TestManager Domain

Introduction

The TestManager domain lets you extract information from Rational TestManager to use in reporting. This information can come from projects, builds, and scripts, among other things.

Aliases

The following names in the TestManager domain are aliased in the reporting interface:

Name in Domain	Name in Interface	Item Type / Location
ImplementingSuite ImplementingScript	AutomatedImplementingSuite AutomatedImplementingScript	Relationship / ConfiguredTestCase Class
EventTypeText ResultText FailureReasonText	EventType Result FailureReason	Property / LogEvent Class
SpecFilePath ScriptType	SpecificationFilePath Type	Property / Script Class
ManualSteps	StepDescriptions	Relationship / Script Class
ImplementingSuite ImplementingScript	AutomatedImplementingSuite AutomatedImplementingScript	Relationship / TestCase Class

Build

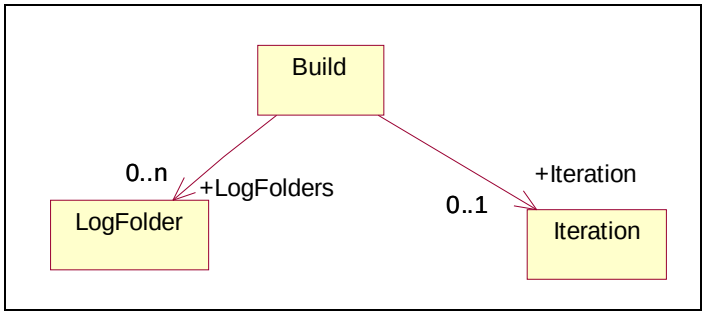
TestManager Domain

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 Log Folder artifacts which in turn contain Log artifacts with actual test results.

Class Hierarchy: Build

SubClasses of Build

This class 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
Iteration	0..1	Iteration	The iteration associated with this Build.
LogFolders	0..n	LogFolder	Log folders that are included with this Build.

Computer
TestManager Domain

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: Computer

SubClasses of Computer

This class 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

This class has no relationships.

ConfiguredTestCase

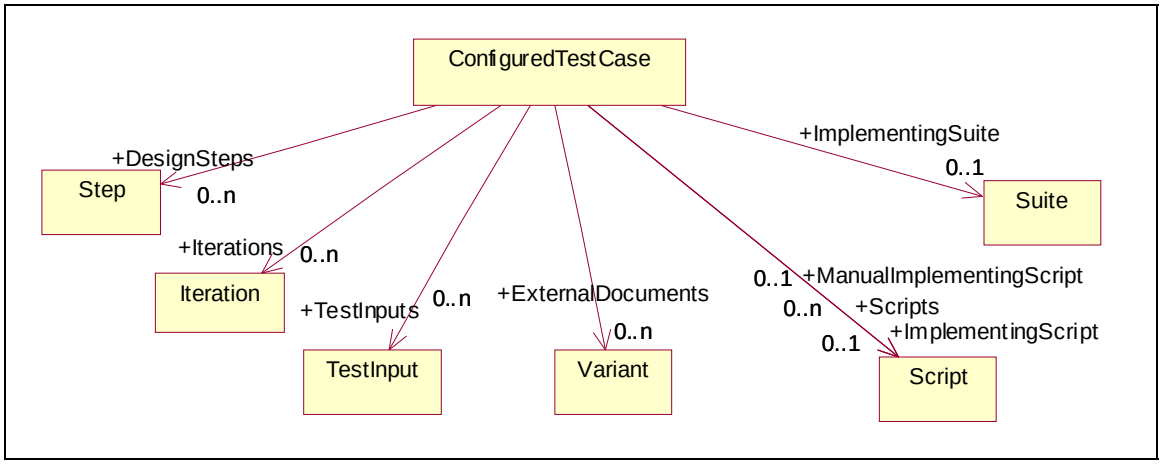
TestManager Domain

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: ConfiguredTestCase

SubClasses of ConfiguredTestCase

This class 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		A custom user-definable value.
Custom2		A custom user-definable value.

Properties	Inherited From	Description
Custom3		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
DesignSteps	0..n	Step	A set of steps that describe how to implement the Configured TestCase.
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.

