

# RequisitePro Domain

## Introduction

The RequisitePro domain lets you extract information from your RequisitePro database. This information can come from projects, requirements, and attributes, among other things.

### Aliases

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

Name in Domain	Name in Interface	Item Type / Location
Members Groups	Users SecurityGroups	Relationship / Project Class
Level	HierarchicalLevel	Property / Requirement Class
AttrValues Discussion RelativeParent ParentReq ChildReqs	Attributes Discussions Parent ParentRequirement ChildRequirements	Relationship / Requirement Class
RelationshipType	Type	Property / Relationship Class
Responses DiscussionRequirements DiscussionUsers DiscussionGroups	Replies AssociatedRequirements Participants ParticipantGroups	Relationship / Discussion Class
Response	Reply	Class
HasResponses	HasReplies	Property / Response Class
DefAttrPermissions DefDocTypePermissions DefListItemPermissions DefProjPermissions DefReqTypePermissions	DefAttrRights DefDocTypeRights DefListItemRights DefProjRights DefReqTypeRights	Property / Group Class
ReqDocument	Document	Class
TemplateDesc	TemplateDescription	Property / DocumentType Class

### Accessing Project-specific Objects

The RequisitePro integration is project specific. Although you see default RequisitePro object information (such as Documents, Requirements, and Attributes) when you design templates, templates do not have knowledge of the project-specific object names unless you are connected to the RequisitePro project.

To use project-specific attribute values in a template, you must provide the path and name of the project file in the OPEN command. This triggers the retrieval of all project-specific objects such as Requirements Types, Document Types, and the Attributes Names for each Requirement Type. This is known as a *dynamic domain*.

## Improving Generation Performance of RequisitePro Templates

For optimal performance, it is important to note which types of queries are supported by the RequisitePro query engine. Using these whenever possible results in the best performance.

- Fast filtering is currently only supported for requirements retrieved for a project.
- The best way to improve performance is to narrow your REPEAT to a particular requirement type. This is done by selecting one of the requirement type-specific classes in the “Where is A” box in the REPEAT command dialog box. This restricts the results to only requirements of that type. Note that this requires that the dynamic domain is loaded when the template is built; otherwise, this selection option is not available.
- If you are specifying additional filter criteria by adding to the “And Where” section in the REPEAT command dialog box, adhere to the following rules for faster queries:
- Use only the "AND" logical operator in your queries (no "OR" queries).
- Use user-defined attributes. Since these are specific to each requirement type, you must select a requirement type-specific class in the “Where is A” box to make these available.
- Reference any of the following attributes within your query. These are all the attributes for the **requirement** class:

Attribute	Operators supported for “Fast” filtering	Description
TagPrefix ReqType->ReqPrefix	=	Search by requirement type
FullTag	=, !=	Search by full tag of a requirement
Text	=, !=, <, <=, >, >=, LIKE <sup>1</sup>	Search by requirement text
Document.Name	=, !=	Search by the name of the containing document.
LatestRevision->Number	=, !=, <, <=, >, >=, LIKE <sup>1</sup>	Search by revision number
HierarchicalLevel	=, !=, <, <=, >, >=	Search by requirement level
LatestRevision->DateTime	=, !=, <, <=, >, >=	Search by the date of the last revision
Bookmark	=, !=, <, <=, >, >=, LIKE <sup>1</sup>	Search by bookmark

<sup>1</sup>These pattern matching metacharacters are supported for fast filtering:

- . (dot)
- \$ (dollar)
- ^ (not)

See “Metacharacters for LIKE” in the Help for information.

# AttributeValue

## ReqPro Domain

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

**Class Hierarchy:** AttributeValue

### SubClasses of AttributeValue

This class has no subclasses.

### Properties Specific to AttributeValue

<b>Properties</b>	<b>Inherited From</b>	<b>Description</b>
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

This class has no relationships.

# Discussion

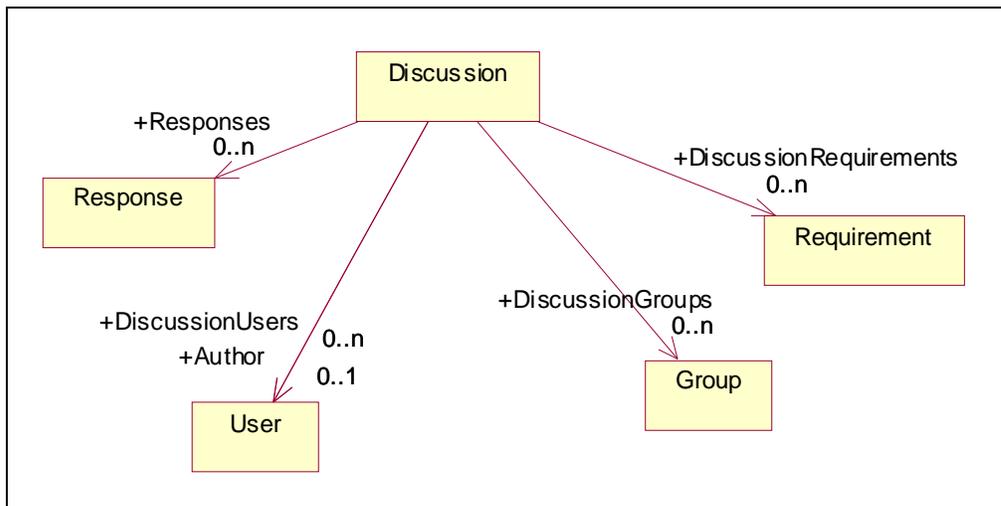
## ReqPro Domain

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:** Discussion

### SubClasses of Discussion

This class 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

<b>Name</b>	<b>Kind</b>	<b>Class</b>	<b>Description</b>
Author	0..1	User	User who created this discussion.
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

