attribute

Attribute has no subclasses.



Properties Specific to Attribute

Properties	Inherited From	Description
Name		Attribute name.
TypeName		Attribute type name.
Value		Attribute value.

Relationships Specific to Attribute

Name	Kind	Class	Description
Type	0..1	AttributeType	Attribute type of this attribute.
VOB	0..1	VOB	VOB containing the object having this attribute.

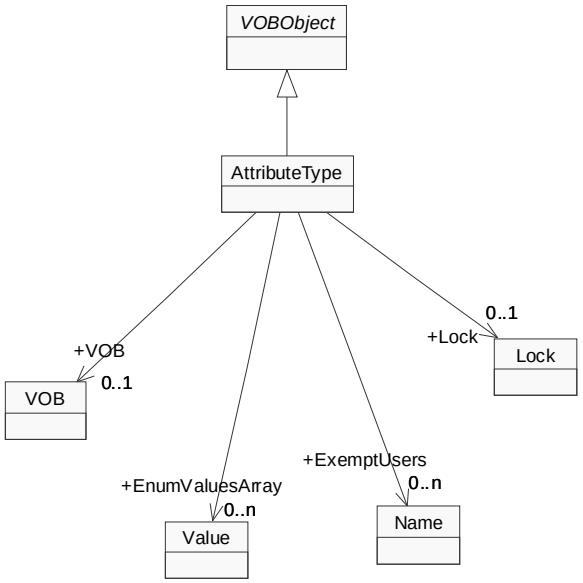
AttributeType (ClearCase Adapter)

An attribute type is a VOB object that defines an attribute name for use within a VOB. It constrains the attribute values that can be paired with the attribute name (for example, an integer in the range 1-10).

Class Hierarchy: VOBObject>AttributeType

SubClasses of AttributeType

AttributeType has no subclasses.



Properties Specific to AttributeType

Properties	Inherited From	Description
Comment	VOBObject	Comment associated with the VOB object.
Constraint		The constraint for this attribute type.
CreatedBy	VOBObject	User who created the object.
CreatedOn	VOBObject	Date the object was created.

DefaultValue		Default value for this attribute type.
Group		Group to which this attribute type belongs.
HasSharedMastership		Whether this attribute type is shared or can be mastered.
LockDescription		User comment for the lock
LockedBy		Name of the user who locked this attribute type.
LockedOn		The date this attribute type was locked.
LowerIsInRange		Whether or not the lower value is in the range of legal values for this attribute type.
LowerValue		Lower value for this attribute type.
Master		Master replica for this attribute type.
Name	VOBObject	Name of the versioned object.
NumberOfEnumValues		Number of enumerated values for this attribute type.
OID	VOBObject	The object identifier for the VOB object.
Owner		Owner of this attribute type.
Scope		Scope of this attribute type (for example, local to this VOB).
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
State		State of the lock on this attribute type.
TypeName	VOBObject	The VOBObject type name.
UpperIsInRange		Whether or not the upper value is in the range of legal values for this attribute type.
UpperValue		Upper value for this attribute type.
ValueType		Value type for this attribute.
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to AttributeType

Name	Kind	Class	Description
EnumValuesArray	0..n	Value	The enumerated values for this attribute type.
ExemptUsers	0..n	Name	List of users who are exempt from the lock.
Lock	0..1	Lock	The lock on this attribute type.
VOB	0..1	VOB	VOB containing this attribute type.

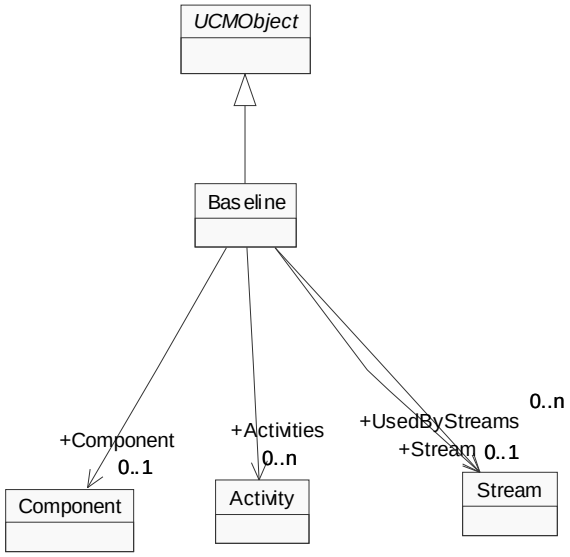
Baseline (ClearCase Adapter)

A ClearCase UCM object that typically represents a stable configuration for one or more components. A baseline identifies activities and one version of every element visible in one or more components.

Class Hierarchy: UCMObject>Baseline

SubClasses of Baseline

Baseline has no subclasses.



Properties Specific to Baseline

Properties	Inherited From	Description
Comment	VOBObject	Comment associated with the VOB object.
CreatedBy	VOBObject	User who created the object.
CreatedOn	VOBObject	Date the object was created.
Group	UCMObject	Group to which the UCM object belongs.

LabelStatus		Label status for the baseline UCM object.
LockDescription	UCMObject	Comment of the user who locked this UCMObject.
LockedBy	UCMObject	User who locked this UCMObject.
LockedOn	UCMObject	Date on which this UCMObject was locked.
Master	UCMObject	The master replica for the UCM object.
Name	VOBObject	Name of the versioned object.
OID	VOBObject	The object identifier for the VOB object.
Owner	UCMObject	Owner of the UCM object.
PromotionLevel		Promotion level for the baseline UCM object.
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
State	UCMObject	State of the lock on this UCMObject.
Title	UCMObject	Title of the UCM object.
TypeName	VOBObject	The VOBObject type name.
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to Baseline

Name	Kind	Class	Description
Activities	0..n	Activity	Activities included in the baseline UCM object.
Component	0..1	Component	Component containing the baseline UCM object.
Stream	0..1	Stream	Stream in which the baseline UCM object was created.
UsedByStreams	0..n	Stream	All of the streams for which the baseline UCM object serves as a foundation.

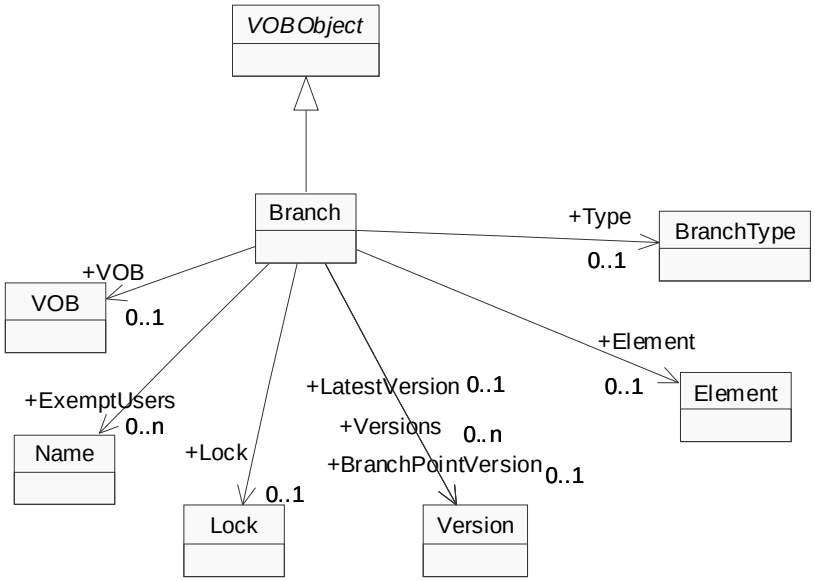
Branch (ClearCase Adapter)

A branch is an object that specifies a linear sequence of versions of an element. The entire set of versions of an element is called a version tree; it always has a single main branch, and may also have subbranches. Each branch is an instance of a branch type object. A branch is a VOBObject, and thus may have a lock-preventing modification.

Class Hierarchy: VOBObject>Branch

SubClasses of Branch

Branch has no subclasses.



Properties Specific to Branch

Properties	Inherited From	Description
BranchPath		Path of this branch.
Comment	VOBObject	Comment associated with the VOB object.
CreatedBy	VOBObject	User who created the object.

CreatedOn	VOBObject	Date the object was created.
ExtendedPath		Extended path of the branch.
LockDescription		Description of the current lock on the branch.
LockedBy		Name of the user who locked the branch.
LockedOn		Date the branch was locked.
Master		Master replica for this branch.
Name	VOBObject	Name of the versioned object.
OID	VOBObject	The object identifier for the VOB object.
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
State		State of the lock on this branch.
TypeName	VOBObject	The VOBObject type name.
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to Branch

Name	Kind	Class	Description
BranchPointVersion	0..1	Version	Version from which this branch sprouts.
Element	0..1	Element	The element to which this branch belongs.
ExemptUsers	0..n	Name	List of users exempt from the lock.
LatestVersion	0..1	Version	Latest version of this branch
Lock	0..1	Lock	Lock on this branch.
Type	0..1	BranchType	Branch type of this branch.
Versions	0..n	Version	An enumeration of all versions along this branch.
VOB	0..1	VOB	VOB containing this branch.

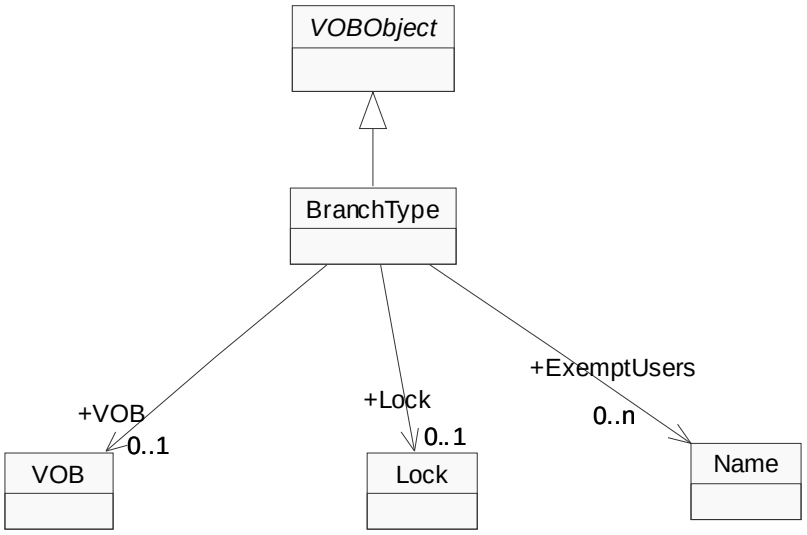
BranchType (ClearCase Adapter)

A branch type defines a branch name for use within a VOB.

Class Hierarchy: VOBObject>BranchType

SubClasses of BranchType

BranchType has no subclasses.



Properties Specific to BranchType

Properties	Inherited From	Description
Comment	VOBObject	Comment associated with the VOB object.
Constraint		The constraint for this branch type.
CreatedBy	VOBObject	User who created the object.
CreatedOn	VOBObject	Date the object was created.
Group		The group to which this branch type belongs.
LockDescription		User comment for the lock on this branch type.

LockedBy		Name of the user who locked this branch type.
LockedOn		Date on which this branch type was locked.
Master		The master replica for this branch type.
Name	VOBObject	Name of the versioned object.
OID	VOBObject	The object identifier for the VOB object.
Owner		Owner of this branch type.
Scope		The scope of this branch type (for example, local to this VOB)
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
State		State of the lock on this branch type.
TypeName	VOBObject	The VOBObject type name.
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to BranchType

Name	Kind	Class	Description
ExemptUsers	0..n	Name	List of users who are exempt from the lock on this branch type.
Lock	0..1	Lock	Lock on this branch type.
VOB	0..1	VOB	VOB containing this branch type.

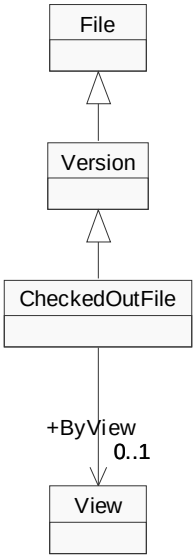
CheckedOutFile (ClearCase Adapter)

A checked out file is a placeholder in the VOB database created by the checkout command. This object corresponds to the view-private object (file or directory) that you work with after checking out an element. A checkout will be marked reserved if reserved checkout has been performed (meaning the file is exclusively locked for one user).

Class Hierarchy: VOBObject>File>Version>CheckedOutFile

SubClasses of CheckedOutFile

CheckedOutFile has no subclasses.



Properties Specific to CheckedOutFile

Properties	Inherited From	Description
Comment	VOBObject	Comment associated with the VOB object.
CreatedBy	VOBObject	User who created the object.
CreatedOn	VOBObject	Date the object was created.
ExtendedPath	File	VOB-extended path of this file system object.

Extension	File	File extension (the portion after the final dot).
Identifier	Version	The version's identifier string.
IsCheckedOut	Version	Whether or not this object represents a checked-out file.
IsDifferent	Version	Whether or not this version is different from its predecessor.
IsDirectory	File	Whether or not the file is a directory.
IsHijacked	Version	Whether or not this version is hijacked.
IsLatest	Version	Whether or not this version is the latest on its branch.
IsReserved		Whether or not this checkout is reserved.
Name	VOBObject	Name of the versioned object.
NameMinusExtension	File	Simple name of the file without the extension and final.dot.
OID	VOBObject	The object identifier for the VOB object.
Path	File	Path to this file system object.
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
SimpleName	File	Simple name of the file, that is, the name of the file without the path.
TypeName	VOBObject	The VOBObject type name.
VersionNumber	Version	This version's version number.
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to CheckedOutFile

Name	Kind	Class	Description
ByView	0..1	View	The view to which this file is checked out.

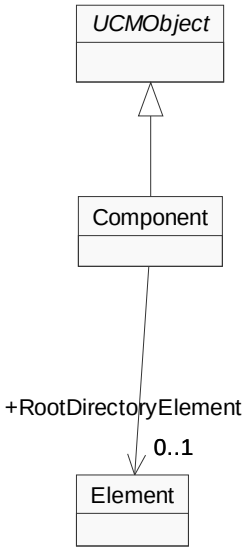
Component (ClearCase Adapter)

A ClearCase object that you use to group a set of related directory and file elements within a UCM project. Typically, you develop, integrate, and release the elements that make up a component together. A project must contain at least one component, and it can contain multiple components. Projects can share components.

Class Hierarchy: UCMObject>Component

SubClasses of Component

Component has no subclasses.



Properties Specific to Component

Properties	Inherited From	Description
Comment	VOBObject	Comment associated with the VOB object.
CreatedBy	VOBObject	User who created the object.
CreatedOn	VOBObject	Date the object was created.

Group	UCMObject	Group to which the UCM object belongs.
LockDescription	UCMObject	Comment of the user who locked this UCMObject.
LockedBy	UCMObject	User who locked this UCMObject.
LockedOn	UCMObject	Date on which this UCMObject was locked.
Master	UCMObject	The master replica for the UCM object.
Name	VOBObject	Name of the versioned object.
OID	VOBObject	The object identifier for the VOB object.
Owner	UCMObject	Owner of the UCM object.
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
State	UCMObject	State of the lock on this UCMObject.
Title	UCMObject	Title of the UCM object.
TypeName	VOBObject	The VOBObject type name.
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to Component

Name	Kind	Class	Description
RootDirectoryElement	0..1	Element	The root directory for the component.

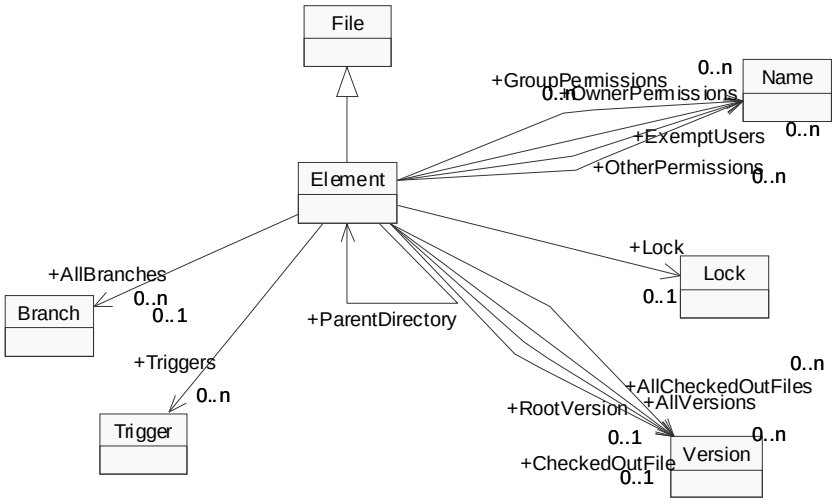
Element (ClearCase Adapter)

An element is an object that encompasses a set of versions, organized into a version tree. An element may have a lock if a version of the element is checked out in a view.

Class Hierarchy: VOBObject>File>Element

SubClasses of Element

Element has no subclasses.



Properties Specific to Element

Properties	Inherited From	Description
Comment	VOBObject	Comment associated with the VOB object.
CreatedBy	VOBObject	User who created the object.
CreatedOn	VOBObject	Date the object was created.
ElementType		Element type of this element.
ExtendedPath	File	VOB-extended path of this file system object.
Extension	File	File extension (the portion after the final dot).

Group		Group to which this element belongs.
IsDirectory	File	Whether or not the file is a directory.
LockDescription		Comment associated with the history record for the lock.
LockedBy		User who locked this element.
LockedOn		Date the element was locked.
Master		Master replica for this element.
Name	VOBObject	Name of the versioned object.
NameMinusExtension	File	Simple name of the file without the extension and final.dot.
OID	VOBObject	The object identifier for the VOB object.
Owner		Owner of the element.
Path	File	Path to this file system object.
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
SimpleName	File	Simple name of the file, that is, the name of the file without the path.
State		Current state of the lock on this element.
TypeName	VOBObject	The VOBObject type name.
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to Element

Name	Kind	Class	Description
AllBranches	0..n	Branch	All branches in the version tree for this element.
AllCheckedOutFiles	0..n	Version	The versions of the element checked out to any view.
AllVersions	0..n	Version	Versions in the version tree for this element.
CheckedOutFile	0..1	Version	Version of the element checked out to the associated view.

ExemptUsers	0..n	Name	Array of string values containing the names of users exempted from the lock being created.
GroupPermissions	0..n	Name	The group permissions of the element (users within the same group have these permissions).
Lock	0..1	Lock	The lock on this element.
OtherPermissions	0..n	Name	Other permissions of the element (all users).
OwnerPermissions	0..n	Name	The owner permissions of the element (the owner has these permissions).
ParentDirectory		Element	This element's parent directory element.
RootVersion	0..1	Version	The particular version of this element specified by the version selector.
Triggers	0..n	Trigger	The collection of triggers attached to this file of directory element.

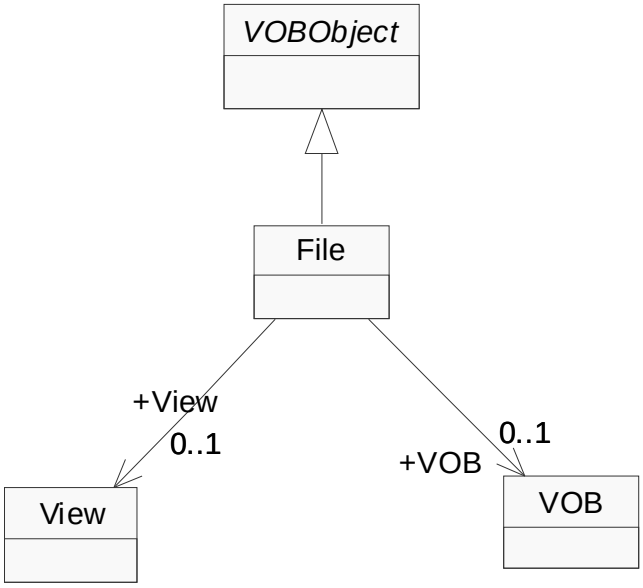
File (ClearCase Adapter)

The File class represents all VOB objects, which are physical files such as elements and versions. A File object does not include view-private objects.

Class Hierarchy: VOBObject>File

SubClasses of File

Element
Version



Properties Specific to File

Properties	Inherited From	Description
Comment	VOBObject	Comment associated with the VOB object.
CreatedBy	VOBObject	User who created the object.
CreatedOn	VOBObject	Date the object was created.
ExtendedPath		VOB-extended path of this file system object.

Extension		File extension (the portion after the final dot).
IsDirectory		Whether or not the file is a directory.
Name	VOBObject	Name of the versioned object.
NameMinusExtension		Simple name of the file without the extension and final.dot.
OID	VOBObject	The object identifier for the VOB object.
Path		Path to this file system object.
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
SimpleName		Simple name of the file, that is, the name of the file without the path.
TypeName	VOBObject	The VOBObject type name.
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to File

Name	Kind	Class	Description
View	0..1	View	The view associated with this file.
VOB	0..1	VOB	The VOB associated with this file.

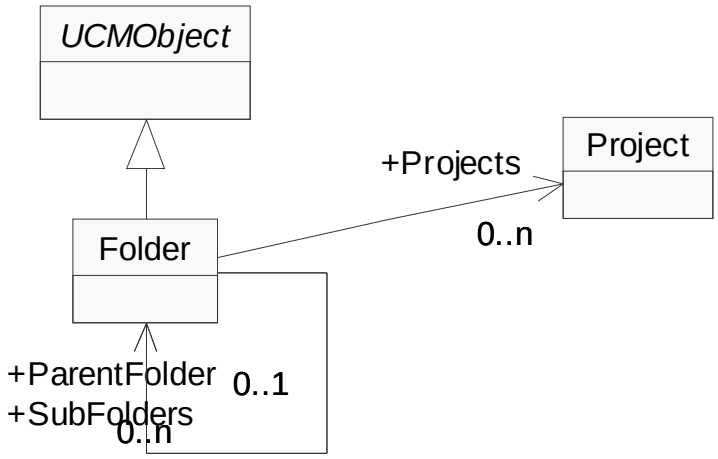
Folder (ClearCase Adapter)

Folder is a ClearCase UCM object that contains one or more projects.

Class Hierarchy: UCMObject>Folder

SubClasses of Folder

Folder has no subclasses.



Properties Specific to Folder

Properties	Inherited From	Description
Comment	VOBObject	Comment associated with the VOB object.
CreatedBy	VOBObject	User who created the object.
CreatedOn	VOBObject	Date the object was created.
Group	UCMObject	Group to which the UCM object belongs.
IsRootFolder		True if the folder is the root of the project hierarchy in its project VOB.
LockDescription	UCMObject	Comment of the user who locked this UCMObject.
LockedBy	UCMObject	User who locked this UCMObject.

LockedOn	UCMObject	Date on which this UCMObject was locked.
Master	UCMObject	The master replica for the UCM object.
Name	VOBObject	Name of the versioned object.
OID	VOBObject	The object identifier for the VOB object.
Owner	UCMObject	Owner of the UCM object.
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
State	UCMObject	State of the lock on this UCMObject.
Title	UCMObject	Title of the UCM object.
TypeName	VOBObject	The VOBObject type name.
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to Folder

Name	Kind	Class	Description
ParentFolder	0..1	Folder	Name of the parent folder.
Projects	0..n	Project	Projects contained in the folder.
SubFolders	0..n	Folder	Folders contained within the folder.

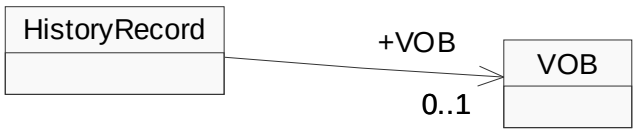
HistoryRecord (ClearCase Adapter)

A history record is meta-data in a VOB, representing an event record involving a VOB object. The history of a file element includes history records for creation of the element, creation of each version of the file, creation of each branch, assignment of attribute to the element and/or its versions, attaching of hyperlinks to the element and/or its versions, and so on.

Class Hierarchy: Artifact>HistoryRecord

SubClasses of HistoryRecord

HistoryRecord has no subclasses.



Properties Specific to HistoryRecord

Properties	Inherited From	Description
Comment		Comment associated with the operation indicated by this history record.
Date		Date and time the operation was executed.
EventKind		Indicates the type of operation that was executed.
Group		Name of the login group that performed the operation indicated by this history record.
Host		Name of the host system from which the operation indicated by this history record was executed.

Selector	An expression used by ClearCase's file typing mechanism to match a file system object (or just the name of one).
UserFullName	Full name of the user who performed the operation indicated by this history record.
UserLoginName	The login name of the user who performed the operation indicated by this history record.

Relationships Specific to HistoryRecord

Name	Kind	Class	Description
VOB	0..1	VOB	The VOB containing the object to which the operation was applied.

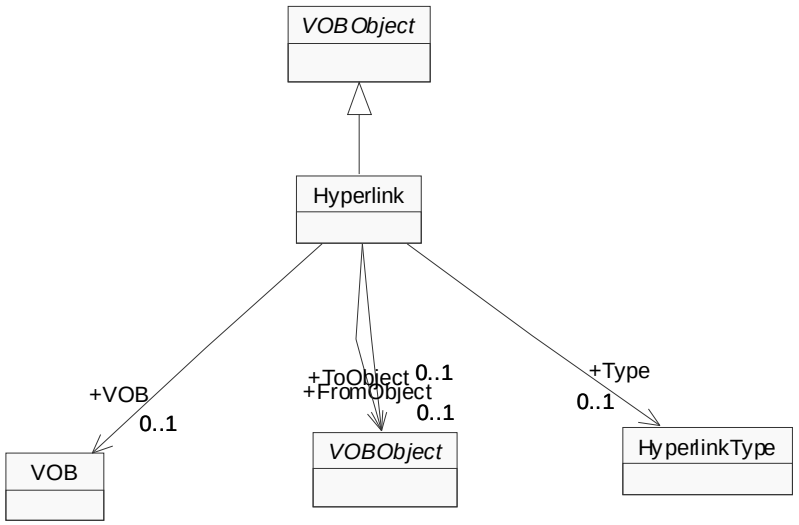
Hyperlink (ClearCase Adapter)

A hyperlink is a logical pointer between two objects. A hyperlink is a VOB object, it derives its name by referencing another VOB object, a hyperlink type. A hyperlink can have FromText and ToText, which are technically string-valued attributes on the hyperlink object. A hyperlink has a From-object and To-object, which are VOB objects. A hyperlink may be bidirectional, indicating that it can be traversed both from To-object to From-object and From-object to To-object. The IsUnidirectional selector will be False if a hyperlink is bidirectional.

Class Hierarchy: VOBObject>Hyperlink

SubClasses of Hyperlink

Hyperlink has no subclasses.



Properties Specific to Hyperlink

Properties	Inherited From	Description
Comment	VOBObject	Comment associated with the VOB object.
CreatedBy	VOBObject	User who created the object.
CreatedOn	VOBObject	Date the object was created.

FromText		The from-text on the from-object of the hyperlink.
Group		The group to which this hyperlink belongs.
IDString		String identifying the hyperlink (type-name@id@vob-selector).
Master		Master replica for this hyperlink.
Name	VOBObject	Name of the versioned object.
OID	VOBObject	The object identifier for the VOB object.
Owner		Owner of this hyperlink.
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
ToText		The to-text on the to-object of the hyperlink.
TypeName	VOBObject	The VOBObject type name.
Unidirectional		Whether or not the hyperlink object can be navigated only in one direction (From-object > To-object).
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to Hyperlink

Name	Kind	Class	Description
FromObject	0..1	VOBObject	The From-object of the hyperlink.
ToObject	0..1	VOBObject	The To-object of the hyperlink.
Type	0..1	HyperlinkType	The hyperlink type of this hyperlink.
VOB	0..1	VOB	The VOB containing this hyperlink.

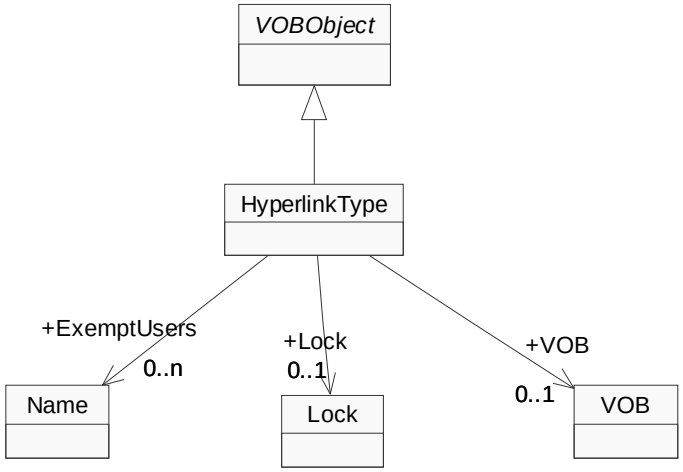
HyperlinkType (ClearCase Adapter)

A HyperlinkType is an object that defines a hyperlink name for use within a VOB. A HyperlinkType may be shared or local.

Class Hierarchy: VOBObject>HyperlinkType

SubClasses of HyperlinkType

HyperlinkType has no subclasses.



Properties Specific to HyperlinkType

Properties	Inherited From	Description
Comment	VOBObject	Comment associated with the VOB object.
CreatedBy	VOBObject	User who created the object.
CreatedOn	VOBObject	Date the object was created.
Group		Group to which this hyperlink type belongs.
HasSharedMastership		Whether this hyperlink type is shared or can be mastered.
LockDescription		The comment of the user who locked this hyperlink type.

LockedBy		Name of the user who locked this hyperlink type.
LockedOn		Date on which this hyperlink type was locked.
Master		The master replica for this hyperlink type.
Name	VOBObject	Name of the versioned object.
OID	VOBObject	The object identifier for the VOB object.
Owner		Owner of this hyperlink type.
Scope		Scope of this hyperlink type (for example, local to this VOB).
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
State		State of the lock on this hyperlink type.
TypeName	VOBObject	The VOBObject type name.
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to HyperlinkType

Name	Kind	Class	Description
ExemptUsers	0..n	Name	The list of users who are exempt from the lock.
Lock	0..1	Lock	Lock on this hyperlink type
VOB	0..1	VOB	The VOB containing this hyperlink type.

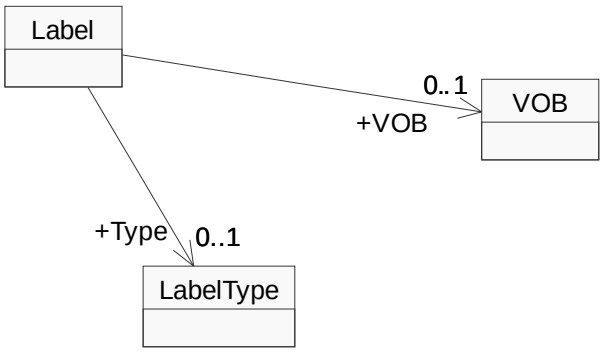
Label (ClearCase Adapter)

A label is an instance of a LabelType object, supplying a user-defined name for a version. One or more labels may be assigned to a given version.

Class Hierarchy: Artifact>Label

SubClasses of Label

Label has no subclasses.



Properties Specific to Label

Properties	Inherited From	Description
TypeName		Label type name.

Relationships Specific to Label

Name	Kind	Class	Description
Type	0..1	LabelType	The label type of this label.
VOB	0..1	VOB	The VOB containing the labeled version.

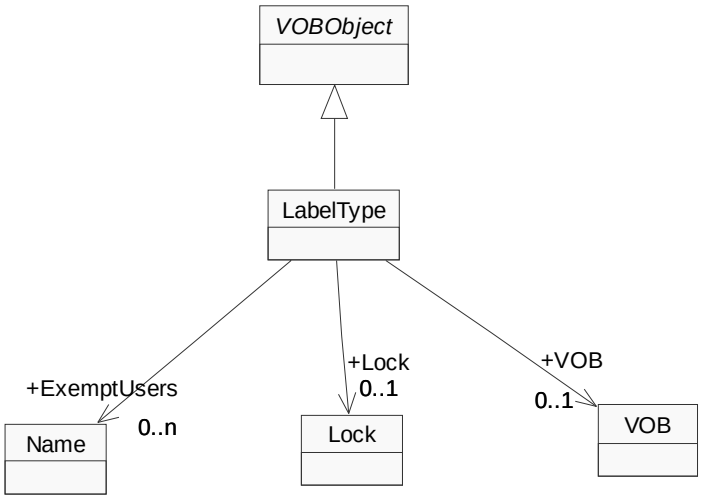
LabelType (ClearCase Adapter)

A label type is a type object that defines a version label for use within a VOB.

Class Hierarchy: VOBObject>LabelType

SubClasses of LabelType

LabelType has no subclasses.



Properties Specific to LabelType

Properties	Inherited From	Description
Comment	VOBObject	Comment associated with the VOB object.
Constraint		Constraint for this label type.
CreatedBy	VOBObject	User who created the object.
CreatedOn	VOBObject	Date the object was created.
Group		Group to which this label type belongs.
HasSharedMastership		Whether this label type is shared or can be mastered.
LockDescription		Comment of the user who locked this label type.

LockedBy		Name of the user who locked this label type.
LockedOn		Date on which this label type was locked.
Master		The master replica for this label type.
Name	VOBObject	Name of the versioned object.
OID	VOBObject	The object identifier for the VOB object.
Owner		Owner of this label type.
Scope		Whether this label type is global for VOBs using this as an admin VOB or local to this VOB.
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
State		State of the lock on this label type.
TypeName	VOBObject	The VOBObject type name.
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to LabelType

Name	Kind	Class	Description
ExemptUsers	0..n	Name	The list of users who are exempt from the lock on this label type.
Lock	0..1	Lock	The lock on this label type.
VOB	0..1	VOB	The VOB containing this label type.

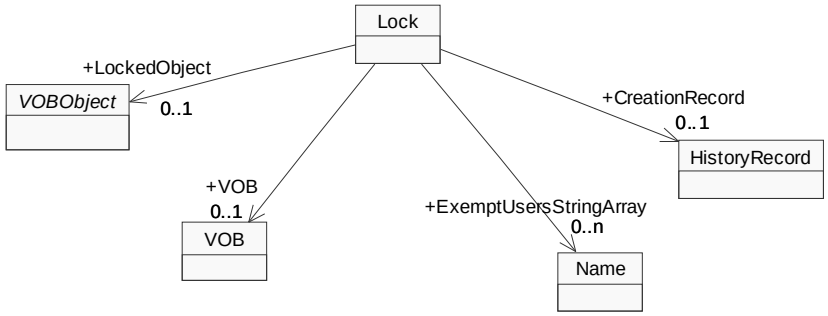
Lock (ClearCase Adapter)

A lock is a mechanism that prevents a VOB object from being modified (for file system objects) or from being instanced (for type objects).