Name	Kind	Class	Description
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.
ManualImplementingScript	0..1	Script	The manual script that implements this Configured Testcase.
Scripts	0..n	Script	The scripts associated with the ConfiguredTestCase.
TestInputs	0..n	TestInput	Test inputs associated with a ConfiguredTestCase.

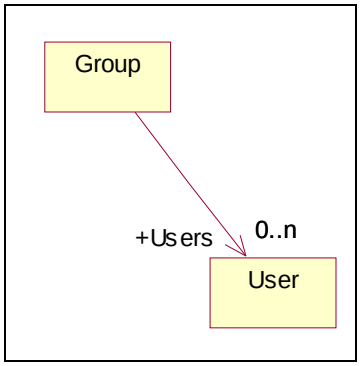
Group
TestManager Domain

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: Group

SubClasses of Group

This class 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
TestManager Domain

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: Iteration

SubClasses of Iteration
This class 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
This class has no relationships.

Log
TestManager Domain

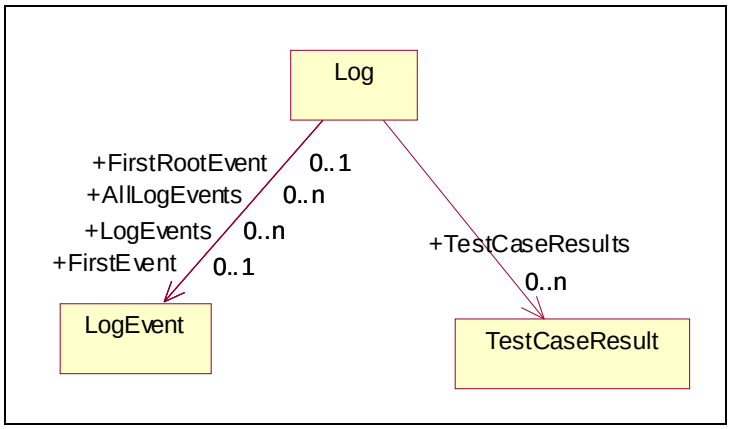
A log is a file that contains the record of events that occur while playing back a script or running a schedule. A log contains the results of all verification points executed as well as performance data. A log contains a collection of Test Case Result artifacts. A log also contains a collection of Load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

SubClasses of Log

This class 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.

Properties	Inherited From	Description
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.

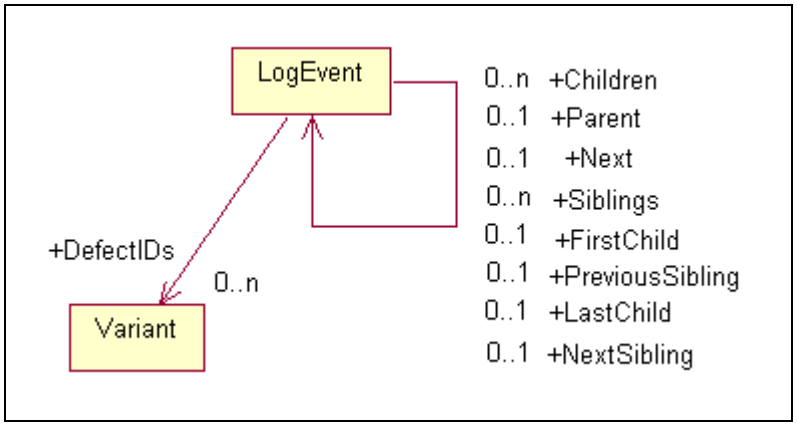
LogEvent
TestManager Domain

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: LogEvent

SubClasses of LogEvent

This class has no subclasses.



Properties Specific to LogEvent

Properties	Inherited From	Description
AdditionalInfo		Text describing any additional information that was available when the log event was created.
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		LogEvent	Associated children log events.
DefectIDs	0..n	Variant	Defect IDs.
FirstChild		LogEvent	First child log event.
LastChild		LogEvent	Last child log event.
Next		LogEvent	Iterates to the next log event.
NextSibling		LogEvent	Iterates to the next sibling log event.
Parent		LogEvent	The parent LogEvent.
PreviousSibling		LogEvent	Iterates to the previous sibling log event.
Siblings		LogEvent	The associated sibling log events.