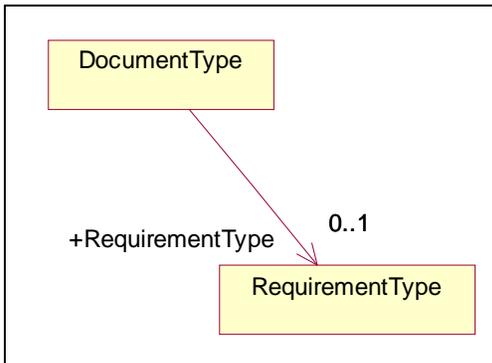
## ReqPro Domain

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:** DocumentType

### SubClasses of DocumentType

This class 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.

## ExtRequirement

### ReqPro Domain

A requirement that resides in an external project.

**Class Hierarchy:** Requirement>ExtRequirement

#### SubClasses of ExtRequirement

This class has no subclasses.

#### Properties Specific to ExtRequirement

Properties	Inherited From	Description
Bookmark	Requirement	Name of the Word bookmark associated with this Requirement.
DocPosn	Requirement	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	Requirement	Full tag of the requirement, such as PR1.
GUID	Requirement	The GUID for the Requirement.
HasChildren	Requirement	True if this requirement has child requirements; otherwise False.
HasParent	Requirement	True if this Requirement has a parent requirement. This would be the same as Level > 0.
Level	Requirement	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	Requirement name.
TagNumber	Requirement	Number of the tag, such as 1.
TagPrefix	Requirement	Prefix of the tag, such as PR.
Text	Requirement	Text of the requirement.
UniqueID	Requirement	The unique ID for the Requirement.

#### Relationships Specific to ExtRequirement

This class has no relationships.

## Group

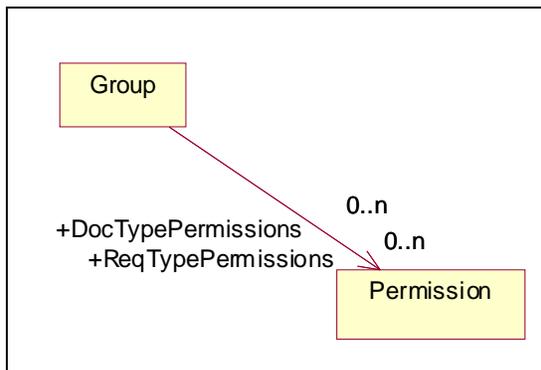
### ReqPro Domain

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:** Group

#### SubClasses of Group

This class 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 to modify requirement types.
GroupID		Group ID.
Name		Name of the user group.
ProjectPermissions		Project permissions of the Group.

**Relationships Specific to Group**

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

# Package

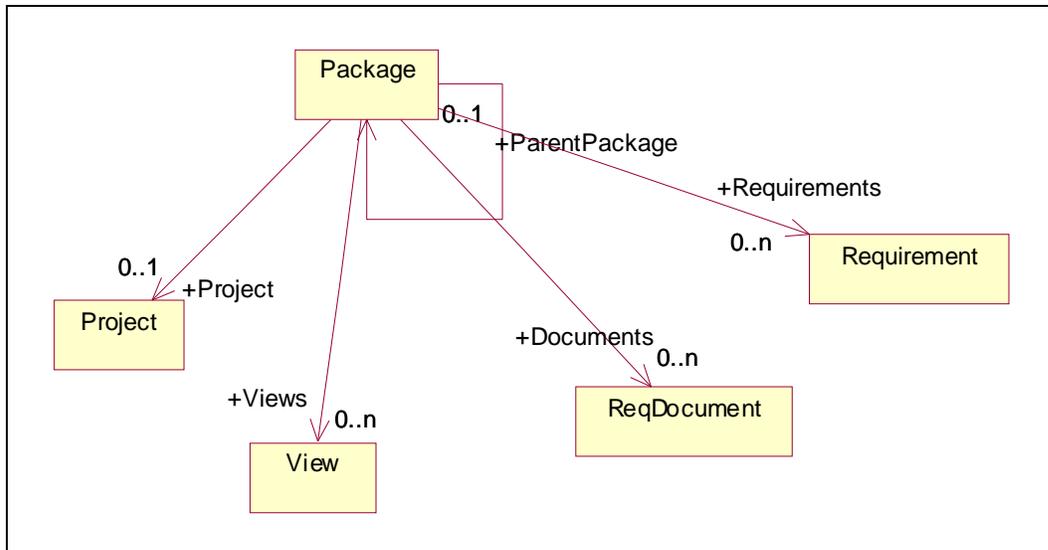
## ReqPro Domain

A container, represented in the RequisitePro explorer as a folder, that can contain requirements, documents, views, and other packages. A package organizes project artifacts.