Class Hierarchy: Artifact>Lock

SubClasses of Lock

Lock has no subclasses.



Properties Specific to Lock

Properties	Inherited From	Description
Comment		User's comment for the lock.
CreatedBy		Name of the user who created the lock.
CreatedOn		Date the lock was created.
Index		An index to the lock.
IsObsolete		Whether the locked object is marked as obsolete.
NumberOfExemptUsers		The number of users who are exempt from this lock.
Selector		An expression used by ClearCase's file typing mechanism to match a file system object (or just the name of one).

Relationships Specific to Lock

Name	Kind	Class	Description
CreationRecord	0..1	HistoryRecord	Creation record for this lock.
ExemptUsersStringArray	0..n	Name	Users who are exempt from this lock.
LockedObject	0..1	VOBObject	The object held by this lock.
VOB	0..1	VOB	The VOB in which this lock resides.

Name (ClearCase Adapter)

The Name class represents a string corresponding to the name or path of a ClearCase object.

Class Hierarchy: Artifact>Name

SubClasses of Name

Name has no subclasses.

Properties Specific to Name

Properties	Inherited From	Description
Name		Simple name string.

Relationships Specific to Name

Name	Kind	Class	Description
This class has no relationships.			

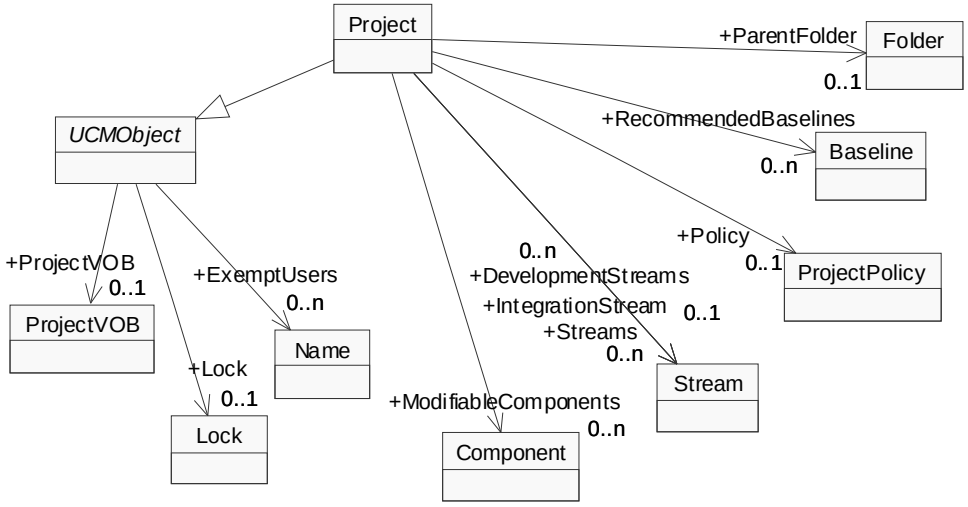
Project (ClearCase Adapter)

A Project defines a set of development policies and a set of configurations used in a development effort.

Class Hierarchy: UCMObject>Project

SubClasses of Project

Project has no subclasses.



Properties Specific to Project

Properties	Inherited From	Description
ClearQuestDatabaseName		Name of the ClearQuest database linked to the CRM-enabled project.
Comment	VOBObject	Comment associated with the VOB object.
CreatedBy	VOBObject	User who created the object.
CreatedOn	VOBObject	Date the object was created.
Group	UCMObject	Group to which the UCM object belongs.

HasStreams		True if there are any streams associated with the project.
IsCRMEnabled		True if the project is CRM enabled (that is, it is linked to a ClearQuest database).
LockDescription	UCMObject	Comment of the user who locked this UCMObject.
LockedBy	UCMObject	User who locked this UCMObject.
LockedOn	UCMObject	Date on which this UCMObject was locked.
Master	UCMObject	The master replica for the UCM object.
Name	VOBObject	Name of the versioned object.
OID	VOBObject	The object identifier for the VOB object.
Owner	UCMObject	Owner of the UCM object.
RequiredPromotionLevel		The minimum promotion level a baseline must have to be a recommended baseline in a rebase operation.
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
State	UCMObject	State of the lock on this UCMObject.
Title	UCMObject	Title of the UCM object.
TypeName	VOBObject	The VOBObject type name.
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to Project

Name	Kind	Class	Description
DevelopmentStreams	0..n	Stream	The development streams of the project.
IntegrationStream	0..1	Stream	Integration stream for the project.
ModifiableComponents	0..n	Component	The set of components that can be modified by the project.
ParentFolder	0..1	Folder	Folder containing the project.

Policy	0..1	ProjectPolicy	Policy settings associated with the project.
RecommendedBaselines	0..n	Baseline	The project's list of recommended baselines.
Streams	0..n	Stream	Streams for the project.

ProjectPolicy (ClearCase Adapter)

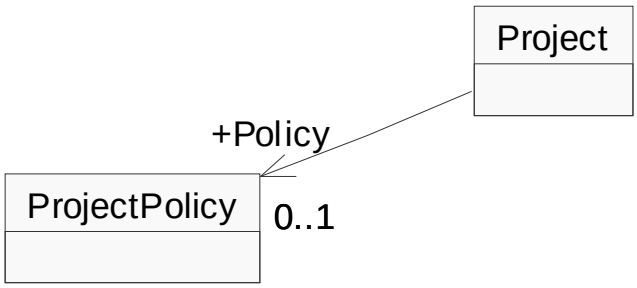
A project's policies specifies how developers access and modify sets of source files and directories (called components). To record and configure the development work that proceeds on components, projects use the following objects:

Baseline
Stream
Activity

Class Hierarchy: Artifact>ProjectPolicy

SubClasses of ProjectPolicy

ProjectPolicy has no subclasses.



Properties Specific to ProjectPolicy

Properties	Inherited From	Description
DeliverRequireCheckin		True if delivery is denied from a development stream that has checkouts.
DeliverRequireRebase		True if development stream must be based on the current recommended baselines before it can be used to deliver changes to the integration stream.
UNIXDevelopmentSnapshot		Recommended snapshot views for development work on UNIX platforms.
UNIXIntegrationSnapshot		Recommended snapshot views for integration work on UNIX platforms.

WinDevelopmentSnapshot	Recommended snapshot views for development work on Window platforms.
WinIntegrationSnapshot	Recommended snapshot views for integration work on Window platforms.

Relationships Specific to ProjectPolicy

Name	Kind	Class	Description

This class has no relationships.

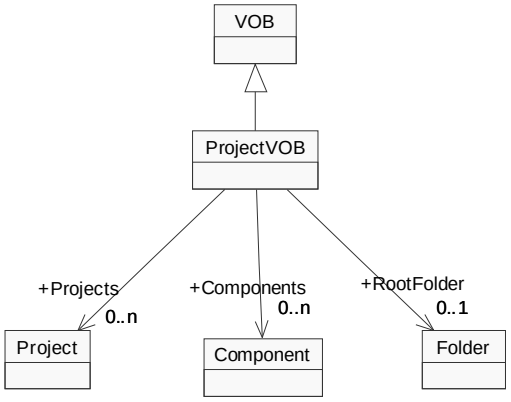
ProjectVOB (ClearCase Adapter)

A special type of VOB used in the Unified Configuration Management (UCM) facilities of ClearCase. Contains some additional properties that a VOB does not.

Class Hierarchy: VOB>ProjectVOB

SubClasses of ProjectVOB

ProjectVOB has no subclasses.



Properties Specific to ProjectVOB

Properties	Inherited From	Description
Comment	VOBObject	Comment associated with the VOB object.
CreatedBy	VOBObject	User who created the object.
CreatedOn	VOBObject	Date the object was created.
DefaultPromotionLevel		Default promotion level in the project VOB.
Group	VOB	Group to which this VOB belongs.
HasMSDOSTextMode	VOB	Whether or not this VOB has MS-DOS text mode enabled.

Host	VOB	Host on which the storage area for this VOB resides.
IsMounted	VOB	Whether or not the VOB is mounted.
IsReplicated	VOB	Whether or not this VOB is replicated.
LockDescription	VOB	Description of the lock for the VOB.
LockedBy	VOB	Name of the user who locked the VOB.
LockedOn	VOB	Date the VOB was locked
Name	VOBObject	Name of the versioned object.
NumberOfAdditionalGroups	VOB	Number of additional groups to which this VOB belongs.
NumberOfPromotionLevels		Number of promotion levels in the project VOB.
NumberOfReplicas	VOB	The number of replica names for the VOB family of this VOB, if this VOB is replicated.
OID	VOBObject	The object identifier for the VOB object.
Owner	VOB	Owner of the VOB.
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
State	VOB	State of the lock on the VOB.
TagName	VOB	The VOB-tag name.
ThisReplica	VOB	Replica name for this VOB, if the VOB is replicated.
TypeName	VOBObject	The VOBObject type name.
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to ProjectVOB

Name	Kind	Class	Description
Components	0..n	Component	Components in the project VOB.
Projects	0..n	Project	Projects in the project VOB.
RootFolder	0..1	Folder	The root folder in the project VOB.

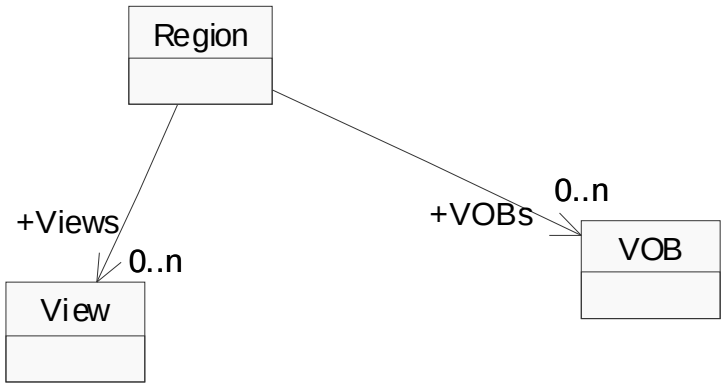
Region (ClearCase Adapter)

Region is a ClearCase file. A network region is a logical subset of a local area network, within which all hosts refer to VOB storage directories and view storage directories with the same network path. The ClearCase domain supports retrieval of VOBs and Views within a region.

Class Hierarchy: Artifact>Region

SubClasses of Region

Region has no subclasses.



Properties Specific to Region

Properties	Inherited From	Description
Region		Name of the region.

Relationships Specific to Region

Name	Kind	Class	Description
Views	0..n	View	Views contained within the region.
VOBs	0..n	VOB	VOBs contained within the region.

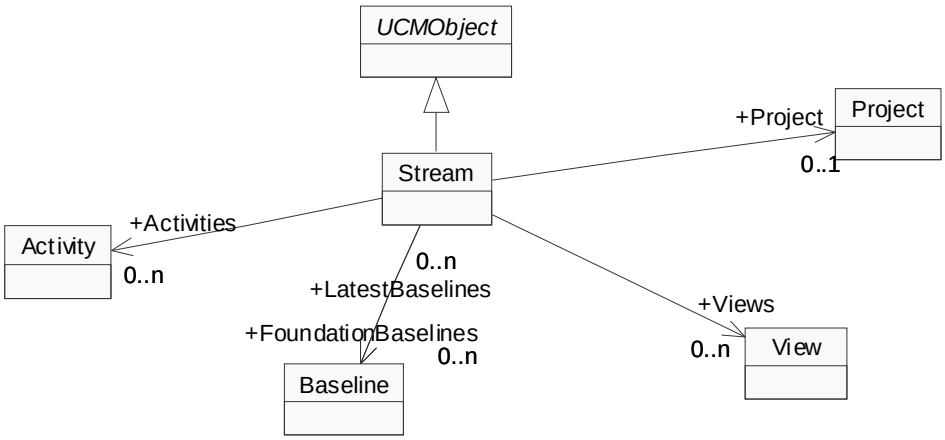
Stream (ClearCase Adapter)

Stream is a mechanism for creating and recording configurations. A stream identifies the exact set of versions currently available for you to view, modify, or build.

Class Hierarchy: UCMObject>Stream

SubClasses of Stream

Stream has no subclasses.



Properties Specific to Stream

Properties	Inherited From	Description
Comment	VOBObject	Comment associated with the VOB object.
CreatedBy	VOBObject	User who created the object.
CreatedOn	VOBObject	Date the object was created.
Group	UCMObject	Group to which the UCM object belongs.
HasActivities		True if there are any activities associated with the stream.
IsIntegrationStream		True if the stream is an integration stream in the project.

LockDescription	UCMObject	Comment of the user who locked this UCMObject.
LockedBy	UCMObject	User who locked this UCMObject.
LockedOn	UCMObject	Date on which this UCMObject was locked.
Master	UCMObject	The master replica for the UCM object.
Name	VOBObject	Name of the versioned object.
OID	VOBObject	The object identifier for the VOB object.
Owner	UCMObject	Owner of the UCM object.
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
State	UCMObject	State of the lock on this UCMObject.
Title	UCMObject	Title of the UCM object.
TypeName	VOBObject	The VOBObject type name.
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to Stream

Name	Kind	Class	Description
Activities	0..n	Activity	Activities associated with the Stream.
FoundationBaselines	0..n	Baseline	The foundation baselines for the stream for all components.
LatestBaselines	0..n	Baseline	The latest baselines in the stream for all components.
Project	0..1	Project	Project for the stream.
Views	0..n	View	The set of views associated with the stream.

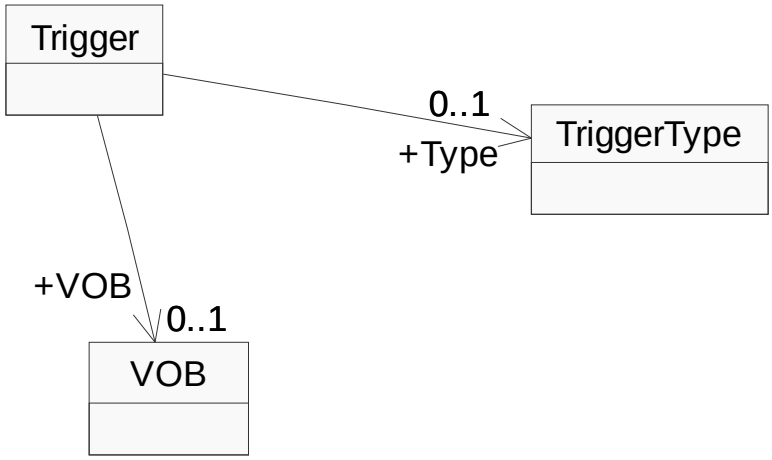
Trigger (ClearCase Adapter)

A trigger is a monitor that specifies one or more standard programs or built-in actions to be executed automatically whenever a certain ClearCase operation is performed. A trigger is associated with a TriggerType object, which groups triggers of similar properties.

Class Hierarchy: Artifact>Trigger

SubClasses of Trigger

Trigger has no subclasses.



Properties Specific to Trigger

Properties	Inherited From	Description
IsOnAttachedList		Whether this trigger is on the attached list of the element.
IsOnInheritenceList		Whether this trigger is on the inheritance list of an element, if the element is a directory element.
TypeName		The trigger type name.

Relationships Specific to Trigger

Name	Kind	Class	Description
Type	0..1	TriggerType	Trigger type of this element trigger.
VOB	0..1	VOB	The VOB containing this element trigger.

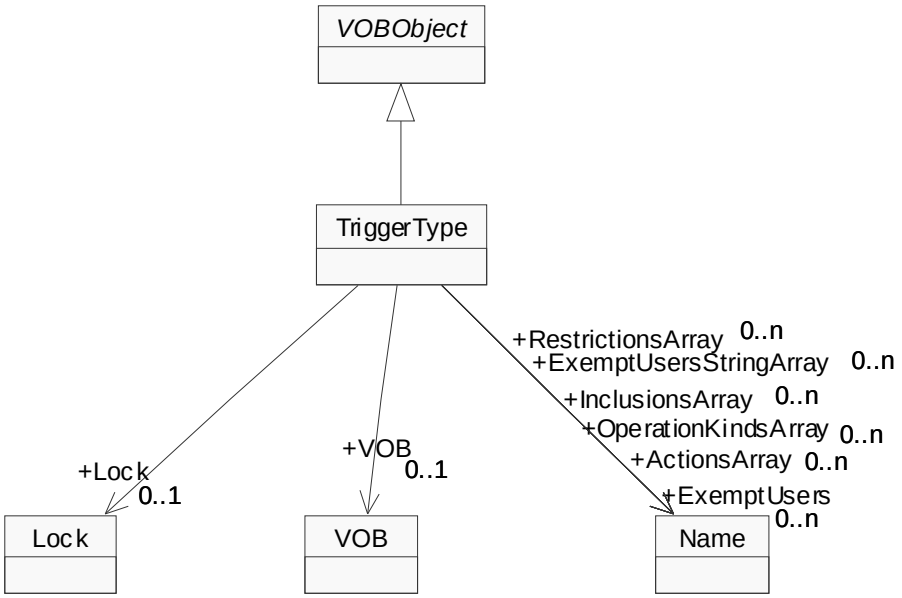
TriggerType (ClearCase Adapter)

A trigger type is an object through which triggers are defined. The trigger kind for a trigger type includes element, all-element, and type. Instances of an element trigger type can be attached to one or more individual elements. An all-element trigger type is implicitly attached to all elements in a VOB. A type trigger type is attached to a specified collection of type object.

Class Hierarchy: VOBObject>TriggerType

SubClasses of TriggerType

TriggerType has no subclasses.



Properties Specific to TriggerType

Properties	Inherited From	Description
Comment	VOBObject	Comment associated with the VOB object.
CreatedBy	VOBObject	User who created the object.
CreatedOn	VOBObject	Date the object was created.

DebugPrinting		Whether or not debug printing happens when the trigger fires.
Firing		The trigger type firing order, before or after the operation (pre-op or post-op).
Group		Group to which this trigger type belongs.
KindOfTrigger		The kind of trigger for this trigger type.
LockDescription		Comment of the user who locked this trigger type.
LockedBy		User who locked this trigger type.
LockedOn		Date on which this trigger type was locked.
Name	VOBObject	Name of the versioned object.
NumberOfActions		Number of actions for this trigger type.
NumberOfExemptUsers		Number of users for whom this trigger type does not fire.
NumberOfInclusions		Number of inclusions for this element trigger type.
NumberOfOperationKinds		Number of operation kinds which fire this trigger type.
NumberOfRestrictions		Number of restrictions for this trigger type
OID	VOBObject	The object identifier for the VOB object.
Owner		Owner of this trigger type.
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
State		State of the lock on this trigger type.
TypeName	VOBObject	The VOBObject type name.
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to TriggerType

Name	Kind	Class	Description
ActionsArray	0..n	Name	An array of action/value pairs for this trigger type (that is, a type followed by one or two values).
ExemptUsers	0..n	Name	The users exempted from the firing of triggers for this trigger type.
ExemptUsersStringArray	0..n	Name	Array of users who are exempt from this trigger type.
InclusionsArray	0..n	Name	The inclusion list for this trigger type.
Lock	0..1	Lock	The lock on this trigger type.
OperationKindsArray	0..n	Name	Array of kinds of operations which fire this trigger type.
RestrictionsArray	0..n	Name	The restriction list for this element trigger type.
VOB	0..1	VOB	The VOB containing this trigger type.

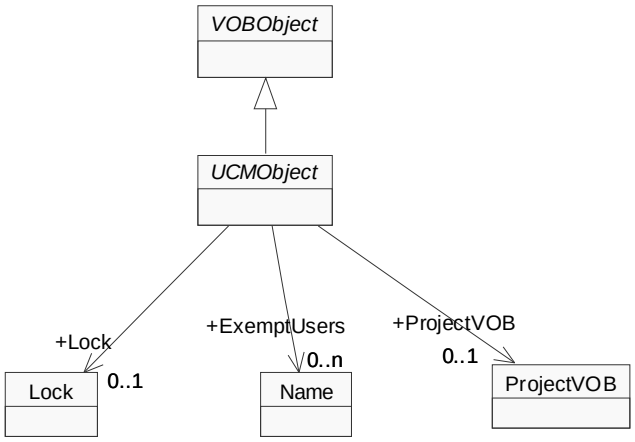
UCMObject (ClearCase Adapter)

The UCMObject class is the class from which all UCM objects are based. For historical reasons, the Activity class is based on VOBObject instead.

Class Hierarchy: VOBObject>UCMObject

SubClasses of UCMObject

Baseline
Component
Folder
Project
Stream



Properties Specific to UCMObject

Properties	Inherited From	Description
Comment	VOBObject	Comment associated with the VOB object.
CreatedBy	VOBObject	User who created the object.
CreatedOn	VOBObject	Date the object was created.
Group		Group to which the UCM object belongs.

LockDescription		Comment of the user who locked this UCMObject.
LockedBy		User who locked this UCMObject.
LockedOn		Date on which this UCMObject was locked.
Master		The master replica for the UCM object.
Name	VOBObject	Name of the versioned object.
OID	VOBObject	The object identifier for the VOB object.
Owner		Owner of the UCM object.
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
State		State of the lock on this UCMObject.
Title		Title of the UCM object.
TypeName	VOBObject	The VOBObject type name.
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to UCMObject

Name	Kind	Class	Description
ExemptUsers	0..n	Name	The users exempted from this UCMObject.
Lock	0..1	Lock	The lock for the UCM object.
ProjectVOB	0..1	ProjectVOB	The project VOB for the UCM object.

Value (ClearCase Adapter)

The value class represents a string value occurring within a collection of values.

Class Hierarchy: Artifact>Value

SubClasses of Value

Value has no subclasses.

Properties Specific to Value

Properties	Inherited From	Description
Value		Simple value string.

Relationships Specific to Value

Name	Kind	Class	Description
This class has no relationships.			

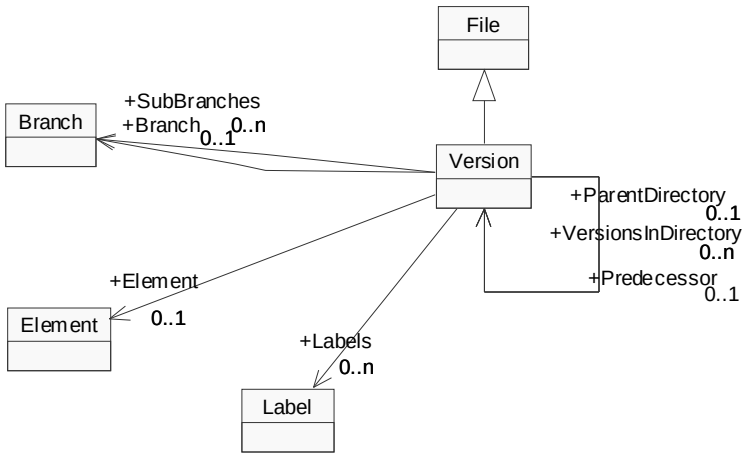
Version (ClearCase Adapter)

A version is an object that implements a particular revision of an element. The versions of an element are organized into a version tree structure. Also, a checked-out version can refer to the view-private file that corresponds to the object created in a VOB database by the checkout command. If a version is a directory, it may contain subversions corresponding to those versions within the directory.

Class Hierarchy: VOBObject>File>Version

SubClasses of Version

CheckedOutFile



Properties Specific to Version

Properties	Inherited From	Description
Comment	VOBObject	Comment associated with the VOB object.
CreatedBy	VOBObject	User who created the object.
CreatedOn	VOBObject	Date the object was created.
ExtendedPath	File	VOB-extended path of this file system object.

Extension	File	File extension (the portion after the final dot).
Identifier		The version's identifier string.
IsCheckedOut		Whether or not this object represents a checked-out file.
IsDifferent		Whether or not this version is different from its predecessor.
IsDirectory	File	Whether or not the file is a directory.
IsHijacked		Whether or not this version is hijacked.
IsLatest		Whether or not this version is the latest on its branch.
Name	VOBObject	Name of the versioned object.
NameMinusExtension	File	Simple name of the file without the extension and final.dot.
OID	VOBObject	The object identifier for the VOB object.
Path	File	Path to this file system object.
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
SimpleName	File	Simple name of the file, that is, the name of the file without the path.
TypeName	VOBObject	The VOBObject type name.
VersionNumber		This version's version number.
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to Version

Name	Kind	Class	Description
Branch	0..1	Branch	The branch for this version.
Element	0..1	Element	This version's element.
Labels	0..n	Label	The collection of labels associated with this version.

ParentDirectory	0..1	Version	The current view's version of this version's parent directory.
Predecessor	0..1	Version	This version's predecessor version.
SubBranches	0..n	Branch	Any branches sprouting from this version.
VersionsInDirectory	0..n	Version	Represents the file and directory versions contained in this (directory) version.

View (ClearCase Adapter)

A View is a ClearCase object that provides a work area for one or more users. Users in different views can work with the same files without interfering with each other. For each element in a VOB, a view's configspec selects one version from the element's version tree, which is visible within the view. Each view can also store view-private files and view-private directories, which do not appear in other views. View-private objects and directories are not represented by any class within the ClearCase domain, however, they may be documented through the File System domain. The ClearCase domain enables you to identify snapshot and dynamic views, as well as views that build nonshareable derived objects

Class Hierarchy: Artifact>View

SubClasses of View

View has no subclasses.

Properties Specific to View

Properties	Inherited From	Description
BuildsShareableDOs		Whether or not this view builds non-shareable derived objects.
ConfigSpec		Configuration spec for this view.
DisplayableConfigSpec		Displayable form of the config spec for this view.
Host		Host on which the storage area for this view resides.
IsActive		Whether or not the view is started on the local system.
IsSnapShot		Whether or not this view is a snapshot view.
TagName		The view-tag name.

Relationships Specific to View

Name	Kind	Class	Description
This class has no relationships.			

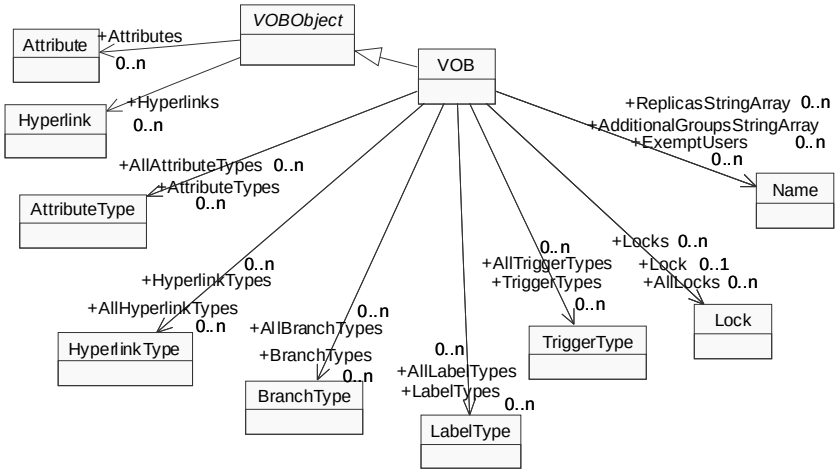
VOB (ClearCase Adapter)

A VOB is the database that stores your project's files. A VOB, or versioned object base, is a repository that stores versions for file elements, directory elements, derived objects, and meta-data associated with these objects. SoDA and Template Builder support MultiSite by enabling retrieval of a list of replicas (by name) for a given VOB. A template can include an OPEN command for a VOB, which must identify the VOB by full path, VOB-tag, or VOB family UUID.

Class Hierarchy: VOBObject>VOB

SubClasses of VOB

ProjectVOB



Properties Specific to VOB

Properties	Inherited From	Description
Comment	VOBObject	Comment associated with the VOB object.
CreatedBy	VOBObject	User who created the object.
CreatedOn	VOBObject	Date the object was created.
Group		Group to which this VOB belongs.
HasMSDOSTextMode		Whether or not this VOB has MS-DOS text mode enabled.

Host		Host on which the storage area for this VOB resides.
IsMounted		Whether or not the VOB is mounted.
IsReplicated		Whether or not this VOB is replicated.
LockDescription		Description of the lock for the VOB.
LockedBy		Name of the user who locked the VOB.
LockedOn		Date the VOB was locked
Name	VOBObject	Name of the versioned object.
NumberOfAdditionalGroups		Number of additional groups to which this VOB belongs.
NumberOfReplicas		The number of replica names for the VOB family of this VOB, if this VOB is replicated.
OID	VOBObject	The object identifier for the VOB object.
Owner		Owner of the VOB.
Selector	VOBObject	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
State		State of the lock on the VOB.
TagName		The VOB-tag name.
ThisReplica		Replica name for this VOB, if the VOB is replicated.
TypeName	VOBObject	The VOBObject type name.
VOBFamilyUUID	VOBObject	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to VOB

Name	Kind	Class	Description
AdditionalGroupsStringArray	0..n	Name	Additional groups to which this VOB belongs.
AllAttributeTypes	0..n	AttributeType	All existing attribute types in the VOB, including obsolete types.

AllBranchTypes	0..n	BranchType	All existing branch types in the VOB, including obsolete types.
AllHyperlinkTypes	0..n	HyperlinkType	All existing hyperlink types in the VOB, including obsolete types.
AllLabelTypes	0..n	LabelType	All existing label types in the VOB, including obsolete types.
AllLocks	0..n	Lock	An enumeration of all the locks in this VOB, including obsolete locks.
AllTriggerTypes	0..n	TriggerType	All existing trigger types in the VOB, including obsolete types.
AttributeTypes	0..n	AttributeType	All existing attribute types in the VOB.
BranchTypes	0..n	BranchType	All existing branch types in the VOB.
ExemptUsers	0..n	Name	The list of users exempted from the lock on the VOB.
HyperlinkTypes	0..n	HyperlinkType	All existing hyperlink types in the VOB.
LabelTypes	0..n	LabelType	All existing label types in the VOB.
Lock	0..1	Lock	The lock on this VOB, if there is one.
Locks	0..n	Lock	An enumeration of all the locks in this VOB.
ReplicasStringArray	0..n	Name	The array of replica names for the VOB family of this VOB, if this VOB is replicated.
TriggerTypes	0..n	TriggerType	All existing trigger types in the VOB.

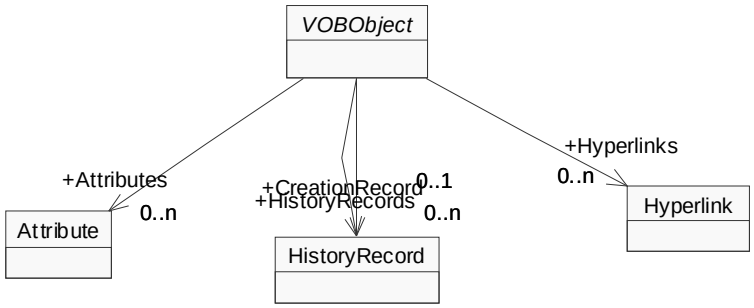
VOBObject (ClearCase Adapter)

A VOB Object represents an object stored in a VOB, including elements, versions, types, hyperlinks, branches, activities, and so on. VOBObject is the base class from which all other VOB object classes derive.

Class Hierarchy: Artifact>VOBObject

SubClasses of VOBObject

Activity
AttributeType
Branch
BranchType
File
Hyperlink
HyperlinkType
LabelType
TriggerType
UCMObject
VOB



Properties Specific to VOBObject

Properties	Inherited From	Description
Comment		Comment associated with the VOB object.
CreatedBy		User who created the object.
CreatedOn		Date the object was created.

Name	Name of the versioned object.
OID	The object identifier for the VOB object.
Selector	Expression used by the ClearCase file typing mechanism to match a file system object (or just the name of one).
TypeName	The VOBObject type name.
VOBFamilyUUID	The VOB family UUID for the VOB of this VOB object.

Relationships Specific to VOBObject

Name	Kind	Class	Description
Attributes	0..n	Attribute	The collection of attributes associated with this VOB object.
CreationRecord	0..1	HistoryRecord	Creation record for the VOB object.
HistoryRecords	0..n	HistoryRecord	The collection of history records for this object.
Hyperlinks	0..n	Hyperlink	The collection of hyperlinks associated with this VOB object.

ClearQuest

Rational ClearQuest

The following Classes are available through the ClearQuest RSE adapter:

- Attachments
- CQDatabase
- Groups
- History
- Query
- Record
- Users

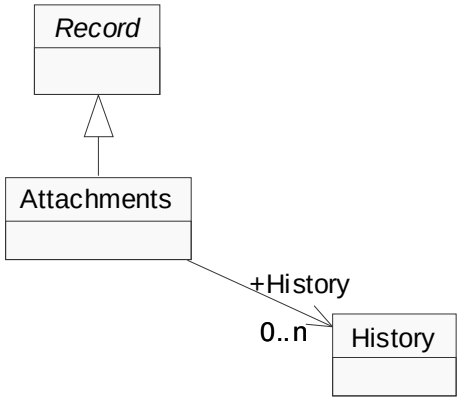
Attachments (ClearQuest Adapter)

An attachment is a file associated with a particular record in the database.

Class Hierarchy: Artifact>Record>Attachments

SubClasses of Attachments

Attachments has no subclasses.



Properties Specific to Attachments

Properties	Inherited From	Description
Dbid	Record	The internal database ID for the record.
Description		The description of the attachment
Entity_dbid		The database ID for the record that owns (has) the attachment. This identifies the database record (entity) that the attachment record belongs to. An Entity object represents a record in the database.
Entity_fielddef_id		The ID of the field definition (fieldInfo) for a field in the record (entity) that references the attachment. This ID identifies the attachment.
Filename		The name of the attachment

Filesize		The size of the attachment
FullPath		The location of the attachment. When the FullPath of an Attachment is requested, the attached file is retrieved from ClearQuest and copied to a temporary file on the local system. The name of this temporary file is returned by FullPath. The temporary file is deleted when the record containing the attachment is released.
Id	Record	The record ID.
Is_active	Record	True if the record is active.
Lock_version	Record	The version of the locking mechanism.
Locked_by	Record	The user who locked the record.
Record_type	Record	The type of record.
Version	Record	The version of the record.