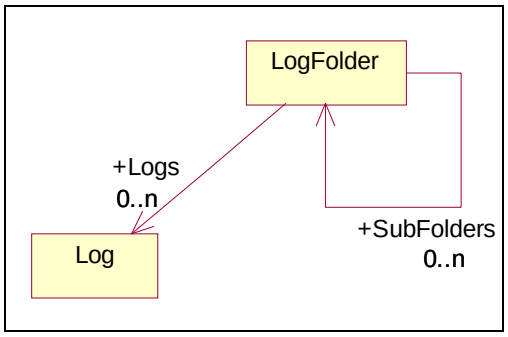
LogFolder
TestManager Domain

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

Class Hierarchy: LogFolder

SubClasses of LogFolder

This class 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
TestManager Domain

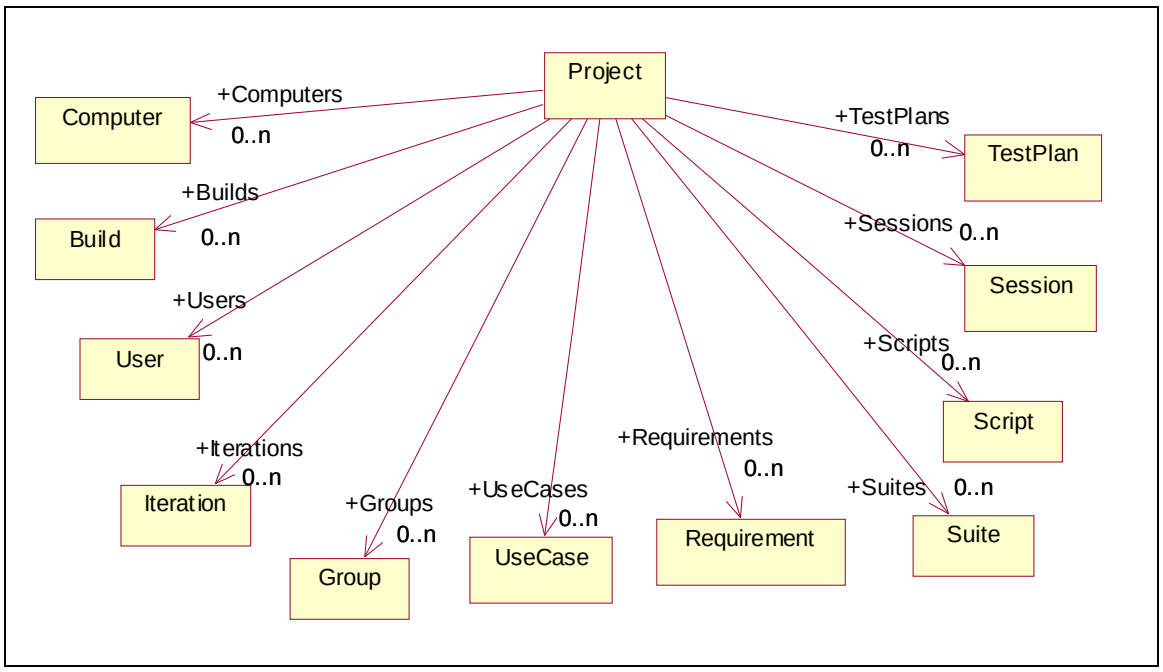
A collection of data, including test assets, defects and requirements, that can facilitate the testing of one or more software components. Projects are managed primarily by the Rational Administrator. Projects contain multiple test plans.

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

Class Hierarchy: Project

SubClasses of Project

This class 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 TestManager 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.
Suites	0..n	Suite	Suites associated with 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
TestManager Domain

Represents a referenced RequisitePro Requirement.

Class Hierarchy: Requirement

SubClasses of Requirement
This class has no subclasses.

Properties Specific to Requirement

Properties	Inherited From	Description
DBName		RequisitePro requirements database name.
FullTag		Unique identifier for the Requirement.
Name		Same as FullTag property.
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
This class has no relationships.