Relationships Specific to Attachments

Name	Kind	Class	Description
History	0..n	History	The history records for this attachment.

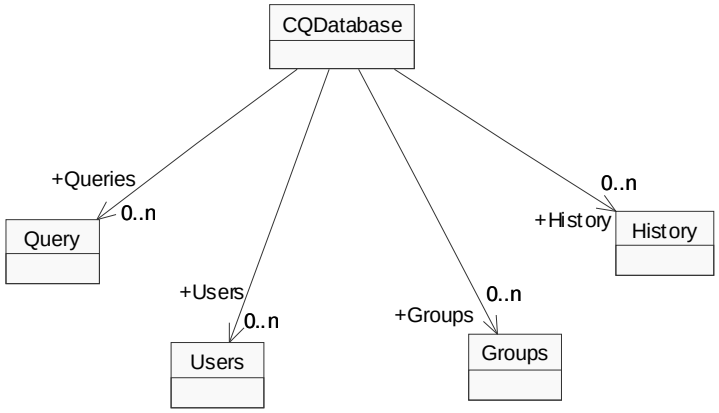
CQDatabase (ClearQuest Adapter)

A CQDatabase contains all user data and a copy of the associated schema.

Class Hierarchy: Artifact>CQDatabase

SubClasses of CQDatabase

CQDatabase has no subclasses.



Properties Specific to CQDatabase

Properties	Inherited From	Description
DatabaseName		Name of the database.
DatabaseSet		DatabaseSet is the name given to a master database or schema. The DatabaseSet name is chosen by the individual user when defining the DatabaseSet.
Description		The description of the database.
MinimizeSpace		MinimizeSpace effects the performance of the adapter by changing how queries are invoked. By default, MinimizeSpace is False. When it is False, the adapter requests that queries return all simple fields of each result record (this

	excludes multiline fields such as Description and fields that contain lists). If MinimizeSpace is TRUE, the adapter only requests the result fields that are already defined by queries.
Name	The logical name of the database. The Database Name is the name of a collection of CQ records that conform to the master database or schema.
QualifiedName	The qualified name of the database. The qualified name consists of DatabaseSet and the database Name. (<DatabaseSet>:<DatabaseName>) For example, CMBU:LABST
SessionType	The database session type.

Relationships Specific to CQDatabase

Name	Kind	Class	Description
Groups	0..n	Groups	All groups stored in the database.
History	0..n	History	The history records for this database.
Queries	0..n	Query	All queries stored in the database.
Users	0..n	Users	All users stored in the database.

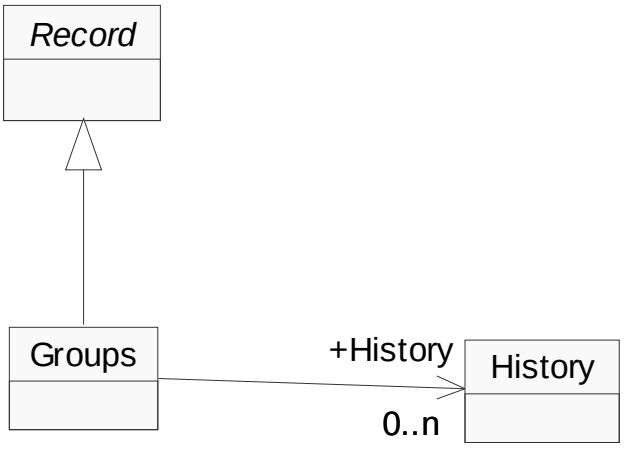
Groups (ClearQuest Adapter)

A group is a list of users with similar privileges.

Class Hierarchy: Record>Groups

SubClasses of Groups

Groups has no subclasses.



Properties Specific to Groups

Properties	Inherited From	Description
Dbid	Record	The internal database ID for the record.
Id	Record	The record ID.
Is_active	Record	True if the record is active.
Lock_version	Record	The version of the locking mechanism.
Locked_by	Record	The user who locked the record.
Master_dbid		The master database ID for the Groups object. A master database is a schema repository for one or more user databases.
Name		The name of the group collection.

Record_type	Record	The type of record.
Version	Record	The version of the record.

Relationships Specific to Groups

Name	Kind	Class	Description
History	0..n	History	The history records of this group.

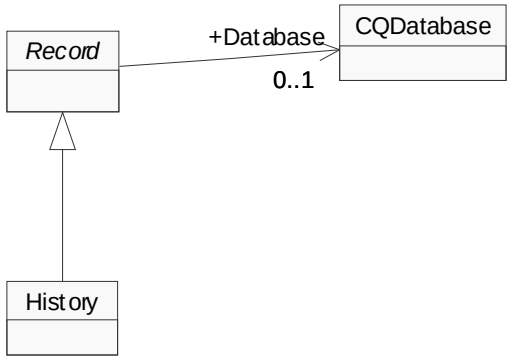
History (ClearQuest Adapter)

History records all changes made to the records in the database.

Class Hierarchy: Record>History

SubClasses of History

History has no subclasses.



Properties Specific to History

Properties	Inherited From	Description
Action_name		The action that was entered.
Action_timestamp		The time the action was entered.
Comments		Any comments associated with the event.
Dbid	Record	The internal database ID for the record.
Entity_dbid		The database ID for the record that owns (has) the history. This identifies the database record (entity) that the History record belongs to. An Entity object represents a record in the database.

Entitydef_id		The database ID of the record type.
Entitydef_name		The name of the record type.
Expired_timestamp		The time the history expires.
Id	Record	The record ID.
Is_active	Record	True if the record is active.
Lock_version	Record	The version of the locking mechanism.
Locked_by	Record	The user who locked the record.
New_state		The state of the record following the action.
Old_state		The state of the record prior to the action.
Record_type	Record	The type of record.
User_name		The user who triggered the action.
Version	Record	The version of the record.

Relationships Specific to History

Name	Kind	Class	Description
This class has no relationships.			

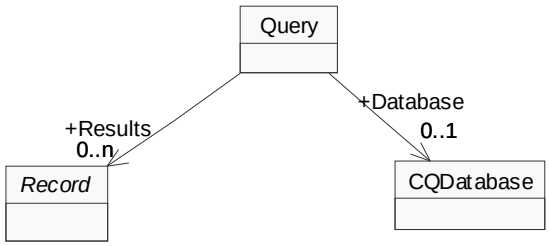
Query (ClearQuest Adapter)

A query is used to retrieve specific records from a database.

Class Hierarchy: Artifact>Record>Query

SubClasses of Query

Query has no subclasses.



Properties Specific to Query

Properties	Inherited From	Description
IsAggregated		Returns a Boolean value indicating whether any fields of the query are aggregated.
IsDirty		Returns a Boolean value indicating whether the query has changed.
IsFamilyQuery		Returns true if the Query defines a family.
Name		The name of the Query.
QueryType		The type of query. An integer indicating list, report, or chart.

ResultType	The result type of the query.
SQL	The SQL string associated with the query.

Relationships Specific to Query

Name	Kind	Class	Description
Database	0..1	CQDatabase	The CQDatabase that the Query is associated with.
Results	0..n	Record	The records associated with a Query.

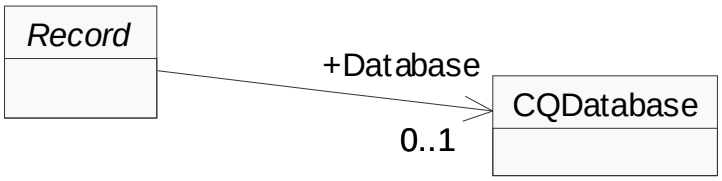
Record (ClearQuest Adapter)

Records represent the data records the user creates, modifies, and views.

Class Hierarchy: Artifact>Record

SubClasses of Record

Attachments
Groups
History
Users



Properties Specific to Record

Properties	Inherited From	Description
Dbid		The internal database ID for the record.
Id		The record ID.
Is_active		True if the record is active.
Lock_version		The version of the locking mechanism.
Locked_by		The user who locked the record.
Record_type		The type of record.
Version		The version of the record.

Relationships Specific to Record

Name	Kind	Class	Description
Database	0..1	CQDatabase	The CQDatabase which this record is associated with.

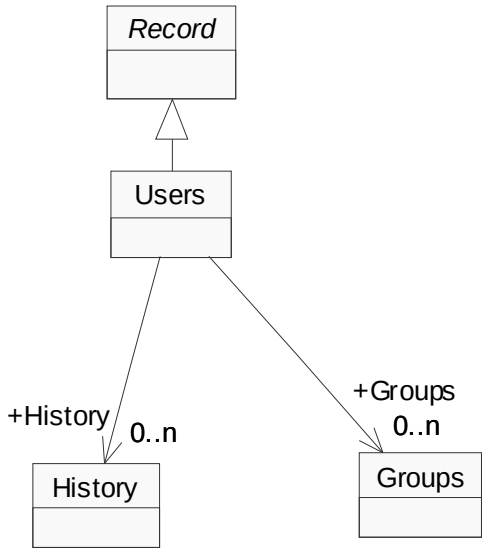
Users (ClearQuest Adapter)

A user is someone who can log on to the ClearQuest database.

Class Hierarchy: Record>Users

SubClasses of Users

Users has no subclasses.



Properties Specific to Users

Properties	Inherited From	Description
Dbid	Record	The internal database ID for the record.
Email		The email address of the user.
Encrypted_password		The password of the user.
Fullname		The full name of the User.
Id	Record	The record ID.
Is_active	Record	True if the record is active.
Is_appbuilder		True if the user has AppBuilder privileges.

Is_superuser		True if the user is a superuser.
Is_user_maint		True if the user has maintenance privileges.
Lock_version	Record	The version of the locking mechanism.
Locked_by	Record	The user who locked the record.
Login_name		The login name for the User.
Master_dbid		The master database ID. A master database is a schema repository for one or more user databases.
Misc_info		Miscellaneous information.
Phone		The phone number of the user.
Record_type	Record	The type of record.
Version	Record	The version of the record.

Relationships Specific to Users

Name	Kind	Class	Description
Groups	0..n	Groups	The groups of which this user is a member.
History	0..n	History	The history records for this user record.

FileSys

Microsoft File System

The following Classes are available through the FileSys RSE adapter:

- Directory
- DirectoryObject
- File
- FileRecord

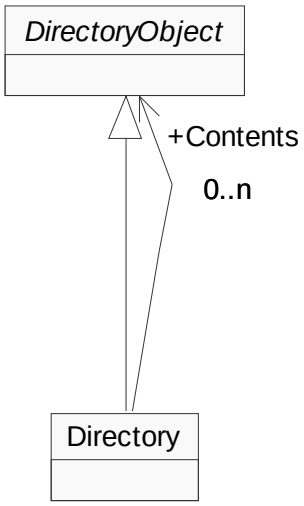
Directory (FileSys Adapter)

A directory, sometimes called a folder, contains other files or directories.

Class Hierarchy: DirectoryObject>Directory

SubClasses of Directory

Directory has no subclasses.



Properties Specific to Directory

Properties	Inherited From	Description
DirectoryPath		Path of the directory.
DriveLetter	DirectoryObject	Drive letter of the location of the directory.
Extension	DirectoryObject	The segment of a SimpleName following the last period. For example, the Extension of C:\bill\file.txt is txt. If the SimpleName contains no period, then Extension returns a null string.

NameMinusExtension	DirectoryObject	The segment of a SimpleName preceding the last period. For example, the NameMinusExtension of C:\bill\file.test.txt is file.test. If the SimpleName contains no period, then NameMinusExtension returns the SimpleName.
NamePrefix	DirectoryObject	The segment of a SimpleName preceding the first period in the file name. For example, the NamePrefix of C:\bill\file.test.txt is file.
Path	DirectoryObject	The complete path of an object. For example, C:\bill\file.txt.
SimpleName	DirectoryObject	The context-independent portion of an object's name. For example, the SimpleName of C:\bill\file.txt is file.txt.

Relationships Specific to Directory

Name	Kind	Class	Description
Contents	0..n	DirectoryObject	The DirectoryObjects that reside within the directory (subdirectories are included but not their contents).

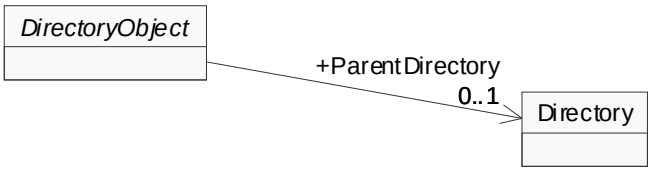
DirectoryObject (FileSys Adapter)

Anything that can be found in a Directory, including files and (sub)directories.

Class Hierarchy: Artifact>DirectoryObject

SubClasses of DirectoryObject

Directory
File



Properties Specific to DirectoryObject

Properties	Inherited From	Description
DriveLetter		Drive letter of the location of the directory.
Extension		The segment of a SimpleName following the last period. For example, the Extension of C:\bill\file.txt is txt. If the SimpleName contains no period, then Extension returns a null string.
NameMinusExtension		The segment of a SimpleName preceding the last period. For example, the NameMinusExtension of C:\bill\file.test.txt is file.test. If the SimpleName contains no period, then NameMinusExtension returns the SimpleName.
NamePrefix		The segment of a SimpleName preceding the first period in the file name. For example, the NamePrefix of C:\bill\file.test.txt is file.

Path	The complete path of an object. For example, C:\bill\file.txt.
SimpleName	The context-independent portion of an object's name. For example, the SimpleName of C:\bill\file.txt is file.txt.

Relationships Specific to DirectoryObject

Name	Kind	Class	Description
ParentDirectory	0..1	Directory	Directory containing the object. If you try to object ParentDirectory from the root directory, SoDA and Template Builder generate an error.

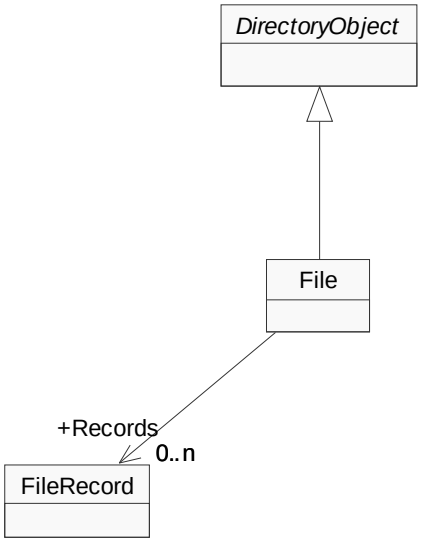
File (FileSys Adapter)

A subclass of DirectoryObject that does not contain other files or directories. Files can be ASCII or binary. They can contain text, bitmaps, program source, object code, executable code, or any other form of information that can be stored in a file. Note that the Graphic and Text attributes may not be defined for certain types of files.

Class Hierarchy: DirectoryObject>File

SubClasses of File

File has no subclasses.



Properties Specific to File

Properties	Inherited From	Description
DriveLetter	DirectoryObject	Drive letter of the location of the directory.
Extension	DirectoryObject	The segment of a SimpleName following the last period. For example, the Extension of C:\bill\file.txt is txt. If the SimpleName contains no period, then Extension returns a null string.

FilePath		Path of the file.
NameMinusExtension	DirectoryObject	The segment of a SimpleName preceding the last period. For example, the NameMinusExtension of C:\bill\file.test.txt is file.test. If the SimpleName contains no period, then NameMinusExtension returns the SimpleName.
NamePrefix	DirectoryObject	The segment of a SimpleName preceding the first period in the file name. For example, the NamePrefix of C:\bill\file.test.txt is file.
Path	DirectoryObject	The complete path of an object. For example, C:\bill\file.txt.
SimpleName	DirectoryObject	The context-independent portion of an object's name. For example, the SimpleName of C:\bill\file.txt is file.txt.
Text		The complete contents of an ASCII text file. Undefined for other file types.

Relationships Specific to File

Name	Kind	Class	Description
Records	0..n	FileRecord	The records contained in a text file. By default, SoDA and Template Builder use newlines to distinguish separate records within a file. It is possible to override this default by including a RECORD_DELIMITER directive in the file.

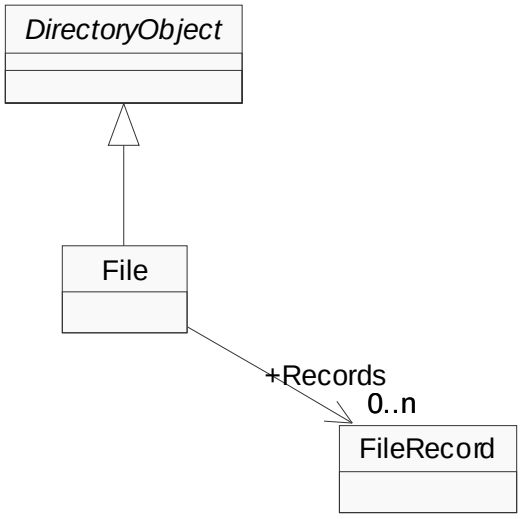
FileRecord (FileSys Adapter)

ASCII text files can be further decomposed into file records. File records are especially useful for parsing flat database files. Records must contain key fields that uniquely identify each record. By default the first field is the key.

Class Hierarchy: Artifact>FileRecord

SubClasses of FileRecord

FileRecord has no subclasses.



Properties Specific to FileRecord

Properties	Inherited From	Description
Field01		Text of the specified field, numbered from left to right. The extent of each field is determined by the field delimiter character, which defaults to a space. You can change the default by including a FIELD_DELIMITER directive in your file. Double quotes (") can be used to designate a single field that includes field delimiters. You can change the

	quote character by including a QUOTE_DELIMITER directive in your file.
Field02	
Field03	
Field04	
Field05	
Field06	
Field07	
Field08	
Field09	
Field10	
Field11	
Field12	
Field13	
Field14	
Field15	
Field16	
Field17	
Field18	
Field19	
Field20	
Field21	
Field22	
Field23	
Field24	
Field25	
Field26	
Field27	
Field28	
Field29	
Field30	

Filename	The full path of the file that contains this record.
Position	
Text	The complete contents of an ASCII text file. Undefined for other file types.
UniqueKey	The field or combination of fields used to uniquely identify the record. The default is to use the first field as the unique key. You can change the default by including a KEY_FIELDS directive in your file. If you have used the KEY_FIELDS directive to specify a multiple-field key, you enter a key by supplying each field, in order, separated by the field delimiter.

Relationships Specific to FileRecord

Name	Kind	Class	Description
This class has no relationships.			

MSProject

Microsoft Project

The MSProject adapter returns Date, Duration, Work, and Cost properties as variants.

The following Classes are available through the MSProject RSE adapter:

- Assignment
- Project
- Resource
- Task
- TaskDependency

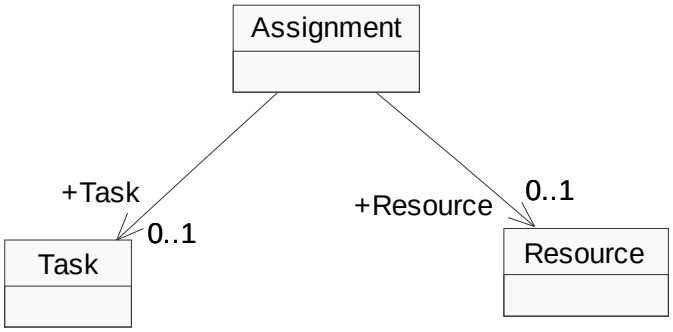
Assignment (MSProject Adapter)

An assignment for a task or a resource.

Class Hierarchy: Artifact>Assignment

SubClasses of Assignment

Assignment has no subclasses.



Properties Specific to Assignment

Properties	Inherited From	Description
ActualCost		The actual cost for this assignment.
ActualWork		The actual work for this assignment.
BaselineFinish		The finish of this assignment.
BaselineStart		The start of this assignment.
Cost		Estimated cost of this assignment.
RemainingCost		Remaining cost for this assignment.
RemainingWork		Remaining work, in hours, for this assignment.
ResourceName		Name of the Resource associated with this assignment.
ResourceUniqueID		The ID of the Resource associated with this assignment.

TaskName	Name of the Task associated with this assignment.
TaskUniqueID	The ID of the task associated with this assignment.
UniqueID	The unique ID of the assignment.
Units	The percentage, or number of units, for which a resource is assigned to a task.
Work	The work for this assignment.

Relationships Specific to Assignment

Name	Kind	Class	Description
Resource	0..1	Resource	The Resource assigned to a task.
Task	0..1	Task	The Task associated with the assignment.

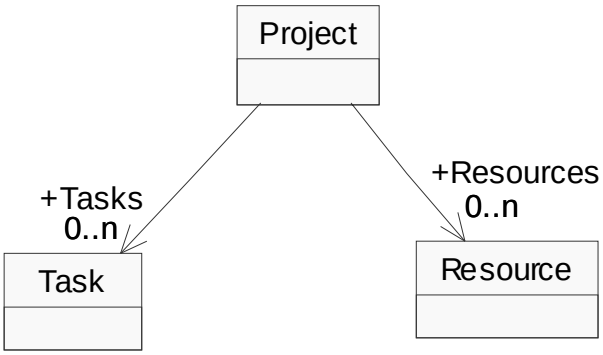
Project (MSProject Adapter)

A project is a collection of data, including assignments, resources, and tasks.

Class Hierarchy: Artifact>Project

SubClasses of Project

Project has no subclasses.



Properties Specific to Project

Properties	Inherited From	Description
Directory		The directory for this project.
FinishDate		Finish date of the project.
Name		Name of the project.
Path		The file path of the project.
StartDate		Start date of the project.

Relationships Specific to Project

Name	Kind	Class	Description
Resources	0..n	Resource	The Resources for the Project.
Tasks	0..n	Task	The Tasks of the Project.

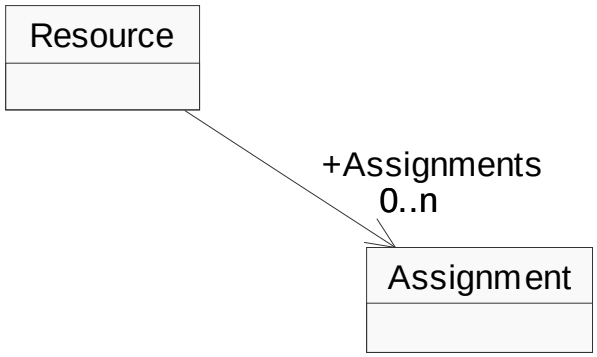
Resource (MSProject Adapter)

A single resource.

Class Hierarchy: Artifact>Resource

SubClasses of Resource

Resource has no subclasses.



Properties Specific to Resource

Properties	Inherited From	Description
BaselineCost		The planned cost for a resource for all assigned tasks.
BaselineWork		The planned work for a resource for all assigned tasks.
CostPerUse		The cost for each time the resource is used.
Group		Name of the group to which this resource belongs.
Name		Name of the Resource.
RemainingCost		The expense to complete the remaining work assigned to a resource.
RemainingWork		The remaining amount of time for a resource to complete all assigned tasks.
UniqueID		The unique ID for this Resource.

Relationships Specific to Resource

Name	Kind	Class	Description
Assignments	0..n	Assignment	The assignments for this Resource.

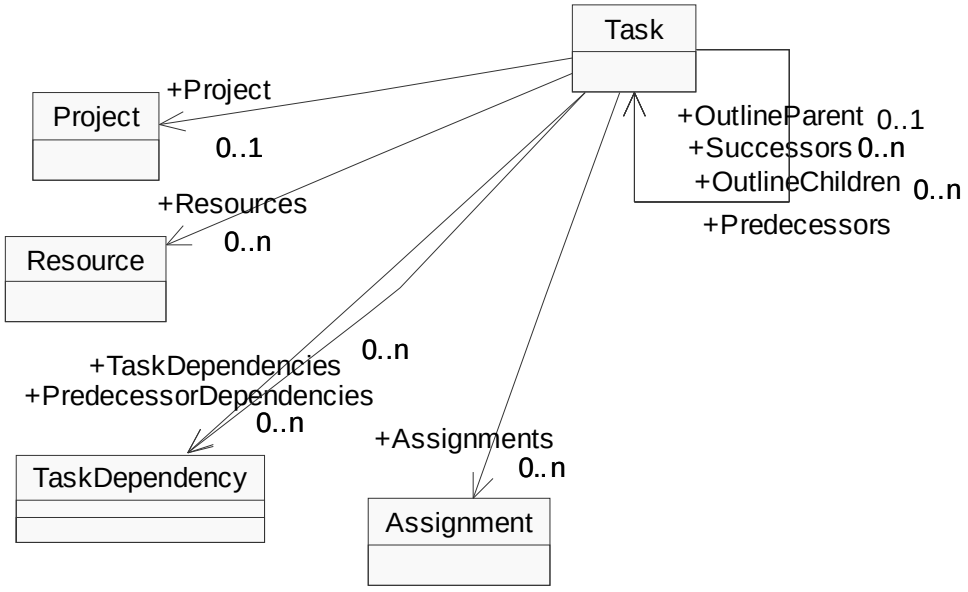
Task (MSProject Adapter)

A specific piece of work to be done.

Class Hierarchy: Artifact>Task

SubClasses of Task

Task has no subclasses.



Properties Specific to Task

Properties	Inherited From	Description
ActualCost		Costs incurred for work already performed by all resources on a task, along with any other recorded costs associated with the task.
ActualDuration		Actual working time for a task, based on the scheduled duration and current remaining work.
ActualFinish		Actual finish date and time of the task.

ActualStart	Actual start date and time of the task.
ActualWork	Actual work that has been done by the resources assigned to the task.
ACWP	Costs incurred for work already done on the task.
BaselineCost	Baseline cost for the Task.
BaselineDuration	Baseline duration for the Task.
BaselineFinish	Baseline finish for the Task.
BaselineStart	Baseline start of the Task.
BaselineWork	Baseline work for the Task.
BCWP	Value of the task's PercentComplete multiplied by the task's BaselineCost.
BCWS	Baseline costs up to the status date or today's date.
ConstraintDate	Date associated with the constraint type.
ConstraintType	The type of constraint for this task.
Cost	Total projected cost for the Task.
CostVariance	Difference between the baseline cost and total cost for the Task.
Critical	True if the Task is critical.
Duration	Total working time for a task.
DurationVariance	Difference between the Task's baseline duration and the total duration.
EffortDriven	True if the Task is effort driven.
Finish	Date and time that the Task is scheduled to be completed.
FinishVariance	Difference between the Task's baseline finish date and its current finish date.
FixedCost	Costs not associated with a Resource for the Task.
FixedCostAccrual	Accrual method for managing the Task's fixed cost.
ID	The unique ID for this Task.
Milestone	True if this Task is a milestone.

Name	Name of the Task.
Notes	Notes for this Task.
OutlineLevel	Outline level of the Task in the Project.
OutlineNumber	Outline number of the task.
OvertimeCost	Total overtime cost for the Task.
OvertimeWork	Overtime work scheduled for this Task.
PercentComplete	Percentage of the task's duration that has been completed.
PercentWorkComplete	Percentage of the task's work that has been completed.
Priority	Level of importance set for this Task.
RegularWork	Regular work scheduled for the Task.
RemainingCost	Remaining cost to complete the Task.
RemainingDuration	Remaining amount of time to complete the Task.
RemainingOvertimeCost	Remaining overtime cost to complete the Task.
RemainingOvertimeWork	Remaining overtime work to complete the Task.
RemainingWork	Remaining work to complete the Task.
Start	Start date and time set for the Task.
StartVariance	Difference between the baseline start date and the scheduled start date for the Task.
Type	The Task type.
UniqueID	The unique ID of the Task.
WBS	Work breakdown structure code for the Task.
Work	The Task's total amount of work scheduled to all assigned resources.
WorkVariance	Difference between the baseline work and the scheduled work for the Task.

Relationships Specific to Task

Name	Kind	Class	Description
Assignments	0..n	Assignment	Assignments for this Task.
OutlineChildren	0..n	Task	The Task's children Tasks.
OutlineParent	0..1	Task	The Task's parent Task.
PredecessorDependencies	0..n	TaskDependency	
Predecessors	0..n	Task	Associated Tasks that must start or finish before this Task starts or finishes.
Project	0..1	Project	The Project this Task is associated with.
Resources	0..n	Resource	Resources for this Task.
Successors	0..n	Task	Associated Tasks that rely on this Task to start or finish.
TaskDependencies	0..n	TaskDependency	

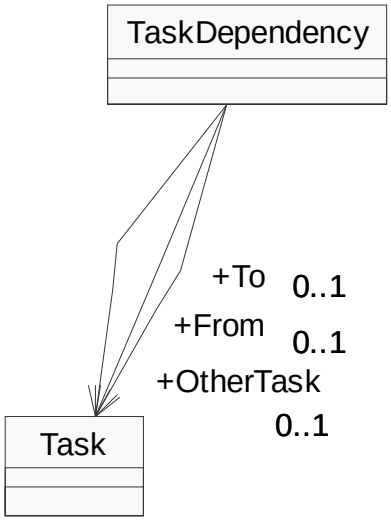
TaskDependency (MSProject Adapter)

A dependency between two linked tasks.

Class Hierarchy: Artifact>Task

SubClasses of TaskDependency

TaskDependency has no subclasses.



Properties Specific to TaskDependency

Properties	Inherited From	Description
Lag		A time delay between the tasks that have a dependency.
OtherTaskID		The ID of another task associated with the task dependency.
Type		The kind of task dependency.

Relationships Specific to TaskDependency

Name	Kind	Class	Description
From	0..1	Task	The dependent task in the dependency.
OtherTask	0..1	Task	Another task associated with the task dependency.
To	0..1	Task	The task that the From task is depending on.

RAdmin

Rational Administrator

The following Classes are available through the RAdmin RSE adapter:

- RAProject
- RAServer
- RoseModel

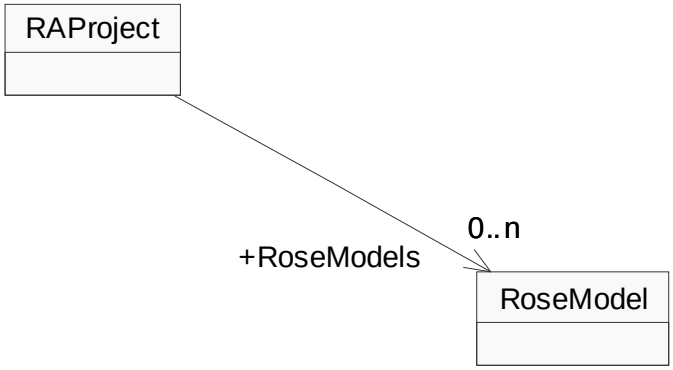
RAProject (RAdmin Adapter)

A project stores software testing and development information.

Class Hierarchy: Artifact>RAProject

SubClasses of RAProject

RAProject has no subclasses.



Properties Specific to RAProject

Properties	Inherited From	Description
ClearQuestDatabaseName		The ClearQuestDatabase (CQDatabase object) name associated with the RAProject.
Location		Location of the RAProject.
Name		Name of the RAProject.
Path		The file path of the RAProject.
RequirementsCMManaged		This property returns a Boolean value indicating whether the Requirements

	associated with the RA Project may be placed under CM.
RequisiteDatastorePath	The path of RequisitePro datastore that contains the CMManaged Requirements.
TestAssetsCMManaged	This property returns a Boolean value indicating whether the Test Manager Test Assets associated with the RA Project may be placed under CM.
TestDatastorePath	The path of the Test Manager datastore that contains the CMManaged TestAssets.
UCMEnabled	True if use case management is enabled.

Relationships Specific to RAProject

Name	Kind	Class	Description
RoseModels	0..n	RoseModel	List all Rose models used in the project.

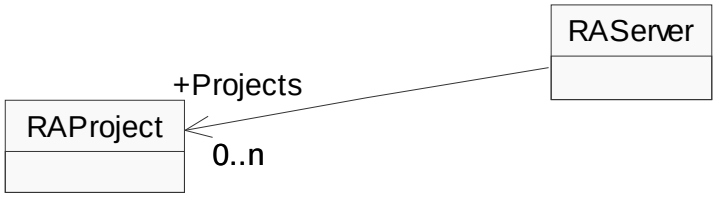
RAServer (RAdmin Adapter)

Provides access to all the other interfaces which provide access to the list of registered projects (RAProjects), the list of registered SQLAnywhere servers (RASQLAnywhereServers) and the ability to start various Rational tools (RationalTools).

Class Hierarchy: Artifact>RAServer

SubClasses of RAServer

RAServer has no subclasses.



Properties Specific to RAServer

Properties	Inherited From	Description
This class has no properties.		

Relationships Specific to RAServer

Name	Kind	Class	Description
Projects	0..n	RAProject	Associated RAProjects.

RoseModel (RAdmin Adapter)

Rose Models associated with an RAProject.

Class Hierarchy: Artifact>RoseModel

SubClasses of RoseModel

RoseModel has no subclasses.

Properties Specific to RoseModel

Properties	Inherited From	Description
Name		Model name.
Path		Model path.

Relationships Specific to RoseModel

Name	Kind	Class	Description
This class has no relationships.			

ReqPro

Rational RequisitePro

The following Classes are available through the ReqPro RSE adapter:

- AttributeValue
- Discussion
- DocumentType
- Group
- Permission
- Project
- Relationship
- ReqDocument
- Requirement
- RequirementType
- Response
- Revision
- User
- View

AttributeValue (ReqPro Adapter)

Attributes are descriptive information attached to a requirement that provide important details about that requirement, such as priority, cost, or difficulty.

Class Hierarchy: Artifact>AttributeValue

SubClasses of AttributeValue

AttributeValue has no subclasses.

Properties Specific to AttributeValue

Properties	Inherited From	Description
DataType		Data type of this attribute.
Label		Name of the attribute.
Text		Text of the attribute.
ValueID		The value ID of the attribute.

Relationships Specific to AttributeValue

Name	Kind	Class	Description
This class has no relationships.			

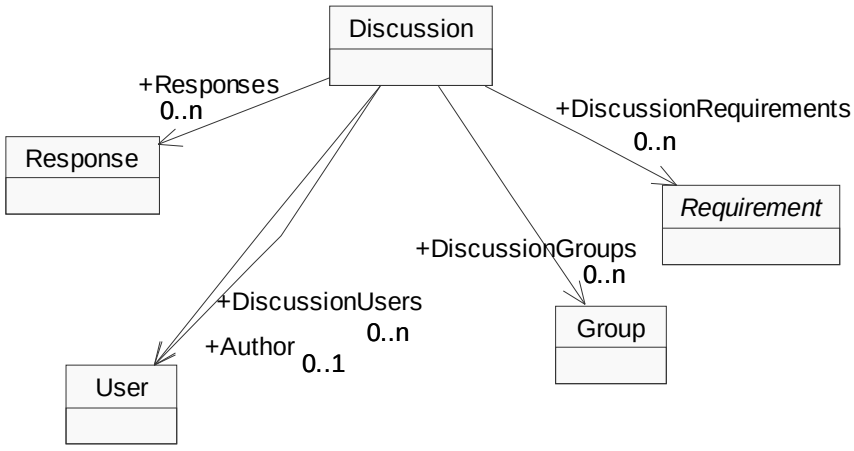
Discussion (ReqPro Adapter)

Discussions let RequisitePro users address comments, issues, and questions to a group of discussion participants. Discussions can be associated with one or more specific requirements, or refer to the project in general.

Class Hierarchy: Artifact>Discussion

SubClasses of Discussion

Discussion has no subclasses.



Properties Specific to Discussion

Properties	Inherited From	Description
DateTime		When the discussion was created.
DiscussionID		The ID of the discussion.
Message		The text of the discussion.
Priority		Priority of the discussion: High, Medium, or Low.
Restricted		True if the discussion is restricted to the listed participants.
Status		Status of the discussion: Open or Closed.
Subject		Subject of the discussion.

Relationships Specific to Discussion

Name	Kind	Class	Description
Author	0..1	User	User who created this response.
DiscussionGroups	0..n	Group	Groups associated with the discussion.
DiscussionRequirements	0..n	Requirement	An associated requirement to the discussion.
DiscussionUsers	0..n	User	Users are participants in the discussion.
Responses	0..n	Response	Responses to this discussion.

DocumentType (ReqPro Adapter)

A document type is a template that is applied to your documents. The template can include the default font for your document, the available heading and paragraph styles, and the default type of requirements for the document. Or it could encompass both formatting conventions and an outline that helps you organize your requirements information.

Class Hierarchy: Artifact>DocumentType

SubClasses of DocumentType

DocumentType has no subclasses.

Properties Specific to DocumentType

Properties	Inherited From	Description
Description		Purpose and content of the document type.
Extension		File extension applied to all documents associated with this document type.
Name		Name of the document type.
TemplateDesc		A description of the template, or outline, for this document type.
TemplateFilename		Filename of the template, or outline, used when documents of this type are created.
TemplateName		Name of the template, or outline, used when documents of this type are created.
TypeID		Document type ID.

Relationships Specific to DocumentType

Name	Kind	Class	Description
RequirementType	0..1	RequirementType	The default type of requirement stored in this type of document.

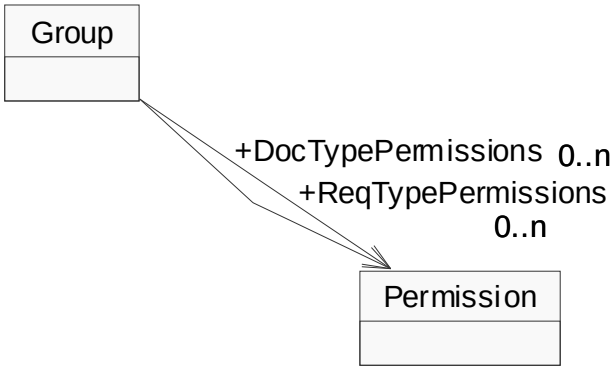
Group (ReqPro Adapter)

Groups are used for project security. A user group is a list of users, defined in project security and organized by the operations they have privileges to perform. For example, members of the Administrators group can create group accounts and add users to groups.

Class Hierarchy: Artifact>Group

SubClasses of Group

Group has no subclasses.



Properties Specific to Group

Properties	Inherited From	Description
DefAttrPermissions		Privileges this group has to create or modify attributes.
DefDocTypePermissions		Privileges this group has to create or modify document types.
DefListItemPermissions		Privileges this group has to create or modify list items.
DefProjPermissions		Privileges this group has to create or modify privileges.
DefReqTypePermissions		Privileges this group has to create or modify requirements.
GroupID		Group ID.
Name		Name of the user group.

ProjectPermissions	Project permissions of the Group.

Relationships Specific to Group

Name	Kind	Class	Description
DocTypePermissions	0..n	Permission	Document type permissions of the Group.
ReqTypePermissions	0..n	Permission	Requirement type permissions of the Group.

Permission (ReqPro Adapter)

A privilege granted to a group of RequisitePro users. RequisitePro administrators or members of a group with project security permissions can assign permissions to groups. Permission types include:

Project Permissions
Document Type and Requirement Type Permissions
Traceability Permissions

Class Hierarchy: Artifact>Permission

SubClasses of Permission

Permission has no subclasses.

Properties Specific to Permission

Properties	Inherited From	Description
IsModified		True if the permission has been modified.
PermissionID		Permission ID.
Permissions		Permissions for this Permission object. The permissions for a group relative to a particular attribute, list item, document type or requirement type. Allowed values are None, Read, Update, and Create.
TypeName		Name of the permission type.

Relationships Specific to Permission

Name	Kind	Class	Description
This class has no relationships.			

Project (ReqPro Adapter)

The concept of a project is used to provide the groundwork for organizing and effectively managing requirements. Each project resides in a separate directory. This storage method simplifies the process of organizing, archiving, and managing project files.

A project includes the following:

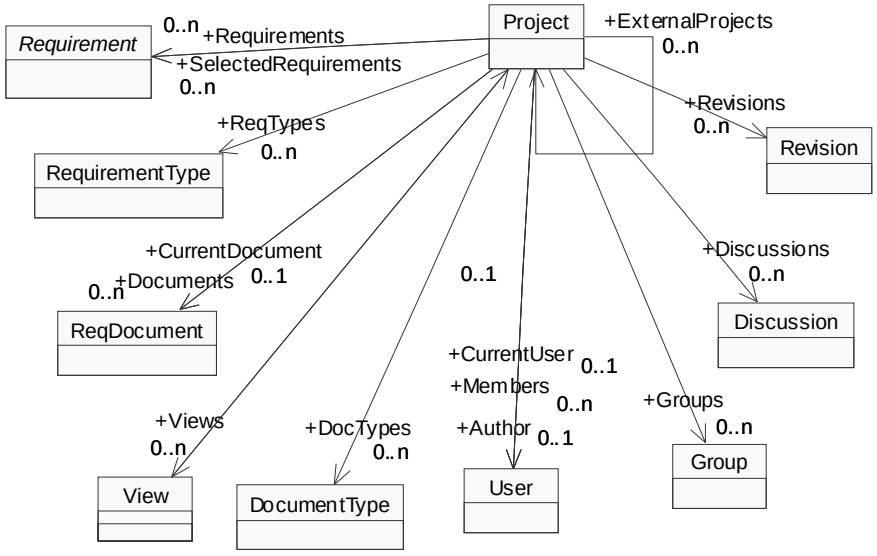
- Database
- Documents
- Document types
- Requirements and their attributes
- Requirement types
- Requirement traceability
- Discussions
- User and group security

Within RequisitePro, you first create a project. You then create requirement documents and requirements in each document.

Class Hierarchy: Artifact>Project

SubClasses of Project

Project has no subclasses.



Properties Specific to Project

Properties	Inherited From	Description
Description		Optional information describing the purpose and content of the project.
FileName		Filename of the Project.
Name		Name of the Project.
Path		Path of the Project.
Prefix		Prefix that is prepended to requirement tags when using external Projects.

Relationships Specific to Project

Name	Kind	Class	Description
Author	0..1	User	Creator of the Project.
CurrentDocument	0..1	ReqDocument	Document that is currently open (if any) in RequisitePro. No document is returned if the open document is not in the open Project.
CurrentUser	0..1	User	Current User of the Project.
Discussions	0..n	Discussion	Discussions associated with the Project.
DocTypes	0..n	DocumentType	Document types defined in this Project.
Documents	0..n	ReqDocument	Set of documents associated with this Project.
ExternalProjects	0..n	Project	External projects that have been attached to this Project.
Groups	0..n	Group	Security groups defined in this Project.
Members	0..n	User	All users who are registered in this Project.
ReqTypes	0..n	RequirementType	Requirement types defined in this Project.
Requirements	0..n	Requirement	All requirements stored in the Project database.
Revisions	0..n	Revision	Historical data of the Project.

SelectedRequirements	0..n	Requirement	Requirements that are currently selected in the current View in RequisitePro. Only requirements in the open Project are included.
Views	0..n	View	Views associated with this Project.

Relationship (ReqPro Adapter)

Traceability relationships are established between two or more requirements that exist in the same document, in different documents, or in the database.

Class Hierarchy: Artifact>Relationship

SubClasses of Relationship

Relationship has no subclasses.

Properties Specific to Relationship

Properties	Inherited From	Description
Direction		Direction of the relationship: TraceTo, TraceFrom, Parent, or Child.
RelationshipID		Relationship ID.
RelationshipType		Type of the relationship, either Hierarchical or Traceability.
Suspect		True if the relationship is suspect; otherwise, False.

Relationships Specific to Relationship

Name	Kind	Class	Description
RelatedReq	0..1	Requirement	The associated requirement.

ReqDocument (ReqPro Adapter)

A requirements document created in Microsoft Word or Rational RequisitePro that captures requirements and is used to communicate product development efforts. Each requirements document addresses a particular requirement type, such as product requirements, software specifications, or test plans.

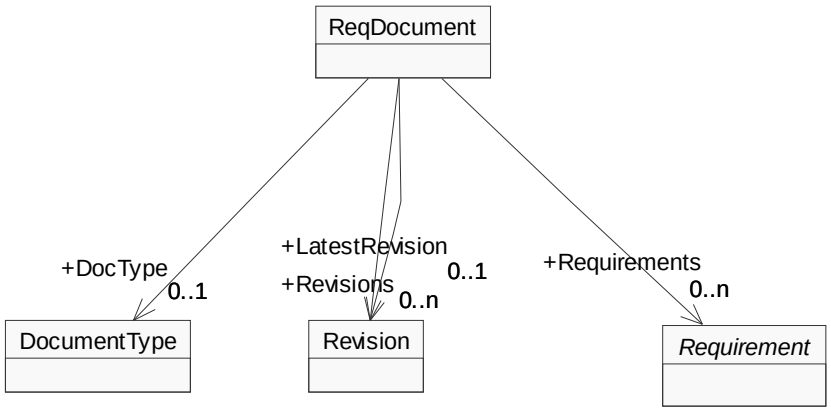
A requirements document differs from a Word document in that you can access requirement attributes and other information directly from within the requirements document.

Note: Also referred to as "document," and "RequisitePro document."

Class Hierarchy: Artifact>ReqDocument

SubClasses of ReqDocument

ReqDocument has no subclasses.



Properties Specific to ReqDocument

Properties	Inherited From	Description
Description		A description for the ReqDocument.
DocumentID		The ID of the ReqDocument.

Extension	The three letter extension of the ReqDocument file.
FileDateTime	Date of the ReqDocument.
FileName	Filename of the ReqDocument.
FullPath	Full path of the ReqDocument.
Name	Name of the ReqDocument.
Path	Path of the ReqDocument.

Relationships Specific to ReqDocument

Name	Kind	Class	Description
DocType	0..1	DocumentType	Document type associated with a ReqDocument.
LatestRevision	0..1	Revision	Latest Revision artifact associated with a ReqDocument.
Requirements	0..n	Requirement	Requirements associated with a ReqDocument.
Revisions	0..n	Revision	Revision artifacts associated with a ReqDocument.

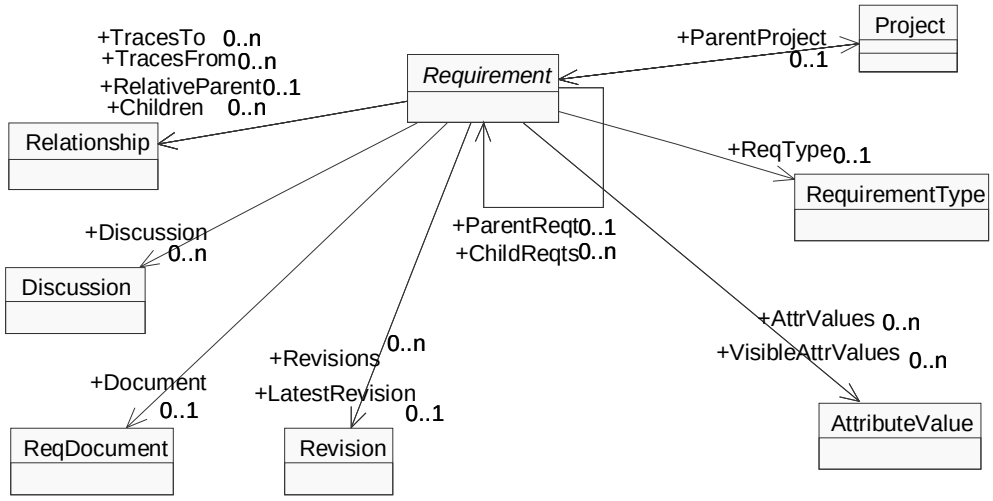
Requirement (ReqPro Adapter)

A requirement is the specification for the externally observable behavior of the system (for example, inputs to the system, outputs from the system, functions of the system, attributes of the system, or attributes of the system environment). In RequisitePro, a requirement defines an entity represented by: a piece of text, a set of attributes, and a set of traceability relationships.
If a template includes the name of a specific project, there will also be subclasses for each <Project-Specific Type>Requirement. RSE automatically creates a series of new classes that are subclasses of the Requirement Class. The name of the class is the concatenation of the requirement type and the word Requirement. For instance, if a Project contains requirement types PR, SR, and TST, the new classes will be PRRequirement, SRRequirement, and TSTRequirement.

Class Hierarchy: Artifact>Requirement

SubClasses of Requirement

Requirement has no subclasses.



Properties Specific to Requirement

Properties	Inherited From	Description
Bookmark		Name of the Word bookmark associated with this Requirement.

DocPosn	Relative position of the Requirement in the document. For instance, the second Requirement in the document would be position 2. Database-only requirements have position 0.
FullTag	Full tag of the requirement, such as PR1.
GUID	The GUI ID for the Requirement.
HasChildren	True if this requirement has child requirements; otherwise False.
HasParent	True if this Requirement has a parent requirement. This would be the same as Level > 0.
Level	The hierarchical level of the requirement. For instance, if the full tag is PR1.1, the level would be 1; PR1 would be level 0.
Name	Requirement name.
TagNumber	Number of the tag, such as 1.
TagPrefix	Prefix of the tag, such as PR.
Text	Text of the requirement.
UniqueID	The unique ID for the Requirement.

Relationships Specific to Requirement

Name	Kind	Class	Description
AttrValues	0..n	AttributeValue	The AttributeValue artifacts for a given Requirement. AttrValues includes both the hidden and visible attributes.
Children	0..n	Relationship	The child Relationship artifacts of this Requirement. The Relationship artifacts represent the child Requirement artifacts returned by the ChildReqts relationship. If a template contains an OPEN command to a specific project, you will also see the ChildRequirements

			option, which lets you go directly to the Requirements.
ChildReqts	0..n	Requirement	Child (nested) requirements for a given Requirement.
Discussion	0..n	Discussion	Discussions attached to this Requirement.
Document	0..1	ReqDocument	Document where the Requirement is stored.
LatestRevision	0..1	Revision	Latest Revision artifact for a given Requirement.
ParentProject	0..1	Project	Project that this Requirement is contained in. This relationship is especially useful when doing cross-project traceability.
ParentReqt		Requirement	Parent Requirement artifact for a given Requirement.
RelativeParent	0..1	Relationship	Parent relationship for a given Requirement. The relationship refers to the parent Requirement artifact returned by the ParentReqt relationship.
ReqType	0..1	RequirementType	The type of this requirement.
Revisions	0..n	Revision	The collection of revision artifacts for a given Requirement.
TracesFrom	0..n	Relationship	Relationships traced out of this Requirement.
TracesTo	0..n	Relationship	Relationships traced in to this Requirement.
VisibleAttrValues	0..n	AttributeValue	Visible AttributeValue artifacts for a given Requirement. VisibleAttrValues does not include hidden attributes. Attributes are hidden or made visible in RequisitePro by selecting the attribute name and editing the "Hidden from display" button.

RequirementType (ReqPro Adapter)

A requirement type defines a set of similar requirements. Requirement types are used to classify similar requirements so they can be efficiently managed. When you define a requirement type, you define a common set of attributes, display style, and tag numbering.
Note: In order for the TeamTest adapter to return user-defined ReqPro requirement types, the word "Requirement" must be included in the new artifact type name.

Class Hierarchy: Artifact>RequirementType

SubClasses of RequirementType

RequirementType has no subclasses.

Properties Specific to RequirementType

Properties	Inherited From	Description
Description		A general description of the requirement type.
Name		Name of the requirement type.
ReqPrefix		Prefix of the type, such as SR.
TypeID		RequirementType ID.

Relationships Specific to RequirementType

Name	Kind	Class	Description
This class has no relationships.			

Response (ReqPro Adapter)

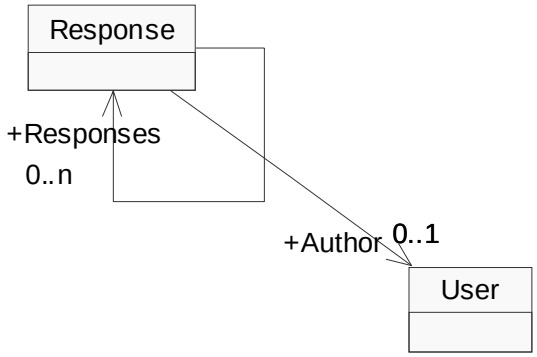
An answer to a discussion item. You can respond to the initial discussion topic or to other, previous responses.

In Rational RequisitePro, the terms "response" and "discussion response" are used interchangeably.

Class Hierarchy: Artifact>Response

SubClasses of Response

Response has no subclasses.



Properties Specific to Response

Properties	Inherited From	Description
DateTime		When the reply was created.
HasResponses		True if someone has replied to this reply.
Message		Text of the reply.
ResponseFullKey		
ResponseID		Response ID.
Subject		Subject of the reply.

Relationships Specific to Response

Name	Kind	Class	Description
Author	0..1	User	User who created this reply.
Responses	0..n	Response	Responses to this reply.

Revision (ReqPro Adapter)

A distinct version of a project, document, or requirement. A revision is identified by a unique internal revision number, generated by Rational RequisitePro. The Revision object lets you document revision information about a requirement, document, or project.

Class Hierarchy: Artifact>Revision

SubClasses of Revision

Revision has no subclasses.

Properties Specific to Revision

Properties	Inherited From	Description
DateTime		Date and time the requirement was created or modified; the correct format for date and time is: yyyy-mm-dd hh:mm:ss. The value "hh" is the two-digit hour in military time. Hyphens and colons must be included as shown. For example, 1999-04-24 20:23:12 means April 24, 1999 at 8:23pm plus 12 seconds. You can drop any trailing part of a date-time, for example >= 1999-04 may be specified in a filter to obtain all Replies marked April 1999 or later.
Description		Change description field.
Label		Text associated with a revision number.
Number		The revision number, incremented automatically for each revision.
ParentID		Revision parent type ID.
ParentType		Revision parent type.
VersionID		Version ID for the Revision.

Relationships Specific to Revision

Name	Kind	Class	Description
Author	0..1	User	User that made the change.

User (ReqPro Adapter)

A Rational RequisitePro user. Users are people who have access to project information. Each user that is associated with a license of RequisitePro is a licensed RequisitePro user. RequisitePro tracks which users make changes to project and requirement information.

Note: In RequisitePro, the terms "user" and "licensed user" are used interchangeably.

Class Hierarchy: Artifact>User

SubClasses of User

User has no subclasses.

Properties Specific to User

Properties	Inherited From	Description
FullName		Full name of the User.
Name		Name for this user.
UserID		UserID for this user.

Relationships Specific to User

Name	Kind	Class	Description
Group	0..1	Group	Group this user belongs to.

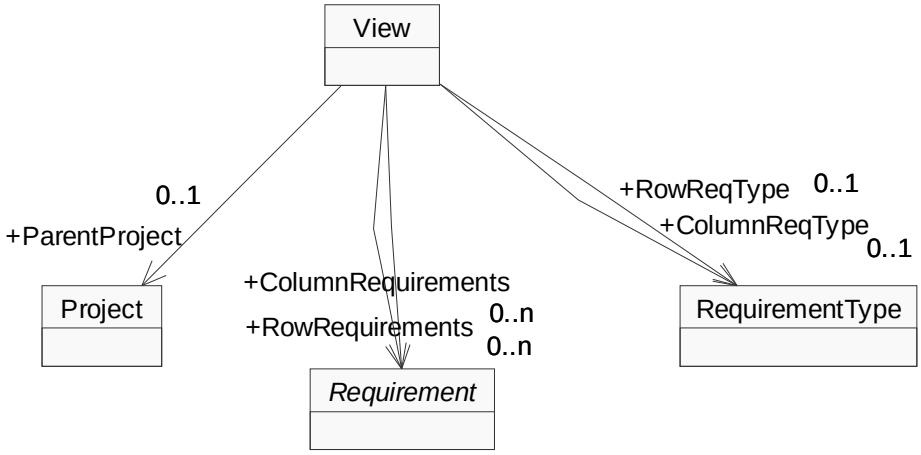
View (ReqPro Adapter)

A view displays information in spreadsheet-like tables or in outline trees. A view window displays requirements, the attributes assigned to requirements, or the relationships between requirements.
Note: In Rational RequisitePro, the terms "view" and "view window" are used interchangeably.

Class Hierarchy: Artifact>View

SubClasses of View

View has no subclasses.



Properties Specific to View

Properties	Inherited From	Description
Description		View description.
FullName		Full name of the View.
SimpleName		Simple name of the View.
State		State of the View.
Type		Type of View.
ViewID		The ID of the View.
Visibility		A string representing the visibility of the view. This can have the value Empty (no visibility), ProjectWide, or Personal.

Relationships Specific to View

Name	Kind	Class	Description
ColumnReqType	0..1	RequirementType	Requirement type of a column in a View.
ColumnRequirements	0..n	Requirement	Requirements in the columns of a View.
ParentProject	0..1	Project	Parent project associated with this View.
RowReqType	0..1	RequirementType	Requirement type of a row in a View.
RowRequirements	0..n	Requirement	Requirements in the rows of a View.

Rose

Rational Rose

The following Classes are available through the Rose RSE adapter:

- Action
- Activity
- Association
- Attribute
- Class
- ClassDiagram
- ClassUtility
- Decision
- DeploymentDiagram
- Device
- Diagram
- ExternalDocument
- HasRelationship
- InheritRelationship
- InstantiatedClass
- InstantiatedClassUtility
- Item
- Link
- Message
- MetaClass
- Model
- Module
- ModuleDiagram
- ModuleVisibilityRelationship
- Node
- Note
- ObjectFlow
- ObjectInstance
- Operation
- Package
- PackageDependency
- Parameter
- ParameterizedClass
- ParameterizedClassUtility
- Process

Processor
Property
RealizeRelationship
Relationship
Role
Scenario
State
StateDiagram
StateMachine
StateTransition
Subsystem
SyncItem
UseCase
UseCaseDiagram
UsesRelationship

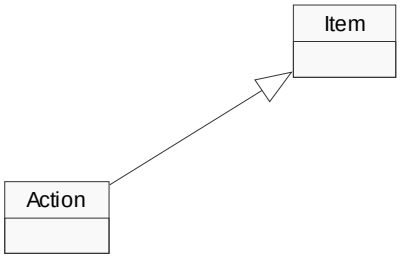
Action (Rose Adapter)

An action is an operation that is associated with a state transition.

Class Hierarchy: Artifact>Item>Action

SubClasses of Action

Action has no subclasses.



Properties Specific to Action

Properties	Inherited From	Description
Arguments		The arguments that accompany the trigger event.
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
Target		The name of the event object.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Action

Name	Kind	Class	Description
This class has no relationships.			

Activity (Rose Adapter)

The Activity class is an abstract class that exposes activity functionality in the Rose extensibility interface. With the Rose Activity class, you can:

· Retrieve information about activities, such as name, documentation, stereotype.

· Retrieve objects associated with activities such as parent activities, parent states, parent state machines, child activities, child decisions, child states, child synchronizations, outgoing transitions, and swimlanes.

· Create and retrieve tool and property settings for activities.

· Open specification sheets for activities.

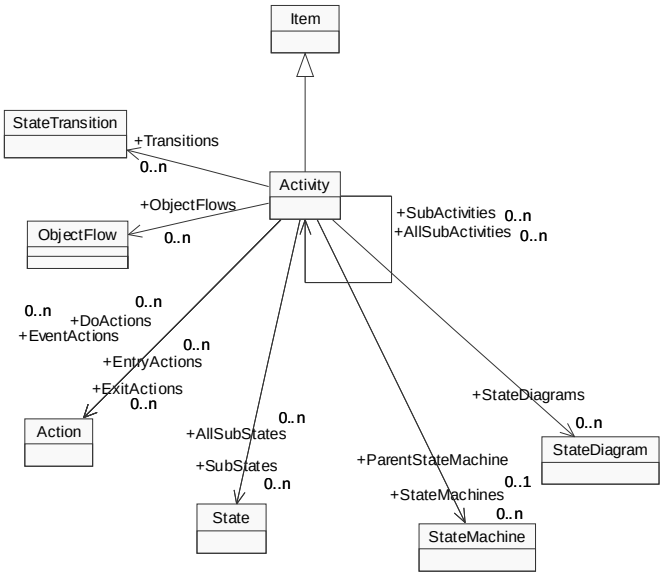
· Add, delete, and retrieve an activity's actions, state machines, and events.

· Add and delete transitions.

Class Hierarchy: Artifact>Item>Activity

SubClasses of Activity

Activity has no subclasses.



Properties Specific to Activity

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.

Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Activity

Name	Kind	Class	Description
AllSubActivities		Activity	All activities associated with this Activity.
AllSubStates	0..n	State	States associated with this Activity.
DoActions	0..n	Action	The Do actions for this Activity.
EntryActions	0..n	Action	The Entry actions for this Activity.
EventActions	0..n	Action	The Event actions for this Activity.
ExitActions	0..n	Action	The Exit actions for this Activity.
ObjectFlows	0..n	ObjectFlow	The associated object flows for this Activity.
ParentStateMachine	0..1	StateMachine	Parent state machine associated with this Activity.
StateDiagrams	0..n	StateDiagram	State or activity diagrams internal to this Activity.
StateMachines	0..n	StateMachine	State machines internal to this Activity.
SubActivities		Activity	Activities that are part of this Activity.
SubStates	0..n	State	States that are part of this Activity.
Transitions	0..n	StateTransition	Transitions that exit from this Activity.

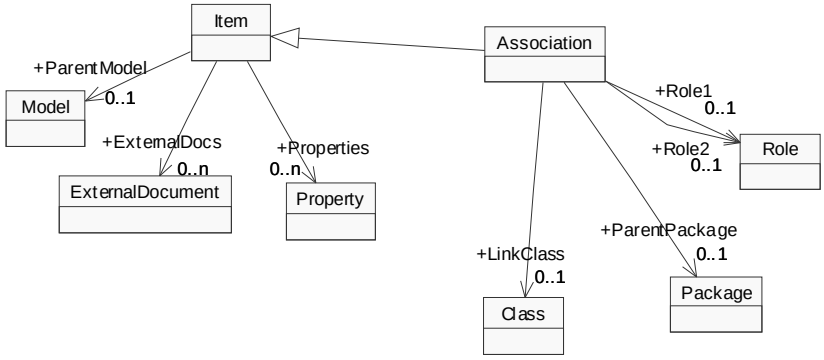
Association (Rose Adapter)

An association provides a pathway for communication. The communication can be between use cases, actors, classes, or interfaces. Associations are the most general of all relationships and consequentially the most semantically weak. If two objects are usually considered independently, the relationship is an association.

Class Hierarchy: Artifact>Item>Association

SubClasses of Association

Association has no subclasses.



Properties Specific to Association

Properties	Inherited From	Description
Constraints		The text from the Constraints field in the association specification.
Derived		True if the Association is derived; otherwise False.
Documentation	Item	Documentation for the item.
HasLinkClass		True if the Association has an attached association class, otherwise False.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Association

Name	Kind	Class	Description
LinkClass	0..1	Class	The linked class attached to the Association.
ParentPackage	0..1	Package	Parent package attached to the Association.
Role1	0..1	Role	The first role defined in the Association.
Role2	0..1	Role	The second role defined in the Association.

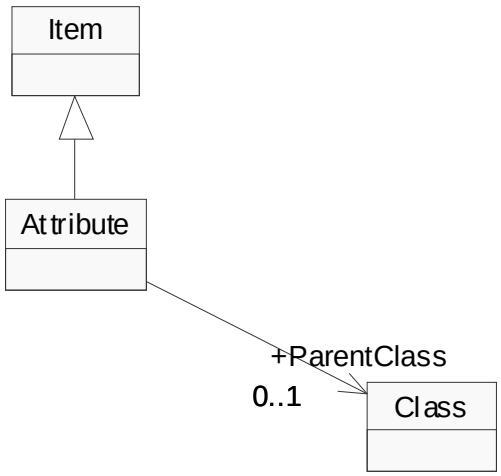
Attribute (Rose Adapter)

Attributes are data members of a class whose type is not another class. Attributes define the characteristics of a class. Each object in a class has the same attributes, but the values of the attributes may be different.

Class Hierarchy: Artifact>Item>Attribute

SubClasses of Attribute

Attribute has no subclasses.



Properties Specific to Attribute

Properties	Inherited From	Description
Containment		Specifies the physical containment of the attribute. Returns Value, Reference, or Unspecified, depending on the state of the Containment radio control on the attribute specification.
Derived		True if the Derived check box is selected in the attribute specification, otherwise False.
Documentation	Item	Documentation for the item.

ExportControl		The export control of the attribute. Returns Public, Protected, Private, or implementation.
InitValue		The initial value of the Attribute.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Static		True if the Static check box is selected in the attribute specification, otherwise False.
Stereotype	Item	Stereotype of the item.
Type		The type of the Attribute.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Attribute

Name	Kind	Class	Description
ParentClass	0..1	Class	The class in which this attribute is defined.

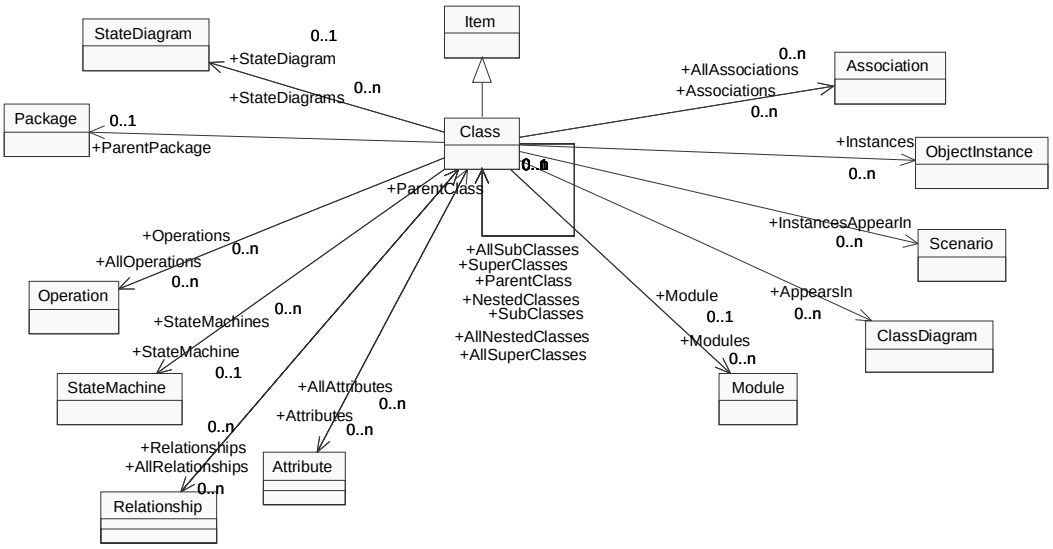
Class (Rose Adapter)

A class captures the common structure and common behavior of a set of objects. A class is an abstraction of real-world items. When these items exist in the real world, they are instances of the class, and referred to as objects. Rational Rose stores class information in a class specification.

Class Hierarchy: Artifact>Item>Class

SubClasses of Class

ClassUtility
InstantiatedClass
InstantiatedClassUtility
MetaClass
ParameterizedClass
ParameterizedClassUtility



Properties Specific to Class

Properties	Inherited From	Description
Abstract		True if the Abstract check box is selected in the class specification, otherwise False.

Cardinality		The string in the Cardinality field of the class specification.
Concurrency		Returns Sequential, Guarded, Active, or Synchronous, depending on the value of the Concurrency radio control in the More dialog of the class specification.
Documentation	Item	Documentation for the item.
ExportControl		Returns Public or Implementation, depending on the value of the Export Control radio control in the class specification.
FundamentalType		Returns True if this Class is a fundamental type.
HasStateDiagram		Returns True if the Class has an associated state diagram, otherwise False.
IsNested		Returns True if the Class is nested, otherwise False.
Kind		The kind of Class.
Name	Item	Name of the item.
Persistence		This property is Persistent or Transient, depending on the value of the Persistence radio control in the More dialog of the class specification.
QualifiedName	Item	Qualified name of the item.
Space		The string in the Space field of the More dialog of the class specification.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Class

Name	Kind	Class	Description
AllAssociations	0..n	Association	All associations where this Class plays a role, including those inherited from other classes.
AllAttributes	0..n	Attribute	All attributes of this Class, including those inherited from other classes.
AllNestedClasses	0..n	Class	All nested classes of this Class.
AllOperations	0..n	Operation	All operations of this Class, including those inherited from other classes.
AllRelationships	0..n	Relationship	All relationships of this Class, including those inherited from other classes.
AllSubClasses	0..n	Class	All classes in the lineage of this Class. For example, if A inherits from B and B inherits from C, then AllSubClasses of C would include B and A.
AllSuperClasses	0..n	Class	All classes in the ancestry of this Class. For example, if A inherits from B and B inherits from C, then AllSuperClasses of A would include B and C.
AppearsIn	0..n	ClassDiagram	The class diagrams where this Class appears.
Associations	0..n	Association	The associations where this Class plays a role.
Attributes	0..n	Attribute	Attributes that are defined by this Class. Does not include inherited attributes.
Instances	0..n	ObjectInstance	Object instances associated with this Class.
InstancesAppearIn	0..n	Scenario	Interaction diagrams that include instances of this Class.
Module	0..1	Module	The first module associated with this Class.
Modules	0..n	Module	All modules associated with this Class.
NestedClasses	0..n	Class	Classes that are nested within this Class.