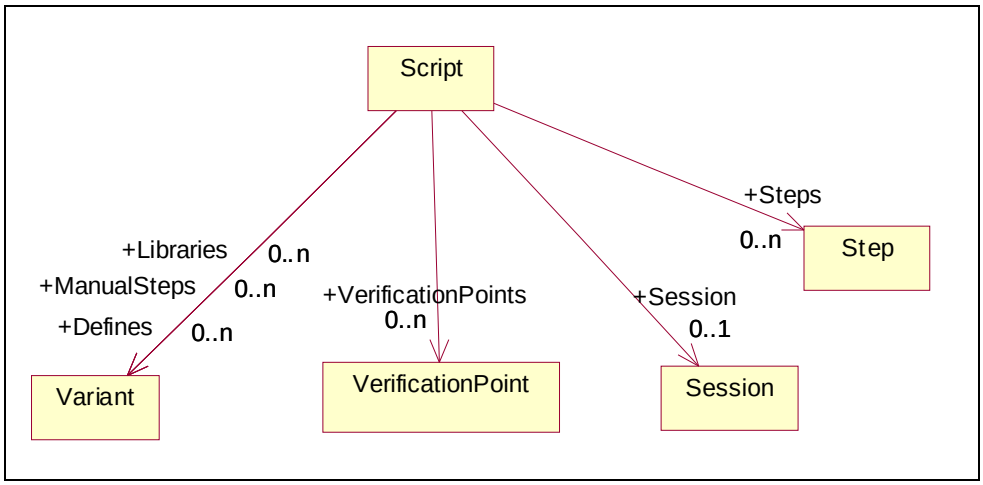
Script
TestManager Domain

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

Class Hierarchy: Script

SubClasses of Script

This class 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		Operating 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).

Properties	Inherited From	Description
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	Manual operations to be performed.
Session	0..1	Session	An associated session to the Script.
Steps	0..n	Step	The numbered steps in the Script.
VerificationPoints	0..n	VerificationPoint	Verification points included in the Script.

Session
TestManager Domain

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

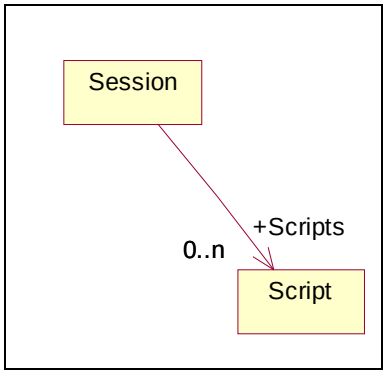
A session is the period of time required to play back a suite. A session thread begins when the first test script of a suite starts execution and ends when the last script finishes.

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: Session

SubClasses of Session

This class 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.

Properties	Inherited From	Description
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.

Step
TestManager Domain

As part of a manual script or test design, a step is an instruction describing one action to take in performing a manual test. Based on the type, a step can also be a manual verification point.

Class Hierarchy: Step

SubClasses of Step

This class has no subclasses.

Properties Specific to Step

Properties	Inherited From	Description
CollIndex		The ordinal position of the step in the collection in which it comes.
Description		The textual description of the step.
Note		Additional information about the step.
Type		Indicates whether this is a step or a verification point.

Relationships Specific to Step

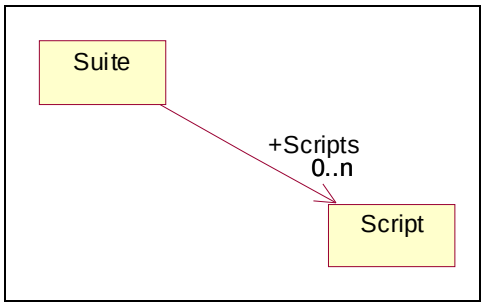
This class has no relationships.

Suite
TestManager Domain

A Suite (TestSuite) is a collection of Scripts.

Class Hierarchy: Suite

SubClasses of Suite
This class 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
TestManager Domain

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.