Operations	0..n	Operation	Operations that are defined by this Class. Does not include inherited operations.
ParentClass	0..1	Class	Parent class of this Class, if it is nested.
ParentPackage	0..1	Package	The enclosing package.
Relationships	0..n	Relationship	Relationships that are defined by this Class. Does not include inherited relationships.
StateDiagram	0..1	StateDiagram	The (first) state/activity diagram associated with this Class.
StateDiagrams	0..n	StateDiagram	All state/activity diagrams associated with this Class.
StateMachine	0..1	StateMachine	The (first) state machine associated with this Class.
StateMachines	0..n	StateMachine	All state machine associated with this Class.
SubClasses	0..n	Class	The classes that directly inherit from this Class. Only includes immediate subclasses. For example, if A inherits from B and B inherits from C, then MySubClasses of C would include B but not A.
SuperClasses	0..n	Class	The classes that this Class directly inherits from. Only includes immediate superclasses. For example, if A inherits from B and B inherits from C, then MySuperClasses of A would include B but not C.

ClassDiagram (Rose Adapter)

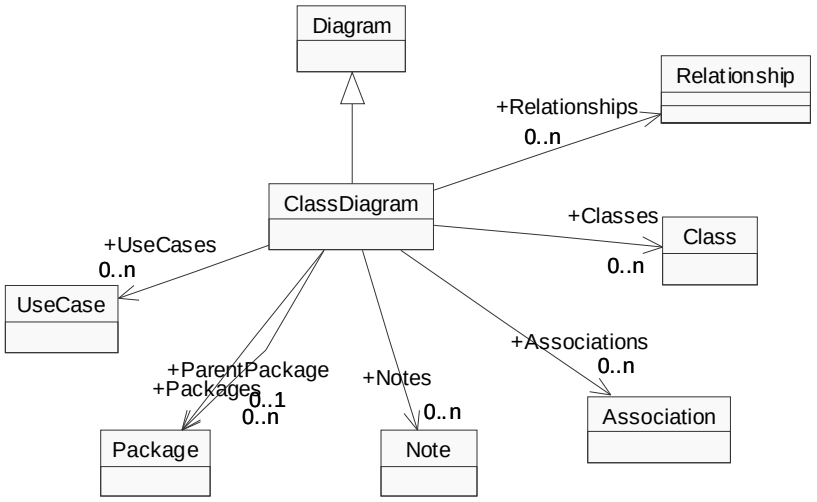
A class diagram shows the relationships between packages and classes; the essential relationships include association, inherits, has, and uses. Each class diagram provides a logical view of the current model.

Class diagrams contain icons representing packages and classes. Class diagrams can be considered as filtered views into the model. They do not necessarily depict all the classes or relationships in the model. For example, iterating over all the classes in the main diagram of a package will not necessarily return all the classes defined in that category.

Class Hierarchy: Artifact>Diagram>ClassDiagram

SubClasses of ClassDiagram

ClassDiagram has no subclasses.



Properties Specific to ClassDiagram

Properties	Inherited From	Description
Documentation	Diagram	The documentation text associated with the Diagram.
MappedPoints	Diagram	A list of coordinates of the items in the Diagram. Each item is specified by a set

		of x/y coordinates designating the location of the corners of the item. The ordering of the items is the same as in the MappedAritifacts artifact collection.
Name	Diagram	Name of the Diagram.
QualifiedName	Diagram	Qualified name of the Diagram.
UniqueID	Diagram	The unique ID for the Diagram.

Relationships Specific to ClassDiagram

Name	Kind	Class	Description
Associations	0..n	Association	The associations where this class diagram plays a role.
Classes	0..n	Class	All of the classes that appear on the diagram.
Notes	0..n	Note	Notes that appear in the diagram.
Packages	0..n	Package	All packages associated with this class diagram.
ParentPackage	0..1	Package	The package that this diagram is contained in, if applicable.
Relationships	0..n	Relationship	All of the relationships that appear on the diagram.
UseCases	0..n	UseCase	All of the use cases that appear on the diagram.

ClassUtility (Rose Adapter)

A class utility is a set of operations that provide additional functions for classes. Class utilities are used to:
· Denote one or more free subprograms.
· Name a class that only provides static members and/or static member functions.

Class Hierarchy: Item>Class>ClassUtility

SubClasses of ClassUtility

ClassUtility has no subclasses.

Properties Specific to ClassUtility

Properties	Inherited From	Description
Abstract	Class	True if the Abstract check box is selected in the class specification, otherwise False.
Cardinality	Class	The string in the Cardinality field of the class specification.
Concurrency	Class	Returns Sequential, Guarded, Active, or Synchronous, depending on the value of the Concurrency radio control in the More dialog of the class specification.
Documentation	Item	Documentation for the item.
ExportControl	Class	Returns Public or Implementation, depending on the value of the Export Control radio control in the class specification.
FundamentalType	Class	Returns True if this class is a fundamental type.
HasStateDiagram	Class	Returns True if the class has an associated state diagram, otherwise False.
IsNested	Class	Returns True if the Class is nested, otherwise False.
Kind	Class	The kind of Class.
Name	Item	Name of the item.

Persistence	Class	This property is Persistent or Transient, depending on the value of the Persistence radio control in the More dialog of the class specification.
QualifiedName	Item	Qualified name of the item.
Space	Class	The string in the Space field of the More dialog of the class specification.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to ClassUtility

Name	Kind	Class	Description
This class has no relationships.			

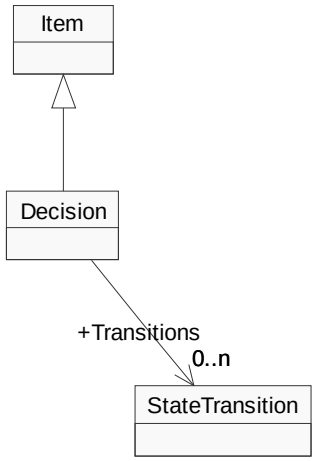
Decision (Rose Adapter)

The Decision class is an abstract class that exposes decision functionality in the Rose extensibility interface.

Class Hierarchy: Artifact>Item>Decision

SubClasses of Decision

Decision has no subclasses.



Properties Specific to Decision

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Decision

Name	Kind	Class	Description
Transitions	0..n	StateTransition	State transition for this Decision.

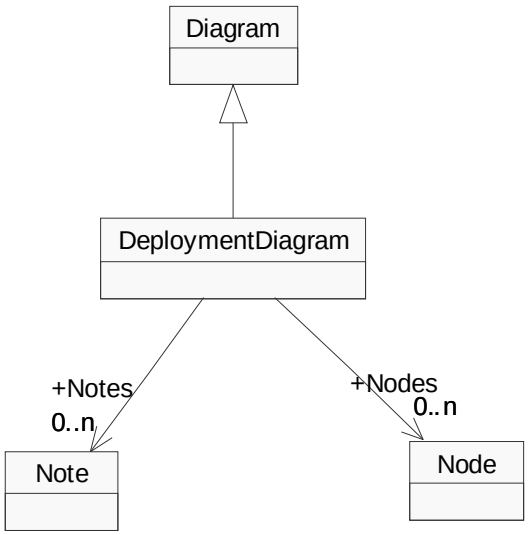
DeploymentDiagram (Rose Adapter)

A deployment diagram shows the allocation of processes to processors in the physical design of a system. A deployment diagram may represent all or part of the process architecture of a system.

Class Hierarchy: Artifact>Diagram>DeploymentDiagram

SubClasses of DeploymentDiagram

DeploymentDiagram has no subclasses.



Properties Specific to DeploymentDiagram

Properties	Inherited From	Description
Documentation	Diagram	The documentation text associated with the Diagram.
MappedPoints	Diagram	A list of coordinates of the items in the Diagram. Each item is specified by a set of x/y coordinates designating the location of the corners of the item. The ordering of the items is the same as in the MappedAritifacts artifact collection.

Name	Diagram	Name of the Diagram.
QualifiedName	Diagram	Qualified name of the Diagram.
UniqueID	Diagram	The unique ID for the Diagram.

Relationships Specific to DeploymentDiagram

Name	Kind	Class	Description
Nodes	0..n	Node	Processors and devices contained in the diagram.
Notes	0..n	Note	Notes that appear in the diagram.

Device (Rose Adapter)

A device is a hardware component with no computing power. The Rose device class exposes properties and methods that allow you to define and manipulate the characteristics of devices.

Class Hierarchy: Item>Node>Device

SubClasses of Device

Device has no subclasses.

Properties Specific to Device

Properties	Inherited From	Description
Characteristics	Node	Characteristics of the processor or device.
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Device

Name	Kind	Class	Description
This class has no relationships.			

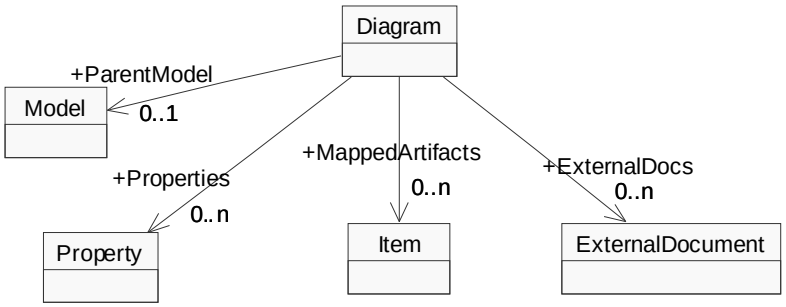
Diagram (Rose Adapter)

Exposes a set of properties and methods, which all other diagram classes (for example, class diagrams, and scenario diagrams) inherit. These properties and methods determine the size and placement of a diagram on the Rose user's computer screen.

Class Hierarchy: Artifact>Diagram

SubClasses of Diagram

ClassDiagram
DeploymentDiagram
ModuleDiagram
Scenario
StateDiagram
UseCaseDiagram



Properties Specific to Diagram

Properties	Inherited From	Description
Documentation		The documentation text associated with the Diagram.
MappedPoints		A list of coordinates of the items in the Diagram. Each item is specified by a set of x/y coordinates designating the location of the corners of the item. The ordering of the items is the same as in the MappedAritifacts artifact collection.
Name		Name of the Diagram.
QualifiedName		Qualified name of the Diagram.

UniqueID	The unique ID for the Diagram.

Relationships Specific to Diagram

Name	Kind	Class	Description
ExternalDocs	0..n	ExternalDocument	External documents attached to this Diagram.
MappedArtifacts	0..n	Item	Items that are associated with this Diagram.
ParentModel	0..1	Model	Model that this Diagram is contained in.
Properties	0..n	Property	Property artifact types associated with this Diagram.

ExternalDocument (Rose Adapter)

Exposes properties and methods that allow you to create external documents (reports) from within the Rose environment. For example, you can start Word for Windows and output information from a Rose model into a Word document.

Class Hierarchy: Artifact>ExternalDocument

SubClasses of ExternalDocument

ExternalDocument has no subclasses.

Properties Specific to ExternalDocument

Properties	Inherited From	Description
CollIndex		The index of the ExternalDocument within the collection of documents that contains it. It is used internally to identify the document.
ParentUID		The unique ID of the external document's parent class
Value		The actual path of the document.

Relationships Specific to ExternalDocument

Name	Kind	Class	Description
This class has no relationships.			

HasRelationship (Rose Adapter)

The Has Relationship indicates a containment or aggregation relationship between classes. The has relationship, available only with the Booch notation, denotes a whole and part relationship between two classes. This relationship is used to show how instances of the supplier, or aggregate, class are physically constructed from instances of the client class. The FromClass relationship returns the aggregate class. The ToClass relationship returns the client class, whose instances are part of aggregate class instances.

Class Hierarchy: Item>Relationship>HasRelationship

SubClasses of HasRelationship

HasRelationship has no subclasses.

Properties Specific to HasRelationship

Properties	Inherited From	Description
ClientCardinality	Relationship	Indicates the number of possible links from an instance of the client class to an instance of the supplier class. Can be the same values as those listed in SupplierCardinality.
Containment		Specifies the physical containment of the relationship. Returns Value, Reference, or Unspecified, depending on the state of the Containment radio control on the relationship specification. Containment is also shown by adornments on relationships in diagrams.
Documentation	Item	Documentation for the item.
ExportControl	Relationship	Specifies the type of access allowed between classes. Returns Public, Protected, Private, or Implementation, depending on the state of the Access radio control on the relationship specification. Access is also shown by adornments on relationships in diagrams.
Kind	Relationship	Kind of the relationship, which will be one of: AggregateRole, AssociationRole,

		HasRelationship, InheritsRelationship, or UsesRelationship.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Static		Specifies whether the instance of the part class is owned by the class itself and not by its individual instances. Returns True, if the Static check box is checked on the relationship specification. Otherwise, returns False. Static relationships are also designated by special adornments on relationships in diagrams.
Stereotype	Item	Stereotype of the item.
SupplierCardinality	Relationship	Indicates the number of possible links from an instance of the supplier class to an instance of the client class. Can be one the following values: n, 1, 0..n, 1..n, 0..1, <literal>, <literal>..n, or <literal>..<literal>.
SupplierName	Relationship	Name of the supplier class or use case.
UniqueID	Item	The unique ID of the item.

Relationships Specific to HasRelationship

Name	Kind	Class	Description
This class has no relationships.			

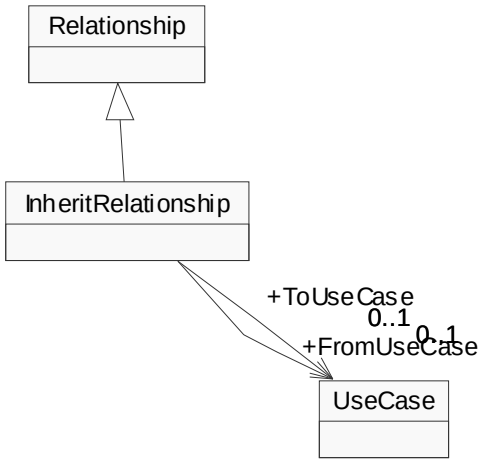
InheritRelationship (Rose Adapter)

Indicates an inheritance relationship between classes.

Class Hierarchy: Item>Relationship>InheritRelationship

SubClasses of InheritRelationship

InheritRelationship has no subclasses.



Properties Specific to InheritRelationship

Properties	Inherited From	Description
ClientCardinality	Relationship	Indicates the number of possible links from an instance of the client class to an instance of the supplier class. Can be the same values as those listed in SupplierCardinality.
Documentation	Item	Documentation for the item.
ExportControl	Relationship	Specifies the type of access allowed between classes. Returns Public, Protected, Private, or Implementation, depending on the state of the Access radio control on the relationship specification. Access is also shown by

		adornments on relationships in diagrams.
FriendshipRequired		Indicates whether the supplier class grants rights to the client class to access its nonpublic parts. Returns True, if the Friendship required check box is checked on the relationship specification. Otherwise, returns False.
Kind	Relationship	Kind of the relationship, which will be one of: AggregateRole, AssociationRole, HasRelationship, InheritsRelationship, or UsesRelationship.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
SupplierCardinality	Relationship	Indicates the number of possible links from an instance of the supplier class to an instance of the client class. Can be one the following values: n, 1, 0..n, 1..n, 0..1, <literal>, <literal>..n, or <literal>..<literal>.
SupplierName	Relationship	Name of the supplier class or use case.
UniqueID	Item	The unique ID of the item.
Virtual		Boolean value indicating whether the relation is virtual.

Relationships Specific to InheritRelationship

Name	Kind	Class	Description
FromUseCase	0..1	UseCase	The supplier use case of the inherits relationship, if it is a use case.
ToUseCase	0..1	UseCase	The client use case of the inherits relationship, if it is a use case.

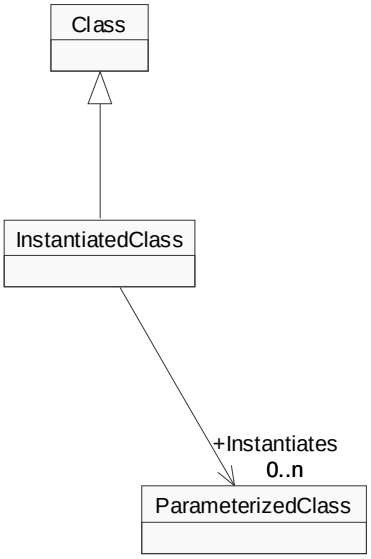
InstantiatedClass (Rose Adapter)

A class which instantiates a parameterized class. Instantiated classes are created by supplying the actual values for the formal parameters of the parameterized class. An instantiated class is concrete, meaning that its implementation is complete, and it may have object instances.

Class Hierarchy: Item>Class>InstantiatedClass

SubClasses of InstantiatedClass

InstantiatedClass has no subclasses.



Properties Specific to InstantiatedClass

Properties	Inherited From	Description
Abstract	Class	True if the Abstract check box is selected in the class specification, otherwise False.
Cardinality	Class	The string in the Cardinality field of the class specification.
Concurrency	Class	Returns Sequential, Guarded, Active, or Synchronous, depending on the value of

		the Concurrency radio control in the More dialog of the class specification.
Documentation	Item	Documentation for the item.
ExportControl	Class	Returns Public or Implementation, depending on the value of the Export Control radio control in the class specification.
FundamentalType	Class	Returns True if this class is a fundamental type.
HasStateDiagram	Class	Returns True if the class has an associated state diagram, otherwise False.
IsNested	Class	Returns True if the Class is nested, otherwise False.
Kind	Class	The kind of Class.
Name	Item	Name of the item.
Persistence	Class	This property is Persistent or Transient, depending on the value of the Persistence radio control in the More dialog of the class specification.
QualifiedName	Item	Qualified name of the item.
Space	Class	The string in the Space field of the More dialog of the class specification.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to InstantiatedClass

Name	Kind	Class	Description
Instantiates	0..n	ParameterizedClass	The parameterized class that this instantiated class instantiates.

InstantiatedClassUtility (Rose Adapter)

A class utility which instantiates a parameterized class utility. Instantiated class utilities are created by supplying the actual values for the formal parameters of the parameterized class utility.

An instantiated class utility is displayed as a 3-part box, with the class name in the top part, a list of attributes (with optional types and values) in the middle part, and a list of operations (with optional argument lists and return types) in the bottom part.

Class Hierarchy: Item>Class>InstantiatedClassUtility

SubClasses of InstantiatedClassUtility

InstantiatedClassUtility has no subclasses.

Properties Specific to InstantiatedClassUtility

Properties	Inherited From	Description
Abstract	Class	True if the Abstract check box is selected in the class specification, otherwise False.
Cardinality	Class	The string in the Cardinality field of the class specification.
Concurrency	Class	Returns Sequential, Guarded, Active, or Synchronous, depending on the value of the Concurrency radio control in the More dialog of the class specification.
Documentation	Item	Documentation for the item.
ExportControl	Class	Returns Public or Implementation, depending on the value of the Export Control radio control in the class specification.
FundamentalType	Class	Returns True if this class is a fundamental type.
HasStateDiagram	Class	Returns True if the class has an associated state diagram, otherwise False.
IsNested	Class	Returns True if the Class is nested, otherwise False.

Kind	Class	The kind of Class.
Name	Item	Name of the item.
Persistence	Class	This property is Persistent or Transient, depending on the value of the Persistence radio control in the More dialog of the class specification.
QualifiedName	Item	Qualified name of the item.
Space	Class	The string in the Space field of the More dialog of the class specification.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to InstantiatedClassUtility

Name	Kind	Class	Description
Instantiates	0..n	ParameterizedClass	The parameterized class utility that this instantiated class utility instantiates.

Item (Rose Adapter)

Item maps to RoseItem objects. Every RoseItem is a model element and therefore inherits all Element properties and methods. Item specifies the type of model element that the stereotype settings apply to. Valid items include:

Class
Component
Package (includes logical package, use case package, and component package)
Logical Package
Component Package
Use Case Package
Processor
Device
Use Case
Association
Generalization
Dependency
Connection
Class Attribute
Operation

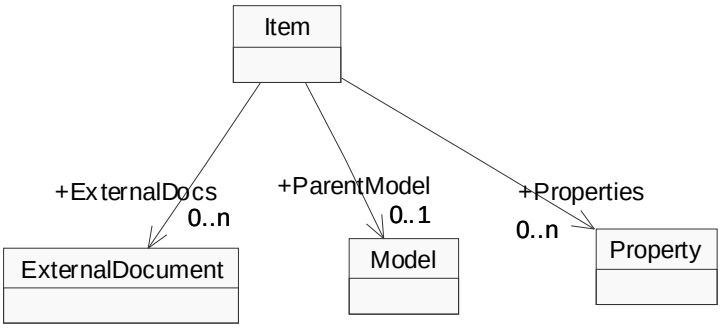
The default setting is Class.

Class Hierarchy: Artifact>Item

SubClasses of Item

Action
Activity
Association
Attribute
Class
Decision
Link
Message
Model
Module
ModuleVisibilityRelationship
Node
ObjectInstance
Operation
Package

PackageDependency
Parameter
Process
Relationship
State
StateTransition
Subsystem
SyncItem
UseCase



Properties Specific to Item

Properties	Inherited From	Description
Documentation		Documentation for the item.
Name		Name of the item.
QualifiedName		Qualified name of the item.
Stereotype		Stereotype of the item.
UniqueID		The unique ID of the item.

Relationships Specific to Item

Name	Kind	Class	Description
ExternalDocs	0..n	ExternalDocument	ExternalDocuments associated with this Item.
ParentModel	0..1	Model	Parent Model associated with this Item.
Properties	0..n	Property	The Property artifact types associated with this Item.

Link (Rose Adapter)

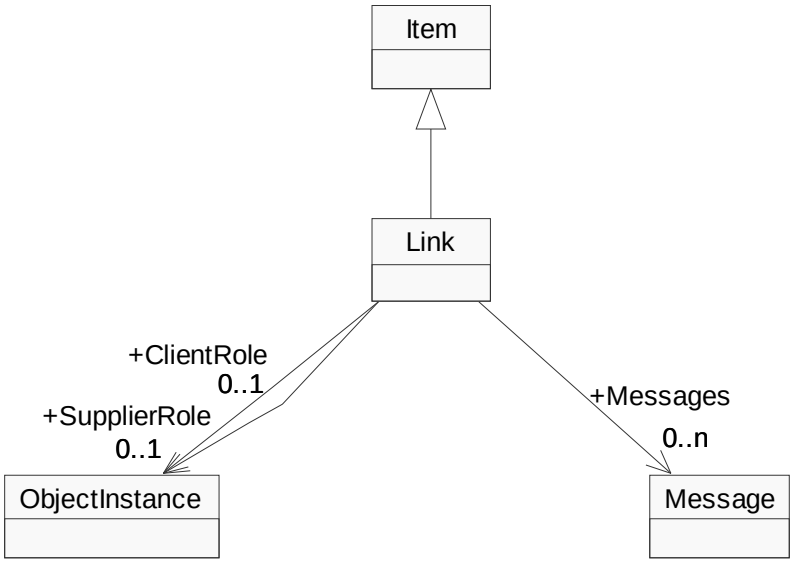
Objects interact through their links to other objects. A Link is an instance of an association, in the same way that an object is an instance of a class.

Rose Link properties and methods allow you to define links between objects and determine the nature of the objects' associations.

Class Hierarchy: Artifact>Item>Link

SubClasses of Link

Link has no subclasses.



Properties Specific to Link

Properties	Inherited From	Description
ClientIsShared		True if the Shared box is checked on the client side; otherwise False.
ClientVisibility		One of Unspecified, Field, Parameters, Local, or Global.

Documentation	Item	Documentation for the item.
IsLinkToSelf		True if the link goes from an object to itself.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
SupplierIsShared		True if the Shared box is checked on the supplier side; otherwise False.
SupplierVisibility		One of Unspecified, Field, Parameters, Local, or Global.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Link

Name	Kind	Class	Description
ClientRole	0..1	ObjectInstance	The client object instance (role) of the Link.
Messages	0..n	Message	Messages associated with the Link.
SupplierRole	0..1	ObjectInstance	The supplier object instance (role) of the Link.

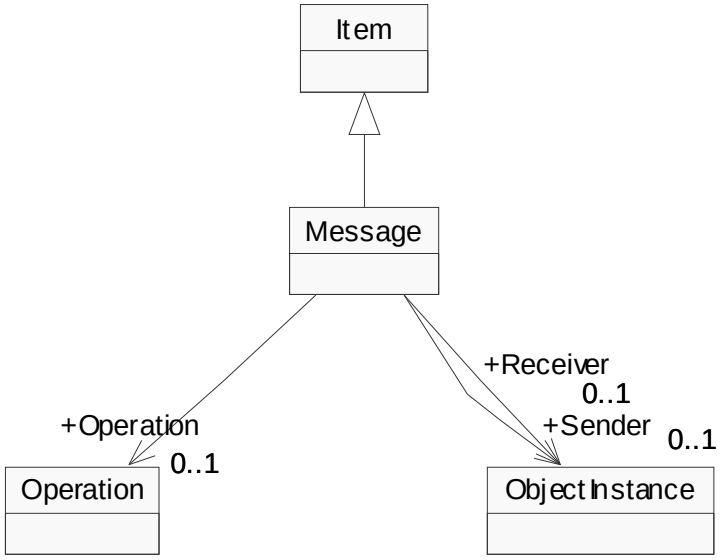
Message (Rose Adapter)

Any message associated with an object. Messages define the interaction between objects. The Rose message class inherits all of the Item (RoseItem) properties and methods. In addition message class methods allow you to retrieve message sender and receiver, along with other message-specific information.

Class Hierarchy: Artifact>Item>Message

SubClasses of Message

Message has no subclasses.



Properties Specific to Message

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
Frequency		Frequency of the message.
HierarchicalSeqNumber		Hierarchical sequence number of the message.
IsOperation		Boolean value indicating whether the message is associated with an operation.

Name	Item	Name of the item.
NameWithoutParentheses		Name of a message without the parameters enclosed in parentheses from Rose that are added when a message is associated with a class operation.
QualifiedName	Item	Qualified name of the item.
SeqNumber		Sequence number of the message.
Stereotype	Item	Stereotype of the item.
Synchronization		Concurrency semantics for the operation named in the Operations Field; one of Simple, Synchronous, Balking, Timeout or Asynchronous.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Message

Name	Kind	Class	Description
Operation	0..1	Operation	The associated Operation with this Message.
Receiver	0..1	ObjectInstance	Object that receives the Message.
Sender	0..1	ObjectInstance	Object that sends the Message.

MetaClass (Rose Adapter)

A metaclass is a class whose instances are classes rather than objects. Metaclasses provide operations for initializing class variables and serve as repositories to hold class variables where a single value is required by all objects of a class. Smalltalk and CLOS support the use of metaclasses. C++ does not directly support metaclasses.

A metaclass is displayed as a 3-part box, with the class name in the top part, a list of attributes (with optional types and values) in the middle part, and a list of operations (with optional argument lists and return types) in the bottom part.

Not all languages directly support metaclasses.

Class Hierarchy: Item>Class>MetaClass

SubClasses of MetaClass

MetaClass has no subclasses.

Properties Specific to MetaClass

Properties	Inherited From	Description
Abstract	Class	True if the Abstract check box is selected in the class specification, otherwise False.
Cardinality	Class	The string in the Cardinality field of the class specification.
Concurrency	Class	Returns Sequential, Guarded, Active, or Synchronous, depending on the value of the Concurrency radio control in the More dialog of the class specification.
Documentation	Item	Documentation for the item.
ExportControl	Class	Returns Public or Implementation, depending on the value of the Export Control radio control in the class specification.
FundamentalType	Class	Returns True if this class is a fundamental type.

HasStateDiagram	Class	Returns True if the class has an associated state diagram, otherwise False.
IsNested	Class	Returns True if the Class is nested, otherwise False.
Kind	Class	The kind of Class.
Name	Item	Name of the item.
Persistence	Class	This property is Persistent or Transient, depending on the value of the Persistence radio control in the More dialog of the class specification.
QualifiedName	Item	Qualified name of the item.
Space	Class	The string in the Space field of the More dialog of the class specification.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to MetaClass

Name	Kind	Class	Description
This class has no relationships.			

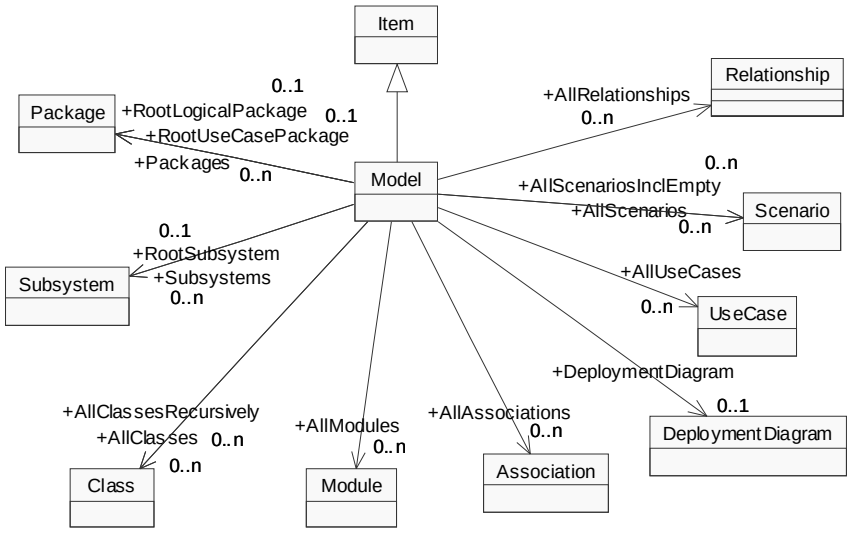
Model (Rose Adapter)

A Rose model file. A model file contains a Rose model, which describes your problem domain and system software. Model files use the default extension .mdl. Models are the highest hierarchical elements of the Rose source domain. Most templates start with connections to a Model.

Class Hierarchy: Artifact>Item>Model

SubClasses of Model

Model has no subclasses.



Properties Specific to Model

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
DriveLetter		Drive letter in the path of the Model.
Extension		The segment of a SimpleName following the last period. For example, the Extension of C:\bill\file.txt is txt. If the SimpleName contains no period, then Extension returns a null string.

FullName		The full name, including the path, of the Model.
Name	Item	Name of the item.
NameMinusExtension		Segment of a SimpleName preceding the last period. For example, the NameMinusExtension of C:\bill\file.txt is file. If the SimpleName contains no period, then NameMinusExtension returns the SimpleName.
NamePrefix		The segment of a SimpleName preceding the first period in the file name. For example, the NamePrefix of C:\bill\file.test.txt is file.
ParentDirectoryPath		Directory containing the object.
Path		The complete path of an object. For example, C:\bill\file.txt
QualifiedName	Item	Qualified name of the item.
SimpleName		The simple name of the Model. The context-independent portion of an object's name. For example, the SimpleName of C:\bill\file.txt is file.txt.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Model

Name	Kind	Class	Description
AllAssociations	0..n	Association	All associations in the Model.
AllClasses	0..n	Class	All classes in the Model, including actors.
AllClassesRecursively	0..n	Class	All classes in the model, including their nested classes recursively. Note that AllClasses does not return nested classes.
AllModules	0..n	Module	All modules in the Model (including subsystems).

AllRelationships	0..n	Relationship	All relationships in the Model.
AllScenarios	0..n	Scenario	All scenarios in the Model.
AllScenariosInclEmpty	0..n	Scenario	
AllUseCases	0..n	UseCase	All use cases in the Model.
DeploymentDiagram	0..1	DeploymentDiagram	The deployment diagram (process diagram) for the Model.
Packages	0..n	Package	All packages in the Model, including use case packages (but not including subsystems in the Component View).
RootLogicalPackage	0..1	Package	The highest-level package in the Model; its name is Logical View. All other packages are nested beneath it.
RootSubsystem	0..1	Subsystem	The highest-level subsystem in the Model; its name is Component View. All other subsystems are nested beneath it.
RootUseCasePackage	0..1	Package	The root use case package in the Model; its name is UseCase View. All other use-case packages are nested beneath it.
Subsystems	0..n	Subsystem	All subsystem components in the Model.

Module (Rose Adapter)

A module is a unit of code that serves as a building block for the physical structure of a system. The module class exposes properties and methods that allow you to define and manipulate the characteristics of modules.

Class Hierarchy: Artifact>Item>Module

SubClasses of Module

Module has no subclasses.

Properties Specific to Module

Properties	Inherited From	Description
AssignedLanguage		Specifies the programming language assigned to the Module.
Declarations		Text of the declarations belonging to the Module.
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
Part		Defines the Module as a part of a subsystem: a Specification, Body, Generic, or Main.
Path		The path of the Module.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
Type		The type of Module.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Module

Name	Kind	Class	Description
AssignedClasses	0..n	Class	Associated classes to this Module.
ParentSubsystem	0..1	Subsystem	Parent subsystem of this Module.
VisibilityRelationships	0..n	ModuleVisibilityRelationship	The module visibility relationships for this Module.

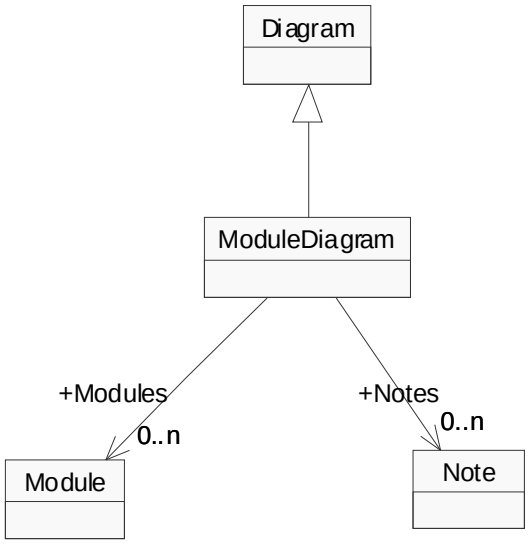
ModuleDiagram (Rose Adapter)

A module diagram maps the allocation classes and objects to modules. The module diagram class exposes properties and methods that allow you to add, retrieve, and delete classes and objects in a module diagram.

Class Hierarchy: Artifact>Diagram>ModuleDiagram

SubClasses of ModuleDiagram

ModuleDiagram has no subclasses.



Properties Specific to ModuleDiagram

Properties	Inherited From	Description
Documentation	Diagram	The documentation text associated with the Diagram.
MappedPoints	Diagram	A list of coordinates of the items in the Diagram. Each item is specified by a set of x/y coordinates designating the location of the corners of the item. The ordering of the items is the same as in the MappedAritifacts artifact collection.

Name	Diagram	Name of the Diagram.
QualifiedName	Diagram	Qualified name of the Diagram.
UniqueID	Diagram	The unique ID for the Diagram.

Relationships Specific to ModuleDiagram

Name	Kind	Class	Description
Modules	0..n	Module	Modules in the module diagram.
Notes	0..n	Note	Notes that appear in the module diagram.

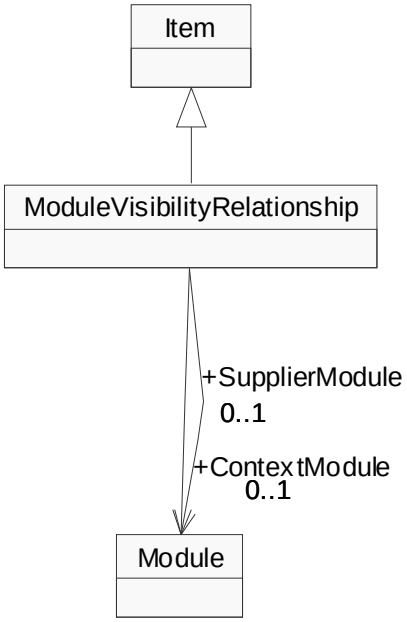
ModuleVisibilityRelationship (Rose Adapter)

The ModuleVisibilityRelationship class describes the context and supplier relationship between modules.

Class Hierarchy: Artifact>Item>ModuleVisibilityRelationship

SubClasses of ModuleVisibilityRelationship

ModuleVisibilityRelationship has no subclasses.



Properties Specific to ModuleVisibilityRelationship

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to ModuleVisibilityRelationship

Name	Kind	Class	Description
ContextModule	0..1	Module	The consumer module.
SupplierModule	0..1	Module	The supplier module.

Node (Rose Adapter)

Node is an abstract class for processors and devices.

Class Hierarchy: Artifact>Item>Node

SubClasses of Node

Device
Processor

Properties Specific to Node

Properties	Inherited From	Description
Characteristics		Characteristics of the processor or device.
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Node

Name	Kind	Class	Description
This class has no relationships.			

Note (Rose Adapter)

A note captures the assumptions and decisions applied during analysis and design. Notes may contain any information, including plain text, fragments of code, or references to other documents. Notes are also used as a means of linking diagrams. A note holds an unlimited amount of text and can be sized accordingly.

Class Hierarchy: Artifact>Note

SubClasses of Note

Note has no subclasses.

Properties Specific to Note

Properties	Inherited From	Description
CollIndex		
Text		Text of the Note.
Type		The type of Note.

Relationships Specific to Note

Name	Kind	Class	Description
This class has no relationships.			

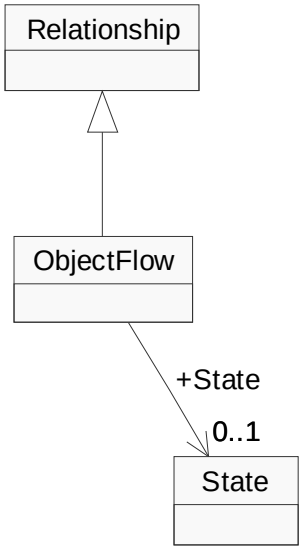
ObjectFlow (Rose Adapter)

The ObjectFlow class is an abstract class that exposes Rose's object flow functionality in the extensibility interface. An object flow on an activity diagram represents the relationship between an activity and the object that creates it (as an output) or uses it (as an input).

Class Hierarchy: Artifact>Item>Relationship>ObjectFlow

SubClasses of ObjectFlow

ObjectFlow has no subclasses.



Properties Specific to ObjectFlow

Properties	Inherited From	Description
ClientCardinality	Relationship	Indicates the number of possible links from an instance of the client class to an instance of the supplier class. Can be the same values as those listed in SupplierCardinality.
Documentation	Item	Documentation for the item.

ExportControl	Relationship	Specifies the type of access allowed between classes. Returns Public, Protected, Private, or Implementation, depending on the state of the Access radio control on the relationship specification. Access is also shown by adornments on relationships in diagrams.
Kind	Relationship	Kind of the relationship, which will be one of: AggregateRole, AssociationRole, HasRelationship, InheritsRelationship, or UsesRelationship.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
SupplierCardinality	Relationship	Indicates the number of possible links from an instance of the supplier class to an instance of the client class. Can be one the following values: n, 1, 0..n, 1..n, 0..1, <literal>, <literal>..n, or <literal>..<literal>.
SupplierName	Relationship	Name of the supplier class or use case.
UniqueID	Item	The unique ID of the item.

Relationships Specific to ObjectFlow

Name	Kind	Class	Description
State	0..1	State	Associated State of the ObjectFlow.

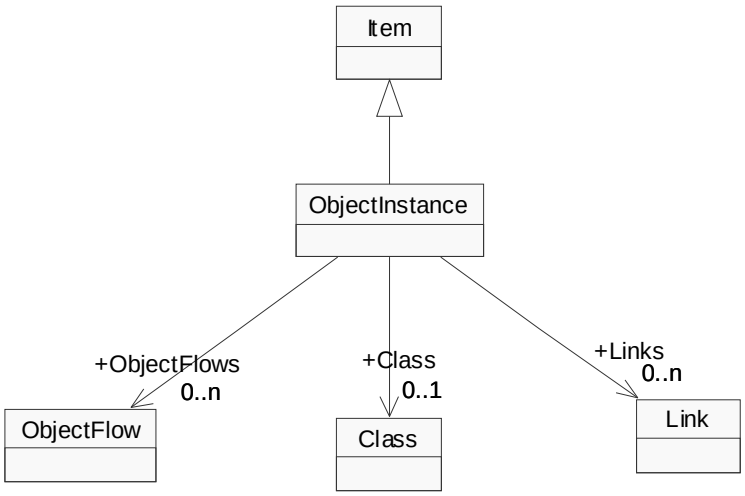
ObjectInstance (Rose Adapter)

The ObjectInstance class exposes a set of properties and methods that:
Determine the characteristics of objects in a model (for example, the class associated with the object and whether multiple instances of the object exist)
Allow you to retrieve objects from a model

Class Hierarchy: Artifact>Item>ObjectInstance

SubClasses of ObjectInstance

ObjectInstance has no subclasses.



Properties Specific to ObjectInstance

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
IsClass		True if the ObjectInstance is a class.
MultipleInstances		True if the Multiple Instances box is checked; otherwise False.
MyClassName		The ObjectInstance class name.
Name	Item	Name of the item.

Persistence		Persistent, Static, or Transient depending on the value of the Persistence radio control in the object specification.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to ObjectInstance

Name	Kind	Class	Description
Class	0..1	Class	The class of the object.
Links	0..n	Link	The links associated with the object.
ObjectFlows	0..n	ObjectFlow	The associated ObjectFlows with this ObjectInstance.

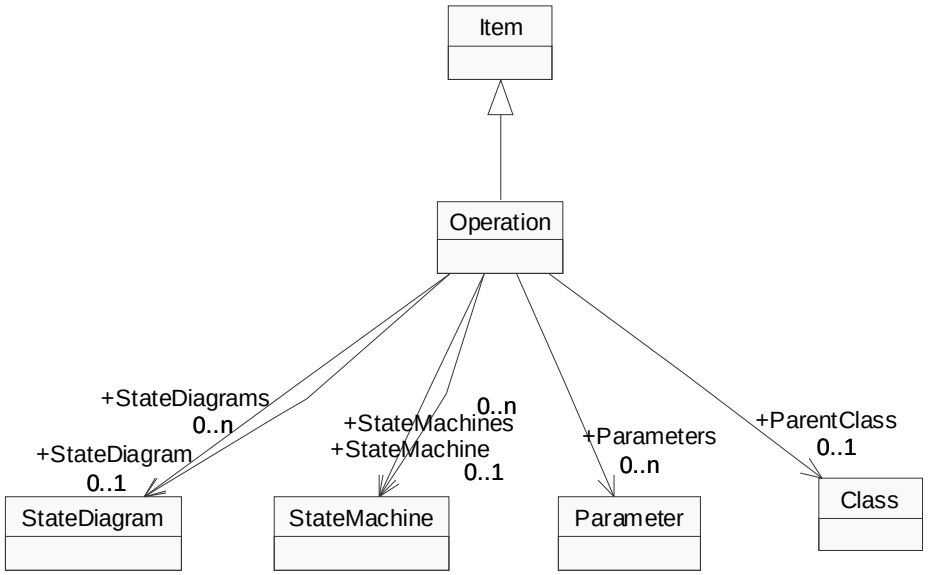
Operation (Rose Adapter)

Operations denote services provided by the class. Operations can be methods for accessing and modifying class fields or methods that implement characteristic behaviors of a class.
The operations of a class are listed in the Operations list box in the class specification. Rational Rose stores operation information in an operation specification. You can access operation specifications only through the class specification.

Class Hierarchy: Artifact>Item>Operation

SubClasses of Operation

Operation has no subclasses.



Properties Specific to Operation

Properties	Inherited From	Description
AdaImage		An Ada code segment that represents the declaration of the operation. This image is derived from the operation name and the operation parameters. Although the AdaImage is semantically consistent with your actual code, it may

		differ in terms of format, depending on the rules and styles you use for code generation and/or reverse engineering.
COMImage		A COM code segment that represents the declaration of the operation.
Concurrency		Denotes the semantics of the operation in the presence of multiple threads of control. Returns Sequential, Guarded, or Synchronous, depending on the state of the Concurrency radio control in the More dialog of the operation specification.
CppImage		A C++ code segment that represents the prototype of the operation. This image is derived from the operation name and the operation parameters. Although the C++Image is semantically consistent with your actual code, it may differ in terms of format, depending on the rules and styles you use for code generation and/or reverse engineering.
Documentation	Item	Documentation for the item.
Exceptions		Textual list of the exceptions that can be raised by the operation. The Exceptions text field appears in the More dialog of the operation specification.
ExportControl		Specifies the type of access allowed by the class for this operation. Will return Public, Protected, Private, or Implementation, depending on the state of the Export Control radio control in the operation specification.
HasStateDiagram		True if the operation has an associated StateDiagram
JavaImage		A Java code segment that represents the declaration of the operation.
Name	Item	Name of the item.

Postconditions		Text describing the post-conditions of the operation. The PostText is that text which appears in the Dynamic Semantics field of the operation specification when the Post radio button is selected.
Preconditions		Text describing the preconditions of the operation. The PreText is that text which appears in the Dynamic Semantics field of the operation specification when the Pre radio button is selected.
Protocol		The Protocol field lists a set of operations that a client may perform on an object and the legal orderings in which they may be invoked. The protocol of an operation has no semantic impact. The Protocol text field appears in the More dialog of the operation specification.
Qualification		Identifies language-specific features that allow you to qualify the method. The Qualification text field appears in the More dialog of the operation specification.
QualifiedName	Item	Qualified name of the item.
ReturnType		For operations that are functions, refers to the class that is returned by the function. The ReturnClass text field appears in the Return Class field on the operation specification.
Semantics		Text describing the action of the main operation. The SemanticsText is that text which appears in the Dynamic Semantics field of the operation specification when the Semantics radio button is selected.
Size		Text describing the size of the class.
Stereotype	Item	Stereotype of the item.

Time		A statement about the relative or absolute time required to complete an operation. The Time text field appears in the More dialog of the operation specification.
UMLImage		The image of the operation and parameters using UML standard notation.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Operation

Name	Kind	Class	Description
Parameters	0..n	Parameter	The formal parameters of the Operation. These appear in the Arguments list box in the operation specification.
ParentClass	0..1	Class	Class to which this Operation belongs.
StateDiagram	0..1	StateDiagram	The top-level state diagram associated with this Operation.
StateDiagrams	0..n	StateDiagram	All state diagrams associated with this Operation.
StateMachine	0..1	StateMachine	The top-level state machine associated with this Operation.
StateMachines	0..n	StateMachine	All state machines associated with this Operation.

Package (Rose Adapter)

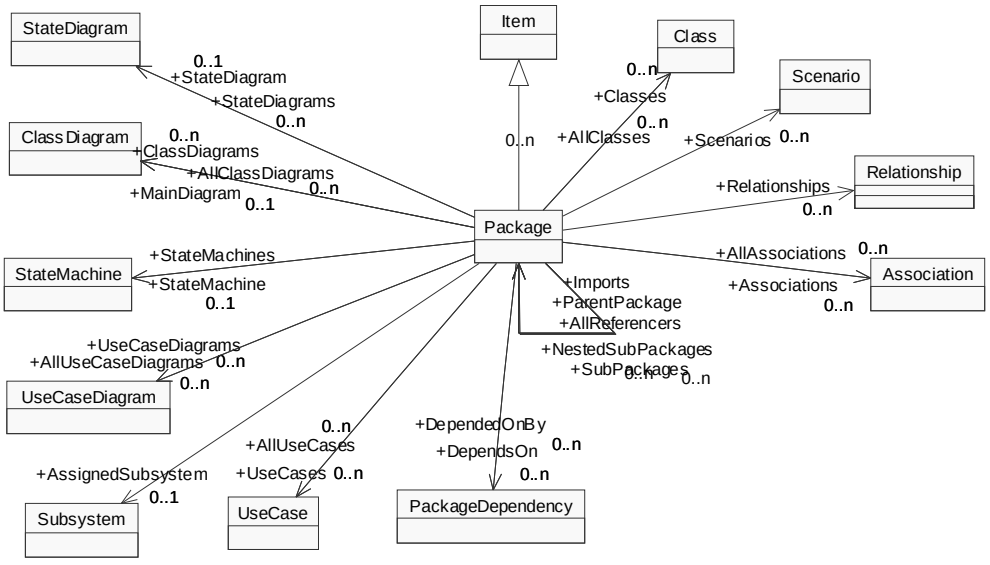
Packages serve to partition the logical model of a system. They are clusters of highly related classes that are themselves cohesive, but are loosely coupled relative to other such clusters. You can use packages to group classes and other packages. Rational Rose stores data describing the package in a package specification.

Note: When you create an OPEN command directly to a package, be sure to specify the name of the .mdl file and the name of the package, even if the package is contained in a separate .cat file.

Class Hierarchy: Artifact>Item>Package

SubClasses of Package

Package has no subclasses.



Properties Specific to Package

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
Global		True if the Package is global, otherwise False.

HasAssignedSubsystem		True if the Package has a subsystem associated with it, otherwise False.
HasStateDiagram		True if the Package has a state/activity diagram.
IsUseCasePackage		True if the Package is a descendent of the UseCase View package, otherwise False.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Package

Name	Kind	Class	Description
AllAssociations	0..n	Association	All associations that are defined in this Package, or in any nested packages.
AllClassDiagrams	0..n	ClassDiagram	All class diagrams that are defined in this Package, or in any nested packages.
AllClasses	0..n	Class	All classes that are defined in this Package, or in any nested packages.
AllReferencers		Package	All packages that import this Package. Does not include indirect referencers.
AllUseCaseDiagrams	0..n	UseCaseDiagram	All use case diagrams that are defined in this Package, or in any nested packages.
AllUseCases	0..n	UseCase	All use cases that are defined in this Package, or in any nested packages.
AssignedSubsystem	0..1	Subsystem	The subsystem associated with this Package, as specified in the package specification.
Associations	0..n	Association	All associations that are immediate members of this Package.
ClassDiagrams	0..n	ClassDiagram	All class diagrams that are immediate members of this Package.

Classes	0..n	Class	All classes that are immediate members of this Package. All member classes are returned, regardless of whether they appear on any diagrams.
DependedOnBy	0..n	PackageDependency	Associates this Package as the supplier package in a package dependency
DependsOn	0..n	PackageDependency	Associates this Package as the receiver package in a package dependency
Imports	0..n	Package	All packages that are imported by this Package. Does not include indirect dependencies. For example if A imports B and B imports C, A does not directly import C.
MainDiagram	0..1	ClassDiagram	The diagram specifically called "Main."
NestedSubPackages	0..n	Package	All packages that are descendents of this Package.
ParentPackage	0..1	Package	The enclosing Package. This relationship will result in an error if applied to the TopLevelCategory.
Relationships	0..n	Relationship	Relationships defined within this Package.
Scenarios	0..n	Scenario	Scenarios associated with this Package.
StateDiagram	0..1	StateDiagram	The top-level state/activity diagram associated with this Package.
StateDiagrams	0..n	StateDiagram	All state diagrams associated with this Package.
StateMachine	0..1	StateMachine	The top-level state machine associated with this Package.
StateMachines	0..n	StateMachine	All state machines associated with this Package.
SubPackages	0..n	Package	All packages that are immediate children of this Package.
UseCaseDiagrams	0..n	UseCaseDiagram	The use case diagrams contained within this Package.
UseCases	0..n	UseCase	All use cases that are immediate members of this Package.

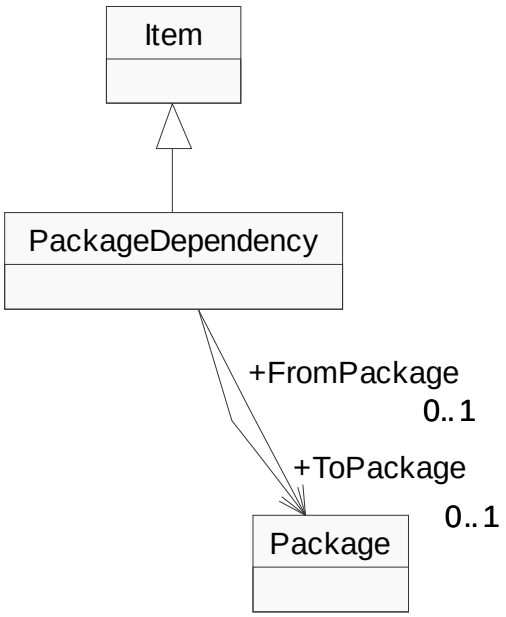
PackageDependency (Rose Adapter)

The package dependency indicates that one package in a model uses the services or facilities of another.

Class Hierarchy: Artifact>Item>PackageDependency

SubClasses of PackageDependency

PackageDependency has no subclasses.



Properties Specific to PackageDependency

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.

SupplierName		Name of the package that is the supplier in the package dependency.
UniqueID	Item	The unique ID of the item.

Relationships Specific to PackageDependency

Name	Kind	Class	Description
FromPackage	0..1	Package	The supplier package.
ToPackage	0..1	Package	The receiver package.

Parameter (Rose Adapter)

Formal parameter of an operation, instantiated class, or instantiated class utility.

Class Hierarchy: Artifact>Item>Parameter

SubClasses of Parameter

Parameter has no subclasses.

Properties Specific to Parameter

Properties	Inherited From	Description
Const		Returns True if the parameter is constant; otherwise False.
Documentation	Item	Documentation for the item.
InitValue		Initial value of the parameter.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
Type		The type of the parameter.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Parameter

Name	Kind	Class	Description
This class has no relationships.			

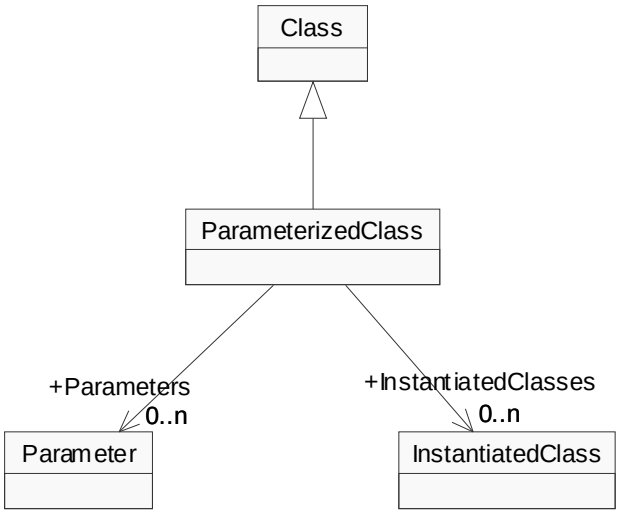
ParameterizedClass (Rose Adapter)

A parameterized class is a template for creating any number of instantiated classes that follow its format. A parameterized class declares formal parameters, which can be classes, objects, or operations.

Class Hierarchy: Item>Class>ParameterizedClass

SubClasses of ParameterizedClass

ParameterizedClass has no subclasses.



Properties Specific to ParameterizedClass

Properties	Inherited From	Description
Abstract	Class	True if the Abstract check box is selected in the class specification, otherwise False.
Cardinality	Class	The string in the Cardinality field of the class specification.
Concurrency	Class	Returns Sequential, Guarded, Active, or Synchronous, depending on the value of

		the Concurrency radio control in the More dialog of the class specification.
Documentation	Item	Documentation for the item.
ExportControl	Class	Returns Public or Implementation, depending on the value of the Export Control radio control in the class specification.
FundamentalType	Class	Returns True if this class is a fundamental type.
HasStateDiagram	Class	Returns True if the class has an associated state diagram, otherwise False.
IsNested	Class	Returns True if the Class is nested, otherwise False.
Kind	Class	The kind of Class.
Name	Item	Name of the item.
Persistence	Class	This property is Persistent or Transient, depending on the value of the Persistence radio control in the More dialog of the class specification.
QualifiedName	Item	Qualified name of the item.
Space	Class	The string in the Space field of the More dialog of the class specification.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to ParameterizedClass

Name	Kind	Class	Description
InstantiatedClasses	0..n	InstantiatedClass	All instantiated classes of this parameterized class.
Parameters	0..n	Parameter	Formal, generic parameters declared by the parameterized class. The parameters appear in the Parameters list box in the More dialog of the class specification.

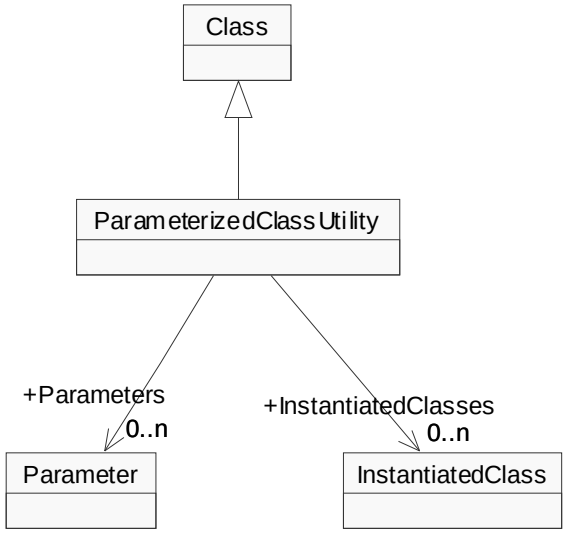
ParameterizedClassUtility (Rose Adapter)

A parameterized class utility is a set of operations or functions that are not associated with a higher level class (free subprograms) and are defined in terms of formal parameters. Parameterized class utilities are used as templates for creating instantiated class utilities.

Class Hierarchy: Item>Class>ParameterizedClassUtility

SubClasses of ParameterizedClassUtility

ParameterizedClassUtility has no subclasses.



Properties Specific to ParameterizedClassUtility

Properties	Inherited From	Description
Abstract	Class	True if the Abstract check box is selected in the class specification, otherwise False.
Cardinality	Class	The string in the Cardinality field of the class specification.

Concurrency	Class	Returns Sequential, Guarded, Active, or Synchronous, depending on the value of the Concurrency radio control in the More dialog of the class specification.
Documentation	Item	Documentation for the item.
ExportControl	Class	Returns Public or Implementation, depending on the value of the Export Control radio control in the class specification.
FundamentalType	Class	Returns True if this class is a fundamental type.
HasStateDiagram	Class	Returns True if the class has an associated state diagram, otherwise False.
IsNested	Class	Returns True if the Class is nested, otherwise False.
Kind	Class	The kind of Class.
Name	Item	Name of the item.
Persistence	Class	This property is Persistent or Transient, depending on the value of the Persistence radio control in the More dialog of the class specification.
QualifiedName	Item	Qualified name of the item.
Space	Class	The string in the Space field of the More dialog of the class specification.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to ParameterizedClassUtility

Name	Kind	Class	Description
InstantiatedClasses	0..n	InstantiatedClass	All instantiated class utilities of this parameterized class utility.
Parameters	0..n	Parameter	Formal, generic parameters declared by the parameterized class utility. The parameters appear in the Parameters list box in the More dialog of the class specification.

Process (Rose Adapter)

A process transforms data values. Lowest-level processes are pure functions without side effects.

Class Hierarchy: Artifact>Item>Process

SubClasses of Process

Process has no subclasses.

Properties Specific to Process

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
Priority		The priority of the Process.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Process

Name	Kind	Class	Description
This class has no relationships.			

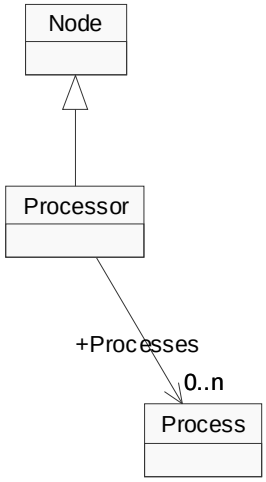
Processor (Rose Adapter)

A processor is a hardware component capable of executing programs.

Class Hierarchy: Item>Node>Processor

SubClasses of Processor

Processor has no subclasses.



Properties Specific to Processor

Properties	Inherited From	Description
Characteristics	Node	Characteristics of the processor or device.
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Scheduling		The text in the Scheduling field of the processor specification.
Stereotype	Item	Stereotype of the item.

UniqueID	Item	The unique ID of the item.

Relationships Specific to Processor

Name	Kind	Class	Description
Processes	0..n	Process	Processes defined by this Processor.

Property (Rose Adapter)

A code-generation property associated with the model, a package, a subsystem, a class, an association, a has relationship, an attribute, a module, or an operation.

Class Hierarchy: Artifact>Property

SubClasses of Property

Property has no subclasses.

Properties Specific to Property

Properties	Inherited From	Description
Name		Name of the property.
ParentUID		The unique ID of this property's parent artifact type.
ToolName		The name of the tool, or tab, for the property, such as “cg” or “DDL.”
Value		String equivalent of the value associated with the property.

Relationships Specific to Property

Name	Kind	Class	Description
This class has no relationships.			

RealizeRelationship (Rose Adapter)

A realize relationship between a logical class and a component class shows that the component class realizes the operations defined by the logical class.

Class Hierarchy: Item>Relationship>RealizeRelationship

SubClasses of RealizeRelationship

RealizeRelationship has no subclasses.

Properties Specific to RealizeRelationship

Properties	Inherited From	Description
ClientCardinality	Relationship	Indicates the number of possible links from an instance of the client class to an instance of the supplier class. Can be the same values as those listed in SupplierCardinality.
Documentation	Item	Documentation for the item.
ExportControl	Relationship	Specifies the type of access allowed between classes. Returns Public, Protected, Private, or Implementation, depending on the state of the Access radio control on the relationship specification. Access is also shown by adornments on relationships in diagrams.
Kind	Relationship	Kind of the relationship, which will be one of: AggregateRole, AssociationRole, HasRelationship, InheritsRelationship, or UsesRelationship.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
SupplierCardinality	Relationship	Indicates the number of possible links from an instance of the supplier class to an instance of the client class. Can be one the following values: n, 1, 0..n, 1..n,

		0..1, <literal>, <literal>..n, or <literal>..<literal>.
SupplierName	Relationship	Name of the supplier class or use case.
UniqueID	Item	The unique ID of the item.

Relationships Specific to RealizeRelationship

Name	Kind	Class	Description
This class has no relationships.			

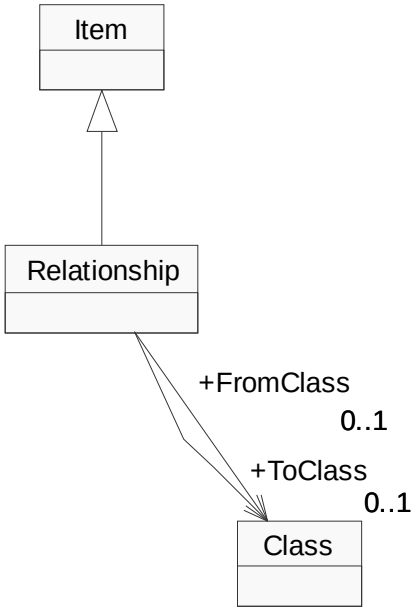
Relationship (Rose Adapter)

A semantic connection between two classes. Rational Rose stores relationship information in a relationship specification.

Class Hierarchy: Artifact>Item>Relationship

SubClasses of Relationship

HasRelationship
InheritRelationship
ObjectFlow
RealizeRelationship
Role
UsesRelationship



Properties Specific to Relationship

Properties	Inherited From	Description
ClientCardinality		Indicates the number of possible links from an instance of the client class to an instance of the supplier class. Can be the same values as those listed in SupplierCardinality.
Documentation	Item	Documentation for the item.
ExportControl		Specifies the type of access allowed between classes. Returns Public, Protected, Private, or Implementation, depending on the state of the Access radio control on the relationship specification. Access is also shown by adornments on relationships in diagrams.
Kind		Kind of the relationship, which will be one of: AggregateRole, AssociationRole, HasRelationship, InheritsRelationship, or UsesRelationship.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
SupplierCardinality		Indicates the number of possible links from an instance of the supplier class to an instance of the client class. Can be one the following values: n, 1, 0..n, 1..n, 0..1, <literal>, <literal>..n, or <literal>..<literal>.
SupplierName		Name of the supplier class or use case.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Relationship

Name	Kind	Class	Description
FromClass	0..1	Class	The client class. For example, if A Has a B, A is the client, or From class.
ToClass	0..1	Class	The supplier class. For example, if A Has a B, B is the supplier, or To class.

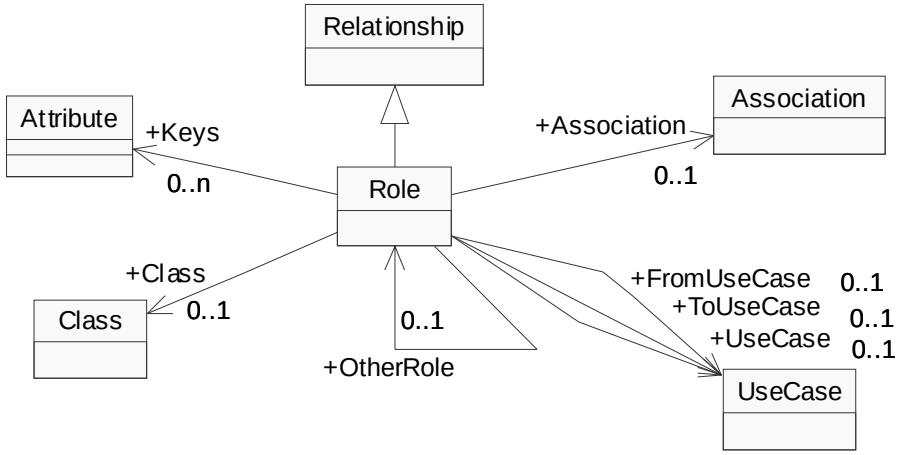
Role (Rose Adapter)

The purpose or capacity where one class associates with another.

Class Hierarchy: Item>Relationship>Role

SubClasses of Role

Role has no subclasses.



Properties Specific to Role

Properties	Inherited From	Description
Aggregate		Returns True if the role is an aggregate relationship.
Cardinality		Cardinality of this Role.
ClientCardinality	Relationship	Indicates the number of possible links from an instance of the client class to an instance of the supplier class. Can be the same values as those listed in SupplierCardinality.
Constraints		Text of the Contraints field in the Role specification.
Containment		Specifies the physical containment of the role. Returns Value, Reference, or Unspecified, depending on the state of

		the Containment radio control on the Role specification.
Documentation	Item	Documentation for the item.
ExportControl	Relationship	Specifies the type of access allowed between classes. Returns Public, Protected, Private, or Implementation, depending on the state of the Access radio control on the relationship specification. Access is also shown by adornments on relationships in diagrams.
Friend		Returns True if the Friend check box is selected in the Role specification, otherwise False.
Kind	Relationship	Kind of the relationship, which will be one of: AggregateRole, AssociationRole, HasRelationship, InheritsRelationship, or UsesRelationship.
Name	Item	Name of the item.
Navigable		Returns True if the Navigable check box is selected, otherwise False.
QualifiedName	Item	Qualified name of the item.
Static		Returns True if the Static check box is selected in the Role specification, otherwise False.
Stereotype	Item	Stereotype of the item.
SupplierCardinality	Relationship	Indicates the number of possible links from an instance of the supplier class to an instance of the client class. Can be one the following values: n, 1, 0..n, 1..n, 0..1, <literal>, <literal>..n, or <literal>..<literal>.
SupplierName	Relationship	Name of the supplier class or use case.
UniqueID	Item	The unique ID of the item.

Relationships Specific to Role

Name	Kind	Class	Description
Association	0..1	Association	The association that this Role is a part of.
Class	0..1	Class	The class associated with this Role
FromUseCase	0..1	UseCase	The supplier use case of the Role, if it is a use case.
Keys	0..n	Attribute	Each key is an attribute that uniquely defines a single target object.
OtherRole		Role	The role at the other end of the association.
ToUseCase	0..1	UseCase	The client use case of the inherits relationship, if it is a use case.
UseCase	0..1	UseCase	The use case associated with this Role.