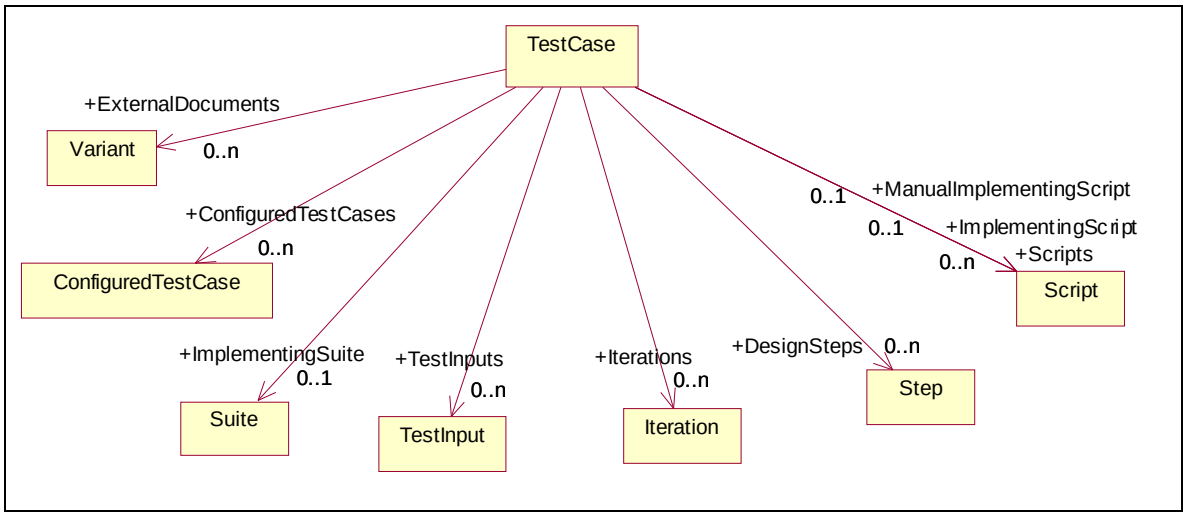
Test cases validate that the system is working the way that it's supposed to work. The test case is the artifact in TestManager that answers the question, "What do I need to test?" You develop test cases to validate particular behaviors. Each test case is owned by or assigned to a team member. This answers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

SubClasses of TestCase

This class 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).

Properties	Inherited From	Description
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.
DesignSteps	0..n	Step	A set of steps that describe how to implement the TestCase.
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 TestCase. If ImplementingSuite returns an object, then ImplementingScript will return NULL - and vice versa.
Iterations	0..n	Iteration	Iterations for this TestCase.

Name	Kind	Class	Description
ManualImplementingScript	0..1	Script	The manual script that implements this Configured TestCase.
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

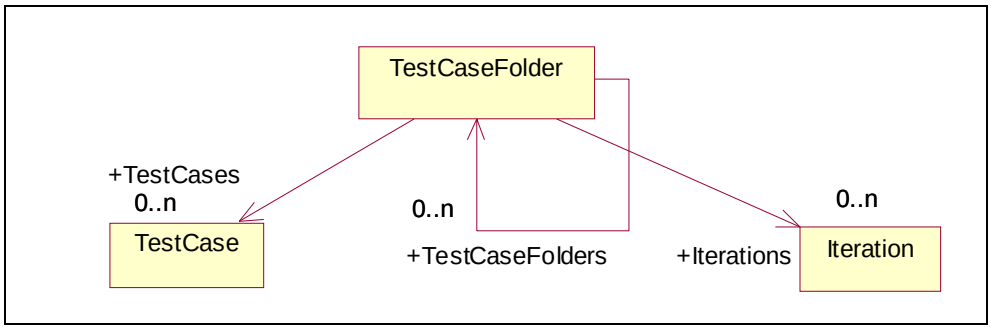
TestManager Domain

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: TestCaseFolder

SubClasses of TestCaseFolder

This class 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		Path of the TestCaseFolder.
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		TestCaseFolder	Associated TestCaseFolders.
TestCases	0..n	TestCase	Test cases contained in this TestCaseFolder.