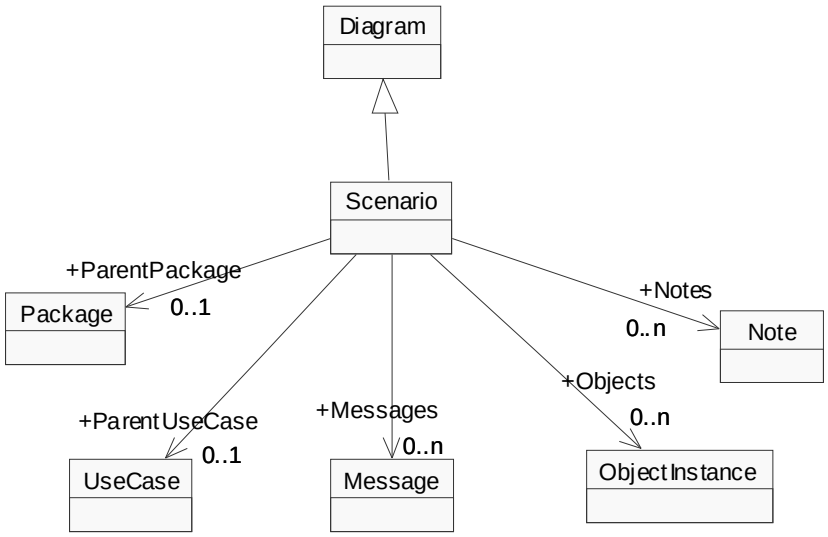
Scenario (Rose Adapter)

A scenario is an instance of a use case; it is an outline of events that occur during system execution.

Class Hierarchy: Artifact>Diagram>Scenario

SubClasses of Scenario

Scenario has no subclasses.



Properties Specific to Scenario

Properties	Inherited From	Description
DiagramType		The diagram type of this scenario
Documentation	Diagram	The documentation text associated with the Diagram.
MappedPoints	Diagram	A list of coordinates of the items in the Diagram. Each item is specified by a set of x/y coordinates designating the location of the corners of the item. The ordering of the items is the same as in the MappedAritifacts artifact collection.

Name	Diagram	Name of the Diagram.
ParentKind		The parent diagram of this scenario
QualifiedName	Diagram	Qualified name of the Diagram.
UniqueID	Diagram	The unique ID for the Diagram.

Relationships Specific to Scenario

Name	Kind	Class	Description
Messages	0..n	Message	Messages associated with this Scenario.
Notes	0..n	Note	Notes associated with this Scenario.
Objects	0..n	ObjectInstance	Object instances associated with the Scenario.
ParentPackage	0..1	Package	Parent package of this Scenario.
ParentUseCase	0..1	UseCase	Parent use case of this Scenario.

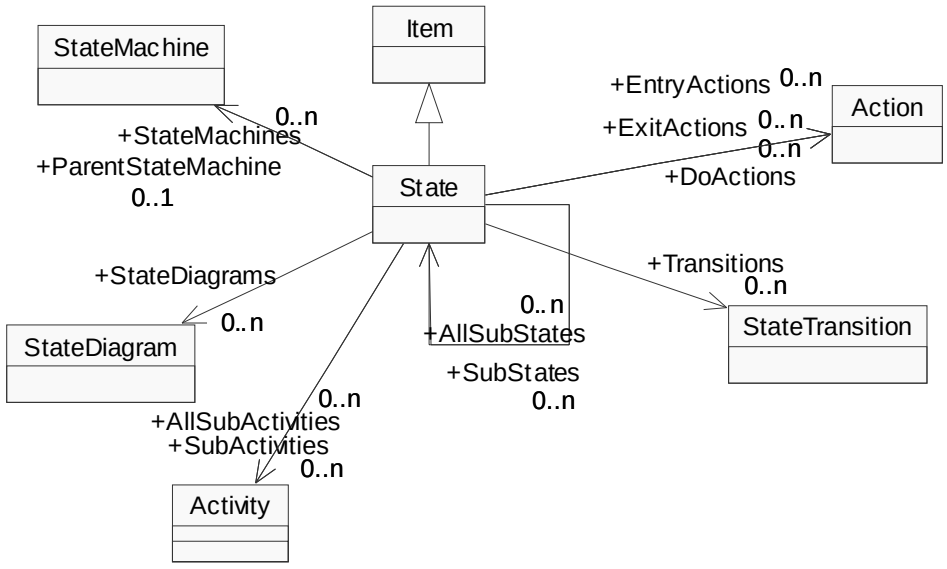
State (Rose Adapter)

The state of an object represents the cumulative history of its behavior. State encompasses all of the object's static properties and the current values of each property.

Class Hierarchy: Artifact>Item>State

SubClasses of State

State has no subclasses.



Properties Specific to State

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
History		The text in the History field of the state specification.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
StateKind		The kind of State. One of Start, Normal, or Stop.
Stereotype	Item	Stereotype of the item.

UniqueID	Item	The unique ID of the item.

Relationships Specific to State

Name	Kind	Class	Description
AllSubActivities	0..n	Activity	All activities that are associated with this State.
AllSubStates	0..n	State	All states that are associated with this State.
DoActions	0..n	Action	The Do actions associated with this State.
EntryActions	0..n	Action	The Entry actions associated with this State.
ExitActions	0..n	Action	The Exit actions associated with this State.
ParentStateMachine	0..1	StateMachine	The top-level state machine associated with this State.
StateDiagrams	0..n	StateDiagram	All state diagrams associated with this State.
StateMachines	0..n	StateMachine	All state machines associated with this State.
SubActivities	0..n	Activity	The activities that are part of this State.
SubStates		State	The states that are part of this State.
Transitions	0..n	StateTransition	The transitions that exit from this State.

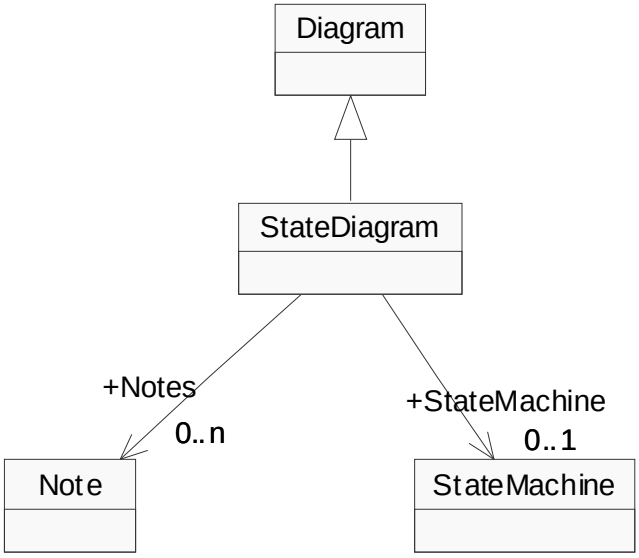
StateDiagram (Rose Adapter)

Depicts significant event-ordered behavior of a particular class. Each class may have one state diagram to describe its behavior.

Class Hierarchy: Artifact>Diagram>StateDiagram

SubClasses of StateDiagram

StateDiagram has no subclasses.



Properties Specific to StateDiagram

Properties	Inherited From	Description
Documentation	Diagram	The documentation text associated with the Diagram.
HasStateMachine		Returns True if the diagram includes a state activity model.
IsActivityDiagram		Returns True if the StateDiagram is an activity diagram
MappedPoints	Diagram	A list of coordinates of the items in the Diagram. Each item is specified by a set

		of x/y coordinates designating the location of the corners of the item. The ordering of the items is the same as in the MappedAritifacts artifact collection.
Name	Diagram	Name of the Diagram.
QualifiedName	Diagram	Qualified name of the Diagram.
UniqueID	Diagram	The unique ID for the Diagram.

Relationships Specific to StateDiagram

Name	Kind	Class	Description
Notes	0..n	Note	Notes that appear in the diagram.
StateMachine	0..1	StateMachine	The top-level state machine associated with this diagram.

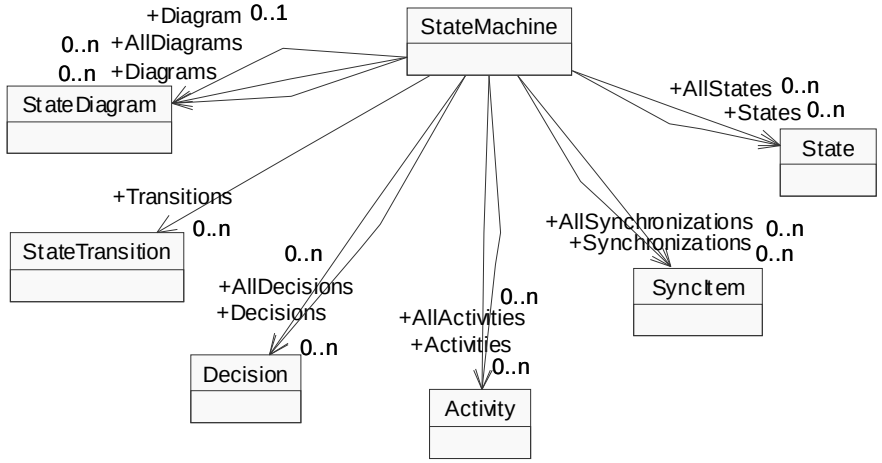
StateMachine (Rose Adapter)

A state machine can be defined as a behavior that specifies the valid sequences of activities that an object or interaction goes through during its life in response to events, together with its responses and actions.

Class Hierarchy: Artifact>StateMachine

SubClasses of StateMachine

StateMachine has no subclasses.



Properties Specific to StateMachine

Properties	Inherited From	Description
Documentation		The documentation for the StateMachine.
HasDiagram		True if the state activity model has at least one state or activity diagram.
Name		Name of the state activity model.
Stereotype		The stereotype of the StateMachine.
UniqueID		The internal unique identifier of the state activity model.

Relationships Specific to StateMachine

Name	Kind	Class	Description
Activities	0..n	Activity	The activities defined in this state activity model.
AllActivities	0..n	Activity	The activities defined in both this state activity model and all nested state activity models.
AllDecisions	0..n	Decision	Decisions defined in both this state activity model and all nested state activity models.
AllDiagrams	0..n	StateDiagram	Diagrams defined in both this state activity model and all nested state activity models.
AllStates	0..n	State	All states that are associated with this state activity model.
AllSynchronizations	0..n	SyncItem	Synchronizations defined in both this state activity model and all nested state activity models.
Decisions	0..n	Decision	Decisions defined in this state activity model.
Diagram	0..1	StateDiagram	The (first) state or activity diagram associated with this state activity model.
Diagrams	0..n	StateDiagram	The state or activity diagrams associated with this state activity model.
States	0..n	State	States that are part of this state activity model.
Synchronizations	0..n	SyncItem	Synchronizations defined in this state activity model.
Transitions	0..n	StateTransition	Transitions that are part of this state activity model.

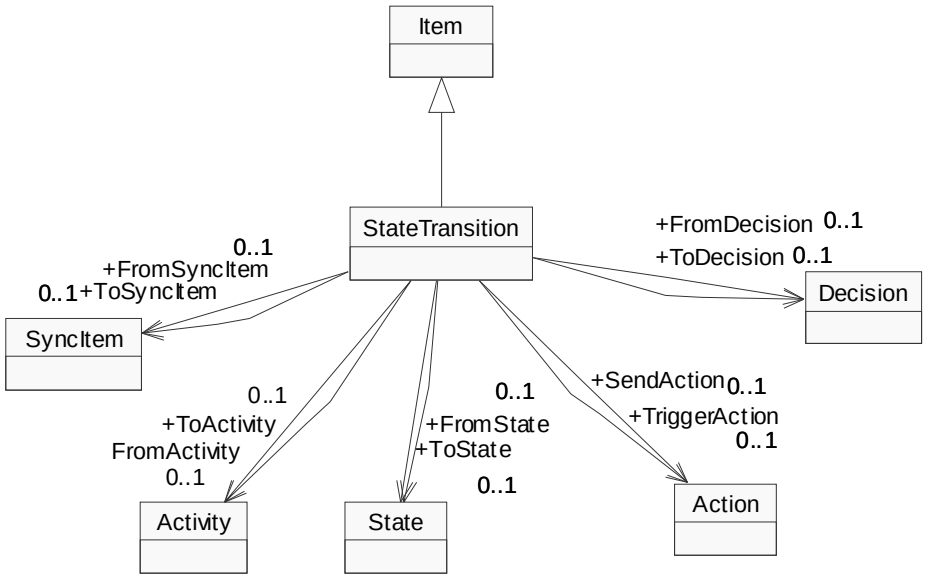
StateTransition (Rose Adapter)

A state transition is a change of state caused by an event. Use state transitions to connect two states in a state diagram or show state transitions from a state to itself.

Class Hierarchy: Artifact>Item>StateTransition

SubClasses of StateTransition

StateTransition has no subclasses.



Properties Specific to StateTransition

Properties	Inherited From	Description
CausingArguments		Arguments that accompany the causing event.
CausingEventName		Name of the event that causes this transition.
Documentation	Item	Documentation for the item.
GuardCondition		

KindofFromItem		Returns a string for the artifact type of the 'From', which is set to either State, or Activity or SyncItem.
KindofToItem		Returns a string for the artifact type of the 'To', which is set to either State, or Activity or SyncItem.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
SendArguments		Arguments that accompany the trigger event.
SendEventName		Name of the event triggered by the transition.
SendTarget		Name of the object that will receive the transition event.
Stereotype	Item	Stereotype of the item.
SupplierName		Name of the object that supplies the transition event.
UniqueID	Item	The unique ID of the item.

Relationships Specific to StateTransition

Name	Kind	Class	Description
FromActivity	0..1	Activity	Activity that this transition emanates from.
FromDecision	0..1	Decision	Decision that this transition emanates from.
FromState	0..1	State	State that this transition emanates from.
FromSyncItem	0..1	SyncItem	SyncItem that this transition emanates from.
SendAction	0..1	Action	Send action of this transition.
ToActivity	0..1	Activity	Activity that this transition leads to.
ToDecision	0..1	Decision	Decision that this transition leads to.
ToState	0..1	State	State that this transition leads to.
ToSyncItem	0..1	SyncItem	SyncItem that this transition leads to.
TriggerAction	0..1	Action	Action that triggers this transition.

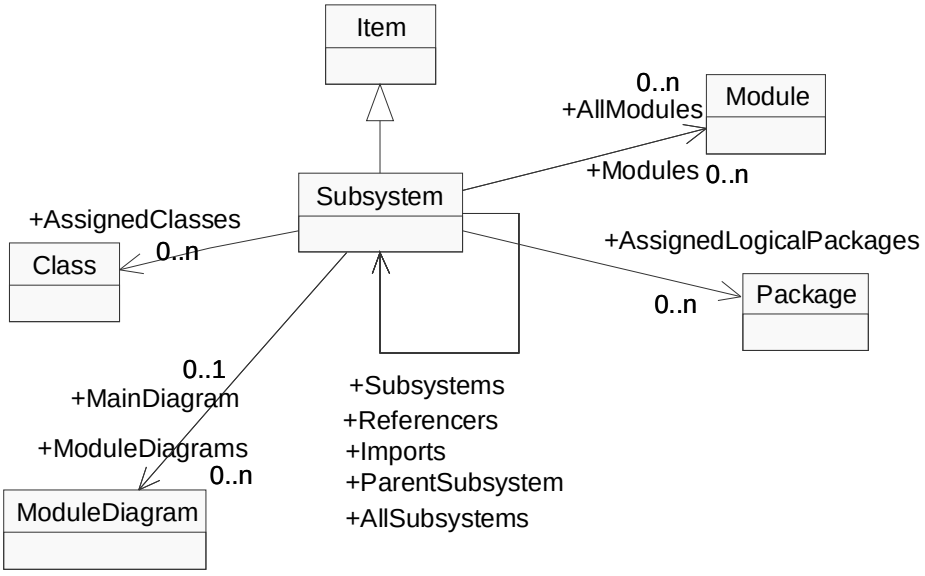
Subsystem (Rose Adapter)

Subsystems represent clusters of logically related components. They parallel the role played by packages for class diagrams, allowing you to partition the physical model of the system. Each subsystem can contain components and other subsystems. Each module in your system must reside in a single subsystem or at the Component View of the model.

Class Hierarchy: Artifact>Item>Subsystem

SubClasses of Subsystem

Subsystem has no subclasses.



Properties Specific to Subsystem

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.

UniqueID	Item	The unique ID of the item.

Relationships Specific to Subsystem

Name	Kind	Class	Description
AllModules	0..n	Module	All modules associated with this Subsystem.
AllSubsystems	0..n	Subsystem	All subsystems associated with this Subsystem.
AssignedClasses	0..n	Class	The classes assigned to this Subsystem.
AssignedLogicalPackages	0..n	Package	The logical packages assigned to this Subsystem.
Imports	0..n	Subsystem	All other Subsystems that this Subsystem directly depends on. Does not include indirect dependencies. For example if A imports B and B imports C, A does not directly import C.
MainDiagram	0..1	ModuleDiagram	All main module diagram contained in this Subsystem.
ModuleDiagrams	0..n	ModuleDiagram	All module diagrams contained in this Subsystem.
Modules	0..n	Module	The modules contained in this Subsystem.
ParentSubsystem	0..1	Subsystem	The parent Subsystem of this Subsystem.
Referencers	0..n	Subsystem	All other Subsystems that directly depend on this Subsystem. Does not include indirect referencers. For example if A imports B and B imports C, A is not a direct referencer of C.
Subsystems	0..n	Subsystem	The subsystems contained in this Subsystem.

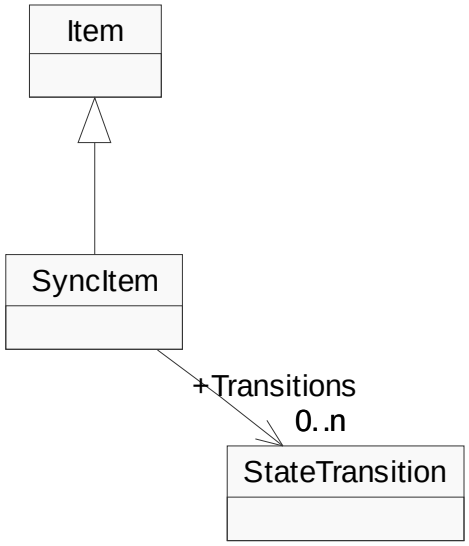
SyncItem (Rose Adapter)

The SyncItem class is an abstract class that exposes the Rose synchronization functionality in the extensibility interface.

Class Hierarchy: Artifact>Item>SyncItem

SubClasses of SyncItem

SyncItem has no subclasses.



Properties Specific to SyncItem

Properties	Inherited From	Description
Documentation	Item	Documentation for the item.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to SyncItem

Name	Kind	Class	Description
Transitions	0..n	StateTransition	The state transitions associated with this SyncItem.

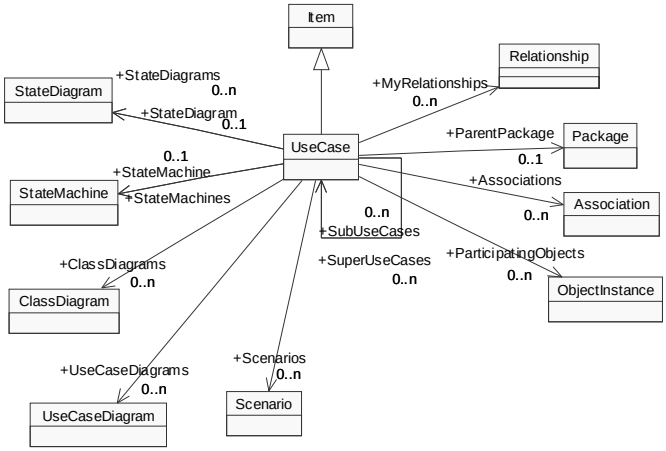
UseCase (Rose Adapter)

A sequence of transactions performed by a system in response to a triggering event initiated by an actor to the system. A full use case should provide a measurable value to an actor when the actor is performing a certain task. A use case contains all the events that can occur between an actor-use case pair, not necessarily the ones that will occur in any particular scenario. A use case contains a set of scenarios that explain various sequences of interaction within the transaction.

Class Hierarchy: Artifact>Item>UseCase

SubClasses of UseCase

UseCase has no subclasses.



Properties Specific to UseCase

Properties	Inherited From	Description
Abstract		True if the abstract check box is checked.
Documentation	Item	Documentation for the item.
HasStateDiagram		True if the UseCase has an associated state diagram.
Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Rank		The rank of the UseCase.
RequisiteProDocName		The associated ReqPro ReqDocument name.
RequisiteProProjectPath		The associated ReqPro Project path.
RequisiteProReqtGUID		The associated ReqPro Requirement GUID.
Stereotype	Item	Stereotype of the item.
UniqueID	Item	The unique ID of the item.

Relationships Specific to UseCase

Name	Kind	Class	Description
Associations	0..n	Association	The associations where this UseCase plays a role.
ClassDiagrams	0..n	ClassDiagram	The class diagrams included in this UseCase.
MyRelationships	0..n	Relationship	The inherits and role relationships defined by this UseCase.
ParentPackage	0..1	Package	The enclosing package.
ParticipatingObjects	0..n	ObjectInstance	The objects included in scenarios defined by this UseCase.
Scenarios	0..n	Scenario	The scenarios by this UseCase.
StateDiagram	0..1	StateDiagram	The top-level state diagram associated with this UseCase.

StateDiagrams	0..n	StateDiagram	All state diagrams associated with this UseCase.
StateMachine	0..1	StateMachine	The top-level state machine associated with this UseCase.
StateMachines	0..n	StateMachine	All state machines associated with this UseCase.
SubUseCases	0..n	UseCase	The UseCases that inherit from this UseCase.
SuperUseCases	0..n	UseCase	The use cases that this UseCase inherits from directly.
UseCaseDiagrams	0..n	UseCaseDiagram	The use case diagrams associated with this UseCase.

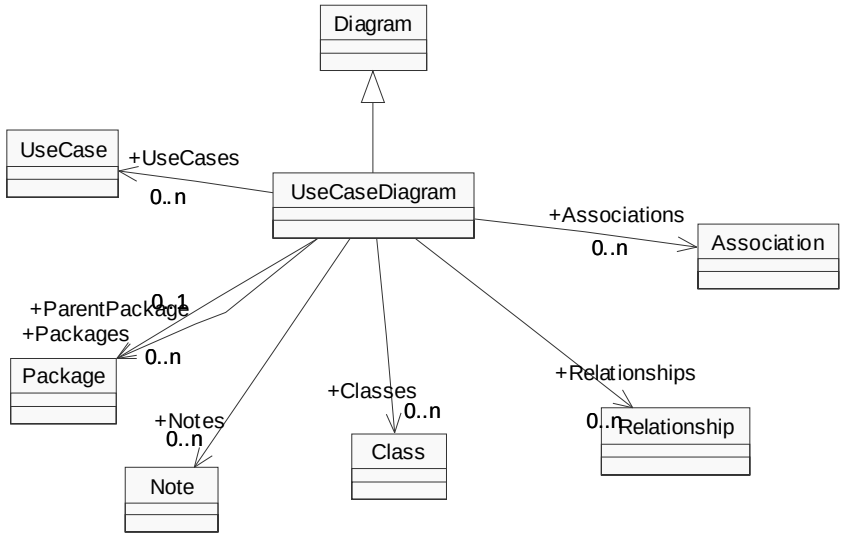
UseCaseDiagram (Rose Adapter)

A use case diagram shows the relationships between use cases and actors. Use case diagrams can be considered as filtered views into the model. They do not necessarily depict all the use cases or relationships in the model. For example, iterating over all the use cases in the main diagram of a package will not necessarily return all the use cases defined in that package.

Class Hierarchy: Artifact>Diagram>UseCaseDiagram

SubClasses of UseCaseDiagram

UseCaseDiagram has no subclasses.



Properties Specific to UseCaseDiagram

Properties	Inherited From	Description
Documentation	Diagram	The documentation text associated with the Diagram.
MappedPoints	Diagram	A list of coordinates of the items in the Diagram. Each item is specified by a set of x/y coordinates designating the location of the corners of the item. The ordering of the items is the same as in the MappedAritifacts artifact collection.

Name	Diagram	Name of the Diagram.
QualifiedName	Diagram	Qualified name of the Diagram.
UniqueID	Diagram	The unique ID for the Diagram.

Relationships Specific to UseCaseDiagram

Name	Kind	Class	Description
Associations	0..n	Association	The associations where this use case diagram plays a role.
Classes	0..n	Class	All of the classes that appear on the diagram.
Notes	0..n	Note	All of the notes associated with the diagram.
Packages	0..n	Package	All of the packages that appear on the diagram.
ParentPackage	0..1	Package	The package that contains the diagram, if applicable.
Relationships	0..n	Relationship	All of the relationships that appear on the diagram.
UseCases	0..n	UseCase	All of the use cases that appear on the diagram.

UsesRelationship (Rose Adapter)

Indicates that the client class depends on the supplier class to provide certain services, such as:
The client class accesses a value (constant or variable) defined in the supplier class.
Operations of the client class invoke operations of the supplier class.
Operations of the client class have signatures whose return class or arguments are instances of the supplier class.

Class Hierarchy: Item>Relationship>UsesRelationship

SubClasses of UsesRelationship

UsesRelationship has no subclasses.

Properties Specific to UsesRelationship

Properties	Inherited From	Description
ClientCardinality	Relationship	Indicates the number of possible links from an instance of the client class to an instance of the supplier class. Can be the same values as those listed in SupplierCardinality.
Documentation	Item	Documentation for the item.
ExportControl	Relationship	Specifies the type of access allowed between classes. Returns Public, Protected, Private, or Implementation, depending on the state of the Access radio control on the relationship specification. Access is also shown by adornments on relationships in diagrams.
InvolvesFriendship		Indicates whether the supplier class grants rights to the client class to access its non-public parts. Returns True, if the Friendship required check box is checked on the relationship specification. Otherwise, returns False.
Kind	Relationship	Kind of the relationship, which will be one of: AggregateRole, AssociationRole, HasRelationship, InheritsRelationship, or UsesRelationship.

Name	Item	Name of the item.
QualifiedName	Item	Qualified name of the item.
Stereotype	Item	Stereotype of the item.
SupplierCardinality	Relationship	Indicates the number of possible links from an instance of the supplier class to an instance of the client class. Can be one the following values: n, 1, 0..n, 1..n, 0..1, <literal>, <literal>..n, or <literal>..<literal>.
SupplierName	Relationship	Name of the supplier class or use case.
UniqueID	Item	The unique ID of the item.

Relationships Specific to UsesRelationship

Name	Kind	Class	Description

This class has no relationships.

TeamTest

Rational Test Manager

The following Classes are available through the TeamTest RSE adapter:

- Build
- Computer
- ConfiguredTestCase
- Group
- Iteration
- Log
- LogEvent
- LogFolder
- Project
- Requirement
- Script
- Session
- Suite
- TestCase
- TestCaseFolder
- TestCaseResult
- TestInput
- TestPlan
- UseCase
- User
- Variant
- VerificationPoint

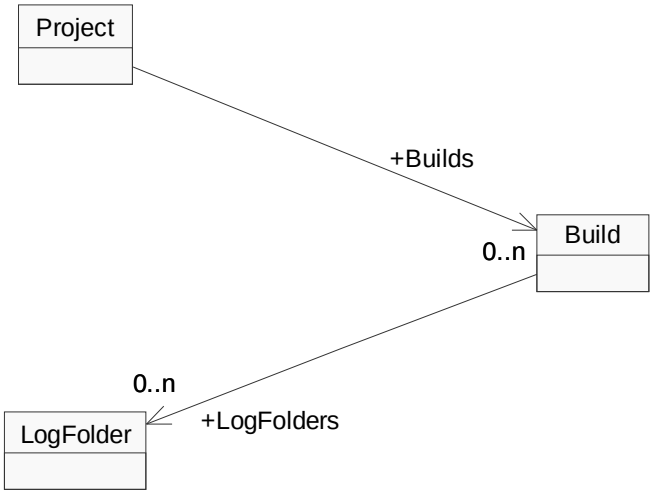
Build (TeamTest Adapter)

A build is a version of the application under test. Typically, engineers add new features or enhancements to each incremental build. You use Rational TestManager to manage builds. A build contains a collection of LogFolder artifacts which in turn contain Log artifacts with actual test results.

Class Hierarchy: Artifact>Build

SubClasses of Build

Build has no subclasses.



Properties Specific to Build

Properties	Inherited From	Description
CreatedBy		User that created the Build.
CreationDate		Date the Build was created.
Description		Description of the Build (from the General tab).
LastModifiedBy		User that last modified the Build.
ModificationDate		Date the Build was last modified.

Name	Name of the Build (from the General tab).
Owner	Owner of the Build (from the General tab).
ProjectName	Name of the project.
Status	Status of the Build (from the General tab).
UID	The unique ID of the Build.

Relationships Specific to Build

Name	Kind	Class	Description
LogFolders	0..n	LogFolder	Log folders that are included with this Build.

Computer (TeamTest Adapter)

You coordinate the activities of all your test scripts from a single NT computer where TestManager is running, known as the Local computer. From the Local computer, you create, run, and monitor suites.

During the execution of a test, you play back test scripts on the Local computer, or on computers that you have designated as Agent computers.

Class Hierarchy: Artifact>Computer

SubClasses of Computer

Computer has no subclasses.

Properties Specific to Computer

Properties	Inherited From	Description
Description		A description for the Computer.
IPAddress		The IP address of the Computer.
Name		Name of the Computer.
UID		The unique ID of the Computer.

Relationships Specific to Computer

Name	Kind	Class	Description
This class has no relationships.			

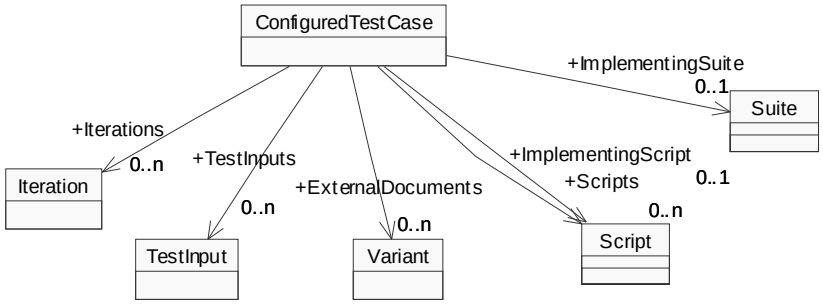
ConfiguredTestCase (TeamTest Adapter)

Configurations specify on what hardware and software configurations test cases must be run. A ConfiguredTestCase is similar to a TestCase except that it is associated with a single Configuration. A ConfiguredTestCase may have zero or more Iterations and zero or more Configurations.

Class Hierarchy: Artifact>ConfiguredTestCase

SubClasses of ConfiguredTestCase

ConfiguredTestCase has no subclasses.



Properties Specific to ConfiguredTestCase

Properties	Inherited From	Description
AcceptanceCriteria		The expected results or performance characteristics that define whether or not the ConfiguredTestCase passed or failed. For example: The response time range should be between 0.5 and 2.0 seconds for pass.
Configured		True if there is a configuration associated with this ConfiguredTestCase.
CreatedBy		User who created the ConfiguredTestCase.
CreationDate		Date the ConfiguredTestCase was created.

Custom1	Used to add a custom user-definable value.
Custom2	Used to add a custom user-definable value.
Custom3	Used to add a custom user-definable value.
Description	A description for this ConfiguredTestCase.
LastModifiedBy	User who last modified the ConfiguredTestCase.
ModificationDate	Date the ConfiguredTestCase was last modified.
Name	Name of the ConfiguredTestCase.
Owner	Owner of the ConfiguredTestCase.
Postconditions	Any cleanup steps that must be performed after the ConfiguredTestCase is run to bring the system back to a known state. For example, after you logon and successfully verify the test case, you need to logout (or bring the system back into a known state for the tests that follow).
Preconditions	Any setup dependency that is required for the ConfiguredTestCase to run. For example, you must have the proper user ID logon available in the system and the system must be in a logged out state.
Suspect	Returns True if an associated test input changes and the test case coverage for that test input is no longer sufficient.
UID	The unique ID of the ConfiguredTestCase.

Relationships Specific to ConfiguredTestCase

Name	Kind	Class	Description
ExternalDocuments	0..n	Variant	Other associated files to the ConfiguredTestCase. The variants represent an array of strings, but since you cannot have a relationship to string(s) you need an artifact type, hence the Variant.
ImplementingScript	0..1	Script	Returns an instance of the Script that implements this ConfiguredTestCase. (Returns only automated scripts.) If ImplementingSuite returns an object, then ImplementingScript will return NULL - and vice versa.
ImplementingSuite	0..1	Suite	Returns an instance of the Suite that implements this ConfiguredTestCase. If ImplementingSuite returns an object, then ImplementingScript will return NULL - and vice versa.
Iterations	0..n	Iteration	Iterations associated with a ConfiguredTestCase.
Scripts	0..n	Script	The scripts associated with the ConfiguredTestCase.
TestInputs	0..n	TestInput	Test inputs associated with a ConfiguredTestCase.

Group (TeamTest Adapter)

A user group is a basic building block for all performance testing suites. A user group is a collection of virtual testers that perform the same activity. Administrators and Public are the default groups.

Class Hierarchy: Artifact>Group

SubClasses of Group

Group has no subclasses.

Properties Specific to Group

Properties	Inherited From	Description
Description		Description of the Group.
IsDefault		True if the Group is the default Group.
Name		Name of the Group.

Relationships Specific to Group

Name	Kind	Class	Description
Users	0..n	User	Users in the Group.

Iteration (TeamTest Adapter)

Iterations specify when a test case must pass. An iteration is a defined span of time during a project. The end of an iteration is a milestone. An iteration says that at some point in time, the product has to meet a certain quality standard to reach a milestone. The quality standard is defined by the test cases that must pass.

An iteration may be assigned to Test Cases and/or Configured Test Cases. This indicates that the Test Case or Configured Test Case is expected to be executed and pass for that iteration.

Class Hierarchy: Artifact>Iteration

SubClasses of Iteration

Iteration has no subclasses.

Properties Specific to Iteration

Properties	Inherited From	Description
CreatedBy		Creator of the Iteration.
CreationDate		Date the Iteration was created.
Description		Description of the Iteration.
EndDate		End date of the Iteration.
LastModifiedBy		User who last modified the Iteration.
ModificationDate		Date the Iteration was modified.
Name		Name of the Iteration.
Owner		Owner of the Iteration.
StartDate		Start date of the Iteration.
UID		The unique ID of the Iteration.

Relationships Specific to Iteration

Name	Kind	Class	Description
This class has no relationships.			

Log (TeamTest Adapter)

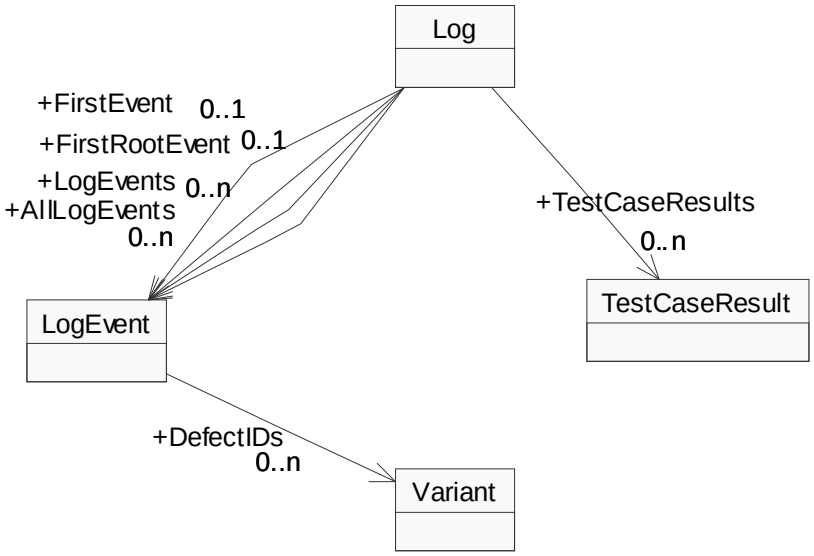
A log is a file that contains the record of events that occur while playing back a script or running a schedule. A log contains the results of all verification points executed as well as performance data. A log contains a collection of Test Case Result artifacts. A log also contains a collection of Load Test Report Output objects.

A log contains the results of a specific test or series of tests. The log contains a hierarchy of log events and a collection of Test Case Results.

Class Hierarchy: Artifact>Log

SubClasses of Log

Log has no subclasses.



Properties Specific to Log

Properties	Inherited From	Description
AgentLogFilesPath		Location of Log files on the agent computer.
CreatedBy		User that created this Log.
CreationDate		Date the Log was created.

Description	A description of the Log.
LastModifiedBy	The ID of the person who last modified the Log.
MasterLogFilePath	Location of the log files on the master computer.
ModificationDate	Date the Log was last modified.
Name	Name of the Log.
Owner	Owner of the Log.
PerformanceDataPath	Location of the performance data.
ProjectName	Test manager project name.
Suite	The test suite.
UAWPath	Location of unexpected active window data.
UID	The unique ID of the Log.
VPPath	Location of verification point data.

Relationships Specific to Log

Name	Kind	Class	Description
AllLogEvents	0..n	LogEvent	All associated log events.
FirstEvent	0..1	LogEvent	First event contained in this Log.
FirstRootEvent	0..1	LogEvent	Root event for the events in this Log.
LogEvents	0..n	LogEvent	Events contained in this Log.
TestCaseResults	0..n	TestCaseResult	Associated test case results to the Log.

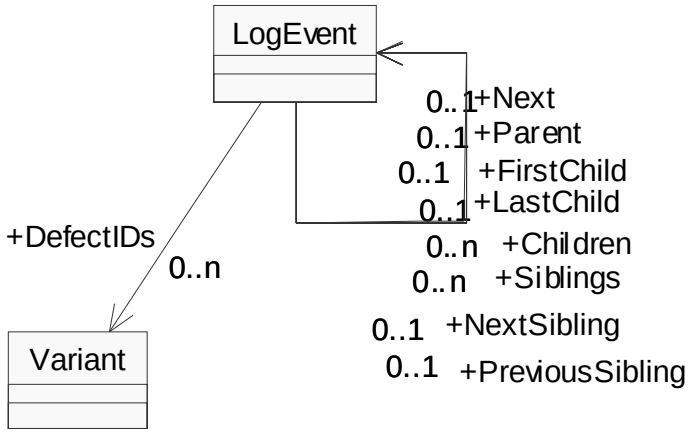
LogEvent (TeamTest Adapter)

Log events are generated when you run a test script, test case, or suite. Log events include script start and end, verification points, manual steps, and unexpected active windows.
LogEvent displays the type of event, the date and time the event was recorded, the script name, result information (if any), and other information about a log event.

Class Hierarchy: Artifact>LogEvent

SubClasses of LogEvent

LogEvent has no subclasses.



Properties Specific to LogEvent

Properties	Inherited From	Description
EndDateTime		Time the event ended.
EventCategoryText		Event category for the log event.
EventTypeText		Event type for the log event.
FailureDescription		Failure description (from the Result tab).
FailureReasonText		Failure reason (from the Result tab).
HasChildren		True if the log event has children log events.
ResultText		Location of the actual results.

StartDateTime	Start date and time (from the General tab).
UID	The unique ID for the log event.

Relationships Specific to LogEvent

Name	Kind	Class	Description
Children	0..n	LogEvent	Associated children log events.
DefectIDs	0..n	Variant	Defect IDs.
FirstChild	0..1	LogEvent	First child log event.
LastChild	0..1	LogEvent	Last child log event.
Next	0..1	LogEvent	Iterates to the next log event.
NextSibling	0..1	LogEvent	Iterates to the next sibling log event.
Parent	0..1	LogEvent	Name of the parent Log.
PreviousSibling	0..1	LogEvent	Iterates to the previous sibling log event.
Siblings	0..n	LogEvent	The associated sibling log events.

LogFolder (TeamTest Adapter)

A log folder is a directory that contains test logs. Contains a collection of Log artifacts.

Class Hierarchy: Artifact>LogFolder

SubClasses of LogFolder

LogFolder has no subclasses.

Properties Specific to LogFolder

Properties	Inherited From	Description
Name		Name of the LogFolder.
ProjectName		Test manager project name.

Relationships Specific to LogFolder

Name	Kind	Class	Description
Logs	0..n	Log	Logs contained in this folder.
SubFolders		LogFolder	Subfolders (LogFolders) contained in this folder.

Project (TeamTest Adapter)

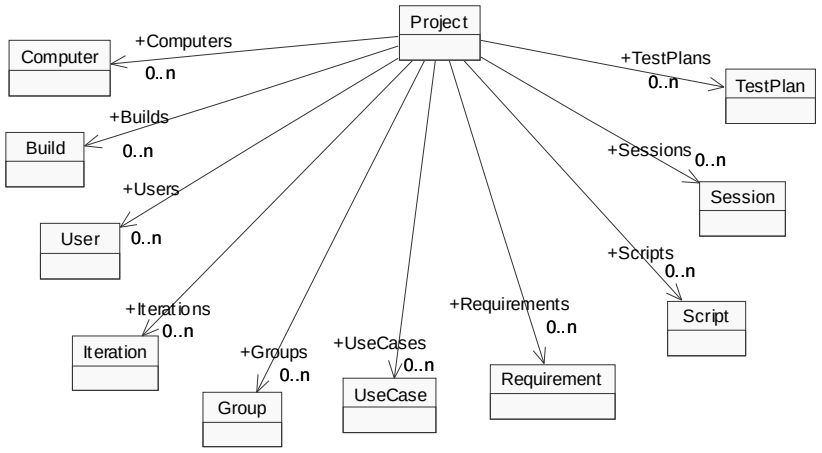
A project is a collection of data, including test assets, defects and requirements, that can facilitate the testing of one or more software components. Projects are managed primarily by the Rational Administrator. Projects contain multiple test plans.

Note: In order for the TeamTest adapter to return user-defined RequisitePro Requirement types, the word "Requirement" must be included in the new artifact type name.

Class Hierarchy: Artifact>Project

SubClasses of Project

Project has no subclasses.



Properties Specific to Project

Properties	Inherited From	Description
Directory		Directory that contains the Project.
Name		Name of the Project.
Path		Full path of the Project.

Relationships Specific to Project

Name	Kind	Class	Description
Builds	0..n	Build	Builds defined in the Project.
Computers	0..n	Computer	Computers associated with the Project.
Groups	0..n	Group	Groups associated with the Project.
Iterations	0..n	Iteration	Iterations of the Project.
Requirements	0..n	Requirement	Requirements associated with the project. Note: In order for the TeamTest adapter to return user-defined RequisitePro Requirement types, the word "Requirement" must be included in the new artifact type name.
Scripts	0..n	Script	All scripts included in the Project.
Sessions	0..n	Session	All sessions included in the Project.
TestPlans	0..n	TestPlan	Test plans associated with the Project.
UseCases	0..n	UseCase	Use cases associated with the Project.
Users	0..n	User	Users of the Project.

Requirement (TeamTest Adapter)

Represents a referenced RequisitePro Requirement.

Class Hierarchy: Artifact>Requirement

SubClasses of Requirement

Requirement has no subclasses.

Properties Specific to Requirement

Properties	Inherited From	Description
DBName		RequisitePro requirements database name.
FullTag		Unique identifier for the Requirement.
Name		Name of the Requirement, identified by the value of NodeID.
NodeID		The unique ID of the requirement. The ID of the requirement for which the client wants to determine the parental status. A NodeID is a GUID for a ReqPro requirement.
SourceUID		Input source ID. A handle, provided by the adapter, that identifies the connection to the ReqPro Project.
Text		Text of the Requirement.

Relationships Specific to Requirement

Name	Kind	Class	Description
This class has no relationships.			

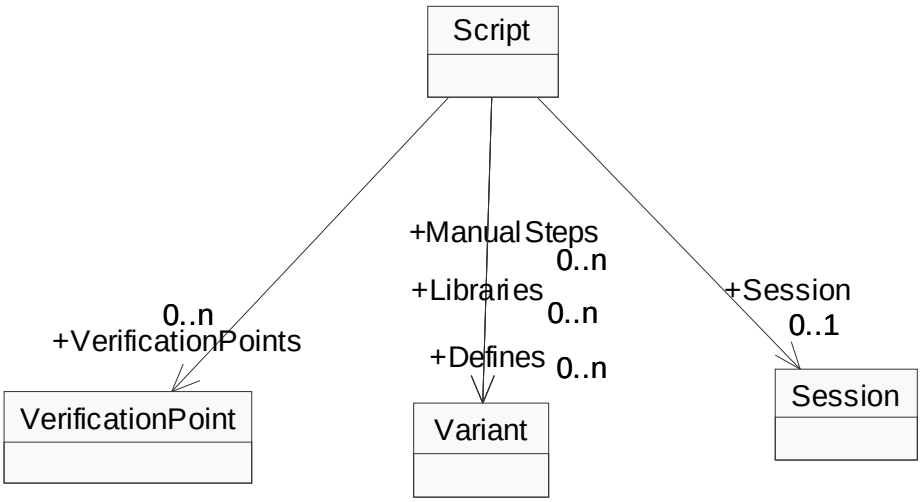
Script (TeamTest Adapter)

A script is a file of SQABasic or VU commands. Scripts may contain VerificationPoint.

Class Hierarchy: Artifact>Script

SubClasses of Script

Script has no subclasses.



Properties Specific to Script

Properties	Inherited From	Description
BinaryFilePath		Location of the binary file.
CreatedBy		User that created the Script.
CreationDate		Creation date of the Script.
Custom1		Value of the Custom 1 field (from the Custom tab).
Custom2		Value of the Custom 2 field (from the Custom tab).
Custom3		Value of the Custom 3 field (from the Custom tab).
Description		Description of the Script (from the General tab).

Environment	Pperating environment for the Script (from the General tab).
FilePath	Location of the Script.
LastModifiedBy	User who last modified the Script.
ModificationDate	Date of the Script modification.
Name	Name of the Script (from the General tab).
Notes	Related notes for the Script (from the Specifications tab).
Owner	Owner of the Script.
ProjectName	Test manager project name.
Purpose	Purpose of the Script (from the General tab).
ScriptType	Script type.
SpecFilePath	Path to the specification file (from the Specifications tab).
Text	Script text.
UID	The unique ID of the Script.

Relationships Specific to Script

Name	Kind	Class	Description
Defines	0..n	Variant	The Defines group is used for adding C-preprocessor directives, such as #define, #include, #ifdef, and #if to VU test scripts.
Libraries	0..n	Variant	The external C Libraries group is used to reference user-written external C libraries that you want to include when you compile VU test scripts.
ManualSteps	0..n	Variant	
Session	0..1	Session	An associated session to the Script.
VerificationPoints	0..n	VerificationPoint	Verification points included in the Script.

Session (TeamTest Adapter)

A session is a recording of network or API traffic. Scripts may reference sessions and sessions may reference scripts.

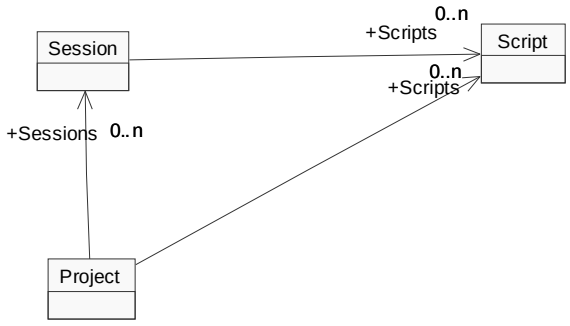
A session is the period of time required to play back a suite. A session thread begins when the first test script of a suite starts execution and ends when the last script finishes.

When you use Rational Robot to record a script, Robot records activities in a session, and then automatically creates a test script that represents the user's interactions with the server, as well as all queries and responses. If you have recorded a session in Robot, you can play back the test scripts in the session through TestManager.

Class Hierarchy: Artifact>Session

SubClasses of Session

Session has no subclasses.



Properties Specific to Session

Properties	Inherited From	Description
CreatedBy		Creator of the Session.
CreationDate		Session creation date.
Custom1		Value of the Custom 1 field (from the Custom tab).

Custom2	Value of the Custom 2 field (from the Custom tab).
Custom3	Value of the Custom 3 field (from the Custom tab).
Description	A description of the Session.
LastModifiedBy	User who last modified the Session.
ModificationDate	Date of the Session modification.
Name	Name of the Session.
Owner	Owner of the Session.
UID	The unique ID of the Session.

Relationships Specific to Session

Name	Kind	Class	Description
Scripts	0..n	Script	Scripts associated with this Session.

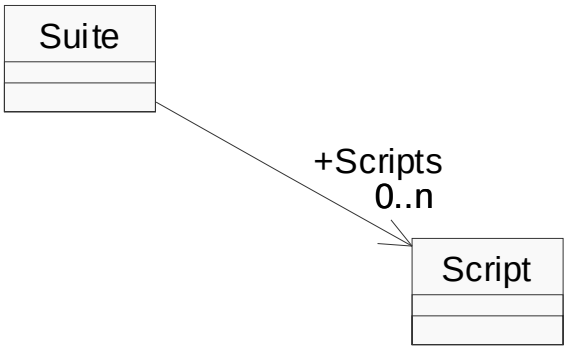
Suite (TeamTest Adapter)

A Suite (TestSuite) is a collection of Scripts.

Class Hierarchy: Artifact>Suite

SubClasses of Suite

Suite has no subclasses.



Properties Specific to Suite

Properties	Inherited From	Description
CreatedBy		Creator of the Suite.
CreationDate		Suite creation date.
Description		A description of the Suite.
LastModifiedBy		User who last modified the Suite.
ModificationDate		Date of the Suite modification.
Name		Name of the Suite.
Owner		Owner of the Suite.
UID		The unique ID of the Suite.

Relationships Specific to Suite

Name	Kind	Class	Description
Scripts	0..n	Script	Returns the set of Test Scripts included in the Suite.

TestCase (TeamTest Adapter)

Users plan what is to be tested and receive report results using test cases. A TestCase describes a specific flow of events through a feature.

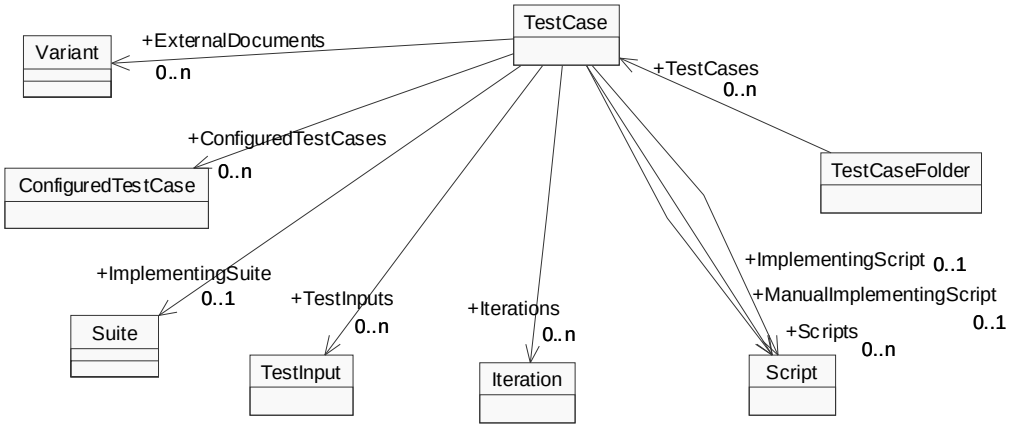
Test cases validate that the system is working the way that it's supposed to work. The test case is the artifact in TestManager that answers the question, "What do I need to test?" You develop test cases to validate particular behaviors. Each test case is owned by or assigned to a team member. This answers the question, "Who will do the testing?"

A test plan contains TestCases. You can create test cases within test case folders to organize your test cases hierarchically. A Test Case contains zero or more Configured Test Cases. A Test Case may have pointers to Iteration artifacts.

Class Hierarchy: Artifact>TestCase

SubClasses of TestCase

TestCase has no subclasses.



Properties Specific to TestCase

Properties	Inherited From	Description
AcceptanceCriteria		The acceptance criteria indicates what needs to be true in order for a particular TestCase to pass.
Configured		True if the TestCase has been configured.
CreatedBy		User that created the TestCase.
CreationDate		Creation date of the TestCase.
Custom1		Value of the Custom 1 field (from the Custom tab).
Custom2		Value of the Custom 2 field (from the Custom tab).
Custom3		Value of the Custom 3 field (from the Custom tab).
Description		A description of this TestCase.
LastModifiedBy		User who last modified the TestCase.
ModificationDate		Date of the TestCase modification.
Name		Name of the TestCase.
Owner		Owner of this TestCase.
Postconditions		Postconditions of the TestCase.
Preconditions		Preconditions of the TestCase.
Suspect		Returns True if an associated test input changes and the test case coverage for that test input is no longer sufficient.
UID		The unique ID of the TestCase.

Relationships Specific to TestCase

Name	Kind	Class	Description
ConfiguredTestCases	0..n	ConfiguredTestCase	Associated configured test cases.
ExternalDocuments	0..n	Variant	Other associated files to the TestCase. The variants represent an array of strings, but since you cannot have a relationship to string(s) you need an artifact type, hence the Variant.
ImplementingScript	0..1	Script	Returns an instance of the Script that implements this TestCase. (Returns only automated scripts.) If ImplementingSuite returns an object, then ImplementingScript will return NULL - and vice versa.
ImplementingSuite	0..1	Suite	Returns an instance of the Suite that implements this Test Case. If ImplementingSuite returns an object, then ImplementingScript will return NULL - and vice versa.
Iterations	0..n	Iteration	Iterations for this TestCase.
ManualImplementingScript	0..1	Script	
Scripts	0..n	Script	Scripts associated with this TestCase. Returns the automated TestScripts referenced by the TestSuite.
TestInputs	0..n	TestInput	Test inputs for this TestCase.

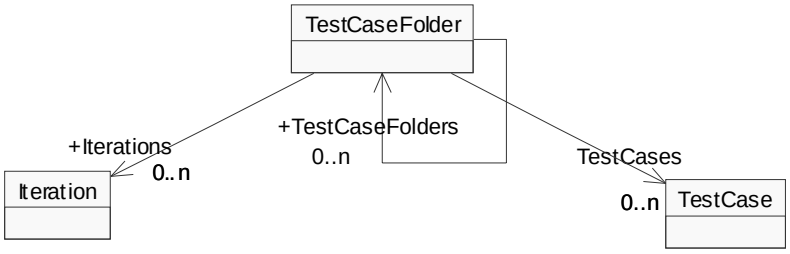
TestCaseFolder (TeamTest Adapter)

TestCaseFolders organize Test Cases. A TestCaseFolder is a directory that contains Test Cases. Test Case Folders can contain other Test Case Folders as well as Test Cases.

Class Hierarchy: Artifact>TestCaseFolder

SubClasses of TestCaseFolder

TestCaseFolder has no subclasses.



Properties Specific to TestCaseFolder

Properties	Inherited From	Description
CreatedBy		User that created the TestCaseFolder.
CreationDate		Creation date of the TestCaseFolder.
Description		A description of this TestCaseFolder.
LastModifiedBy		User who last modified the TestCaseFolder.
ModificationDate		Date of the TestCaseFolder modification.
Name		Name of the TestCaseFolder.
Owner		Owner of this TestCaseFolder.
QualifiedName		
UID		The unique ID of the TestCaseFolder.

Relationships Specific to TestCaseFolder

Name	Kind	Class	Description
Iterations	0..n	Iteration	Iterations contained in this TestCaseFolder.
TestCaseFolders	0..n	TestCaseFolder	Associated TestCaseFolders.
TestCases	0..n	TestCase	Test cases contained in this TestCaseFolder.

TestCaseResult (TeamTest Adapter)

A TestCaseResult is generated from a LogEvent that is recorded during test execution.

Class Hierarchy: Artifact>TestCaseResult

SubClasses of TestCaseResult

TestCaseResult has no subclasses.

Properties Specific to TestCaseResult

Properties	Inherited From	Description
ActualResult		The actual result.
InterpretedResult		The interpreted result.
IsPromoted		True if the test passes, False if the test fails.
Name		Name of the TestCaseResult.
Notes		Text for this TestCaseResult.
UID		The unique ID of the TestCaseResult.

Relationships Specific to TestCaseResult

Name	Kind	Class	Description
LogEvent	0..1	LogEvent	Associated LogEvent for this TestCaseResult.
TestCase	0..1	TestCase	Associated TestCase for the TestCaseResult.

TestInput (TeamTest Adapter)

A TestInput is any requirement, use case, change request, or other input that requires validation by a Test Case. Test inputs are anything that the test designer uses to determine what needs to be tested.

Class Hierarchy: Artifact>TestInput

SubClasses of TestInput

TestInput has no subclasses.

Properties Specific to TestInput

Properties	Inherited From	Description
Arg1		Used to add a custom user-definable argument.
Arg2		Used to add a custom user-definable argument.
Arg3		Used to add a custom user-definable argument.
Arg4		Used to add a custom user-definable argument.
Arg5		Used to add a custom user-definable argument.
CollIndex		
IsContainer		True if the TestInput contains a model, a use case, or a requirement.
Kind		The kind of TestInput.
Name		Name of the TestInput.
NeedsValidation		True if this TestInput needs to be validated.
SubType		
Type		The TestInput type.

Relationships Specific to TestInput

Name	Kind	Class	Description
This class has no relationships.			

TestPlan (TeamTest Adapter)

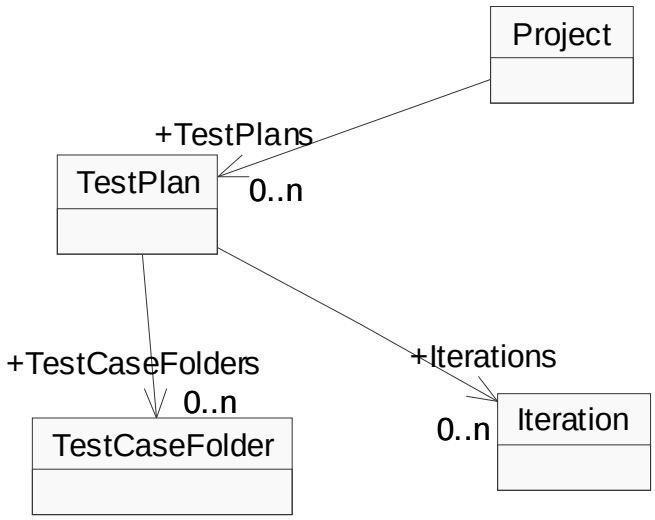
A TestPlan contains information about the purpose and goals of testing within the project, and the strategies to be used to implement and execute testing. Projects can contain multiple test plans.

Each test plan can contain test case folders and test cases. A test plan may contain zero or more Test Case Folders, Iterations, and Configurations.

Class Hierarchy: Artifact>TestPlan

SubClasses of TestPlan

TestPlan has no subclasses.



Properties Specific to TestPlan

Properties	Inherited From	Description
CreatedBy		Creater of the TestPlan.
CreationDate		Creation date of the TestPlan.
Custom1		Used to add a custom user-definable value.

Custom2	Used to add a custom user-definable value.
Custom3	Used to add a custom user-definable value.
Description	A description for the TestPlan.
LastModifiedBy	User who last modified the TestPlan.
ModificationDate	Most recent modification date of the TestPlan.
Name	Name of the TestPlan.
Owner	Owner of the TestPlan.
UID	The unique ID of the TestPlan.

Relationships Specific to TestPlan

Name	Kind	Class	Description
Iterations	0..n	Iteration	Iterations associated for this TestPlan.
TestCaseFolders	0..n	TestCaseFolder	TestCaseFolders in this TestPlan.

UseCase (TeamTest Adapter)

Represents a referenced Rose Use Case.

Class Hierarchy: Artifact>UseCase

SubClasses of UseCase

UseCase has no subclasses.

Properties Specific to UseCase

Properties	Inherited From	Description
Name		Name of the UseCase, identified by the value of NodeID.
NodeID		The unique ID of the UseCase. The ID of the UseCase for which the client wants to determine the parental status.
QualifiedName		The Rose UseCase qualified name.
SourceUID		Input source ID. A handle, provided by the adapter, that identifies the connection to the Rose UseCase.

Relationships Specific to UseCase

Name	Kind	Class	Description
This class has no relationships.			

User (TeamTest Adapter)

A user is an individual tester. Users are members of groups. The default user is admin.

Class Hierarchy: Artifact>User

SubClasses of User

User has no subclasses.

Properties Specific to User

Properties	Inherited From	Description
Company		User's company.
Department		User's department.
Email		User's e-mail address.
First		User's first name.
Last		User's last name.
Name		Name of the User.
Phone		User's telephone number.
Title		User's title.
UserID		User ID.

Relationships Specific to User

Name	Kind	Class	Description
Groups	0..n	Group	Groups this User is a member of.

Variant (TeamTest Adapter)

Variant is an internal type used for representing relationships between artifact types and simple data types. You cannot create these directly, you can only resolve them through relationships.

Class Hierarchy: Artifact>Variant

SubClasses of Variant

Variant has no subclasses.

Properties Specific to Variant

Properties	Inherited From	Description
CollIndex		Column index.
IntValue		Integer value.
RelName		Relationship type name.
StrValue		String value.

Relationships Specific to Variant

Name	Kind	Class	Description
This class has no relationships.			

VerificationPoint (TeamTest Adapter)

A verification point is a point in an SQABasic test script that confirms the state of one or more objects. Verification points capture some aspect of the application or system under test and store it away for later comparison to the actual state of the system or application.

Class Hierarchy: Artifact>VerificationPoint

SubClasses of VerificationPoint

VerificationPoint has no subclasses.

Properties Specific to VerificationPoint

Properties	Inherited From	Description
BaselineFilePath		Path of the associated baseline VerificationPoint.
DataType		
MetadataFilePath		
Name		Name of the VerificationPoint.
Type		Type of the VerificationPoint.

Relationships Specific to VerificationPoint

Name	Kind	Class	Description
This class has no relationships.			

Word

Microsoft Word

Users of the Word adapter should always call Refresh when use of the adapter is finished. The Word adapter Refresh cleans up the Word session by removing files that it opened. These files may confuse the users who are using the Word session.

The following Classes are available through the Word RSE adapter:

- Bookmark
- Document
- Heading
- Paragraph

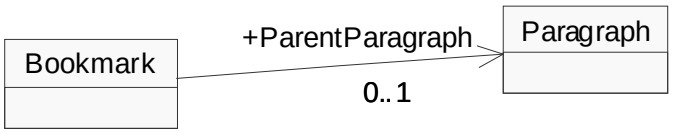
Bookmark (Word Adapter)

Bookmarks are found in a Word document.
A bookmark has a starting and ending point in the text. The start or end position is a character position or index, such as an integer. For example, a bookmark that starts at the beginning of text starts at position 0. The end of a bookmark may have the same position as the start, in which case the bookmark text is empty.

Class Hierarchy: Artifact>Bookmark

SubClasses of Bookmark

Bookmark has no subclasses.



Properties Specific to Bookmark

Properties	Inherited From	Description
EndPosition		End position of the Bookmark. Allows sorting of Bookmarks by the end position in the document.
FormattedText		Path to the formatted text defined by the Bookmark.
FormattedTextBefore		Path to the formatted text that precedes the beginning of the Bookmark and follows the end of the previous Bookmark. Or, the beginning of the document if there is no such Bookmark.
Name		Name of the Bookmark.
StartPosition		Start position of the Bookmark. Allows sorting of Bookmarks by the start position in the document.

Text	Text of the area defined by the Bookmark, pasted as a string.

Relationships Specific to Bookmark

Name	Kind	Class	Description
ParentParagraph	0..1	Paragraph	Paragraph that contains the first character of the Bookmark.

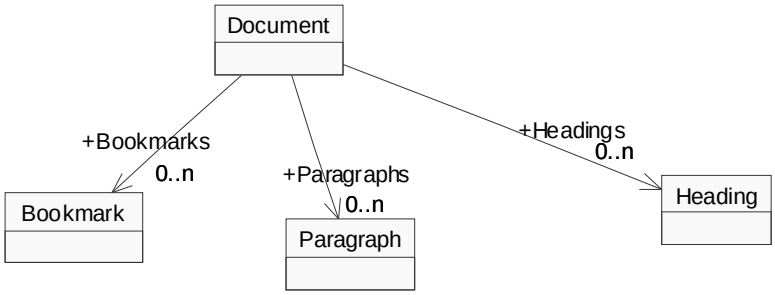
Document (Word Adapter)

A Word document.

Class Hierarchy: Artifact>Document

SubClasses of Document

Document has no subclasses.



Properties Specific to Document

Properties	Inherited From	Description
FormattedText		Path to the complete contents of the Word Document, pasted as formatted text.
FormattedTextAfterLastBookmark		Path to the formatted text beginning at the end of the last bookmark and ending at the end of the Document. If no bookmarks are found, the entire Document is returned.
FullName		Full name of the Document.
Name		Name of the Document.
Text		Complete contents of the Word Document, pasted as a string.

Relationships Specific to Document

Name	Kind	Class	Description
Bookmarks	0..n	Bookmark	Bookmarks defined in this Document.
Headings	0..n	Heading	All Headings found in this Document.
Paragraphs	0..n	Paragraph	All Paragraphs, including Headings, found in this Document.

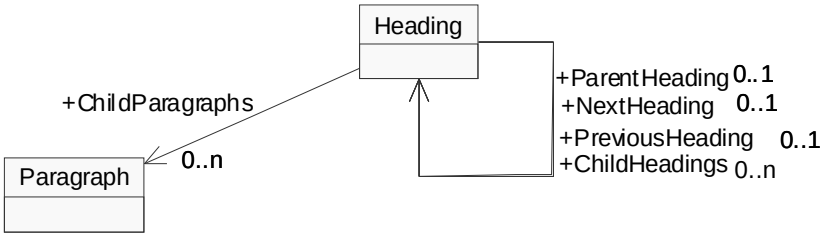
Heading (Word Adapter)

Headings are paragraphs with a style "Heading1", "Heading2", and so on.

Class Hierarchy: Artifact>Heading

SubClasses of Heading

Heading has no subclasses.



Properties Specific to Heading

Properties	Inherited From	Description
FormattedText		Path to the text of the Heading, pasted as formatted text.
Label		The label of the Heading. For example, "1.1.2."
Position		The character position of the first character of the paragraph.
StyleDescription		A description of the style.
StyleName		Style of the Heading. For example, "Normal Arial 10."
Text		Text of the Heading, pasted as a string.

Relationships Specific to Heading

Name	Kind	Class	Description
ChildHeadings	0..n	Heading	Headings contained within this Heading, one level in.
ChildParagraphs	0..n	Paragraph	Paragraphs contained within this Heading, including headings.
NextHeading	0..1	Heading	The next Heading at the same level.
ParentHeading	0..1	Heading	Parent Heading for this Heading, one level up.
PreviousHeading	0..1	Heading	Previous Heading at the same level.

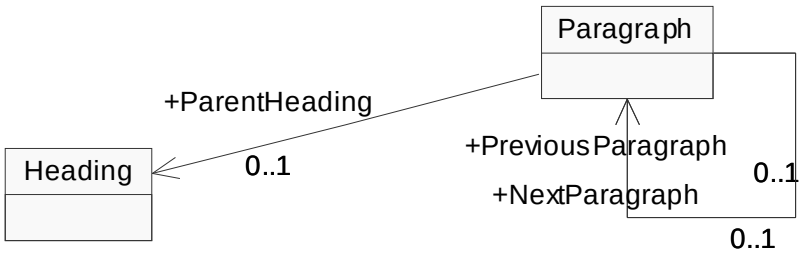
Paragraph (Word Adapter)

Paragraphs in a Word document.

Class Hierarchy: Artifact>Paragraph

SubClasses of Paragraph

Paragraph has no subclasses.



Properties Specific to Paragraph

Properties	Inherited From	Description
FormattedText		Path to the complete contents of the Word document, pasted as formatted text.
Position		Character position of the first character of the Paragraph.
StyleDescription		Description of the Paragraph style.
StyleName		Style name of the Paragraph. For example, "Normal."
Text		Complete contents of the Word document, pasted as a string.

Relationships Specific to Paragraph

Name	Kind	Class	Description
NextParagraph	0..1	Paragraph	Next Paragraph in the document.
ParentHeading	0..1	Heading	Nearest heading above the current Paragraph.
PreviousParagraph	0...1	Paragraph	Previous Paragraph in the document.