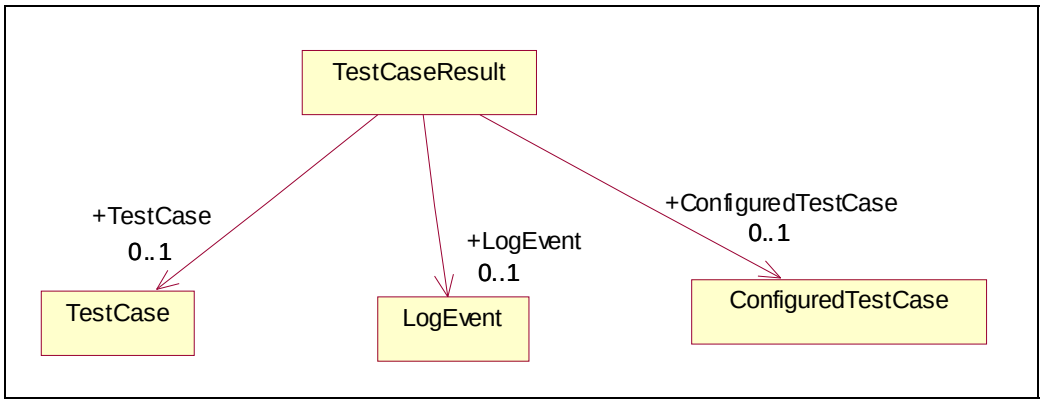
TestCaseResult
TestManager Domain

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

Class Hierarchy: TestCaseResult

SubClasses of TestCaseResult

This class 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.
TestCaseName		Name of the Test Case.
UID		The unique id of the TestCaseResult.

Relationships Specific to TestCaseResult

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

TestInput
TestManager Domain

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: TestInput

SubClasses of TestInput

This class has no subclasses.

Properties Specific to TestInput

Properties	Inherited From	Description
Arg1		A custom user-definable argument.
Arg2		A custom user-definable argument.
Arg3		A custom user-definable argument.
Arg4		A custom user-definable argument.
Arg5		A custom user-definable argument.
CollIndex		Index of TestInput in collection of inputs to parent TestCase.
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		The TestInput subtype.
Type		The TestInput type.

Relationships Specific to TestInput

This class has no relationships.

TestPlan
TestManager Domain

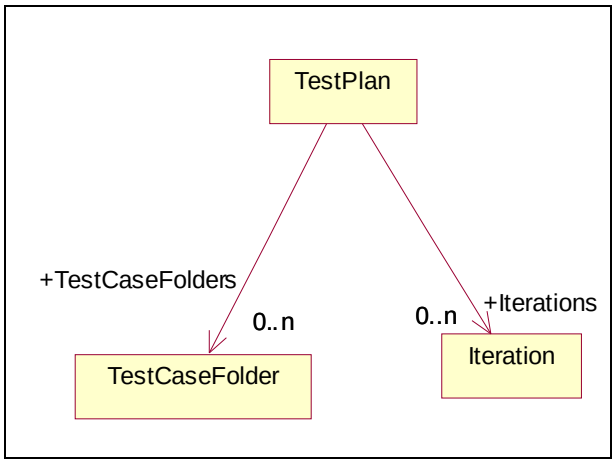
A TestPlan contains information about the purpose and goals of testing within the project, and the strategies to be used to implement and execute testing. Projects can contain multiple test plans.

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: TestPlan

SubClasses of TestPlan

This class has no subclasses.



Properties Specific to TestPlan

Properties	Inherited From	Description
CreatedBy		Creater of the TestPlan.
CreationDate		Creation date of the TestPlan.
Custom1		A custom user-definable value.
Custom2		A custom user-definable value.
Custom3		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
TestManager Domain

Represents a referenced Rose Use Case.

Class Hierarchy: UseCase

SubClasses of UseCase
This class 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.

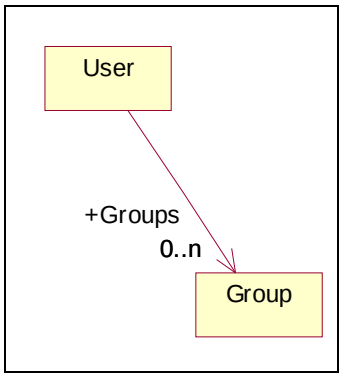
Relationships Specific to UseCase
This class has no relationships.

User
TestManager Domain

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

Class Hierarchy: User

SubClasses of User
This class 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
TestManager Domain

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: Variant

SubClasses of Variant
This class has no subclasses.

Properties Specific to Variant

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

Relationships Specific to Variant
This class has no relationships.

VerificationPoint
TestManager Domain

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: VerificationPoint

SubClasses of VerificationPoint
This class has no subclasses.

Properties Specific to VerificationPoint

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

Relationships Specific to VerificationPoint
This class has no relationships.