**Class Hierarchy:** Package

### SubClasses of Package

This class has no subclasses.



### Properties Specific to Package

Properties	Inherited From	Description
Description		The description of the package.
GUID		The GUID for the package
Name		The name of the package
UniqueID		The unique database ID that identifies the package.

### Relationships Specific to Package

Name	Kind	Class	Description
Documents	0..n	ReqDocument	Documents associated with the package.
ParentPackage		Package	Parent package associated with the package.
Project	0..1	Project	The project associated with this package.
Requirements	0..n	Requirement	All requirements associated with this package.
Views	0..n	View	All views in the package..

## Permission

### ReqPro Domain

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:** Permission

#### SubClasses of Permission

This class 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

This class has no relationships.

# Project

## ReqPro Domain

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:

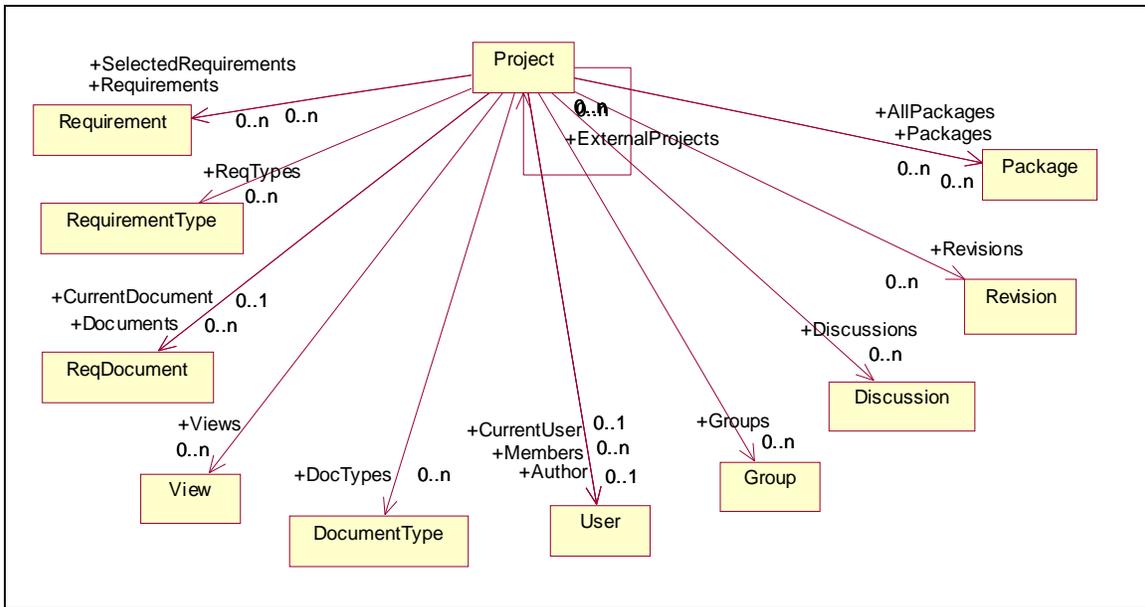
- Database
- Documents
- Document types
- Requirements and their attributes
- Requirement types
- Requirement traceability
- Discussions
- User and group security

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

### Class Hierarchy: Project

#### SubClasses of Project

This class 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.

<b>Properties</b>	<b>Inherited From</b>	<b>Description</b>
Path		Path of the Project.
Prefix		Prefix that is prepended to requirement tags when using external Projects.

### Relationships Specific to Project

<b>Name</b>	<b>Kind</b>	<b>Class</b>	<b>Description</b>
AllPackages	0..n	Package	All packages in the Project, including packages contained in other packages, recursively.
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	The User currently working with 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		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.
Packages	0..n	Package	Packages associated with this project or Package, but not including packages contained in those packages.
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

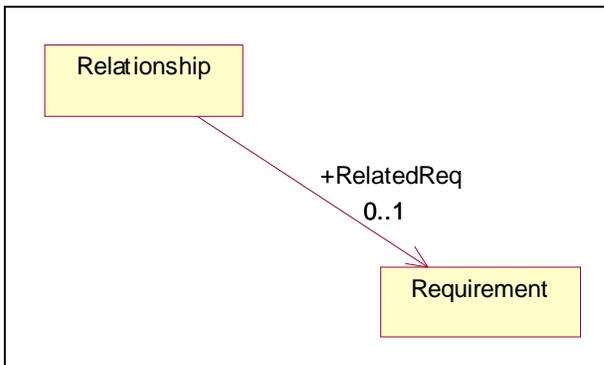
## ReqPro Domain

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

**Class Hierarchy:** Relationship

### SubClasses of Relationship

This class 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 Domain

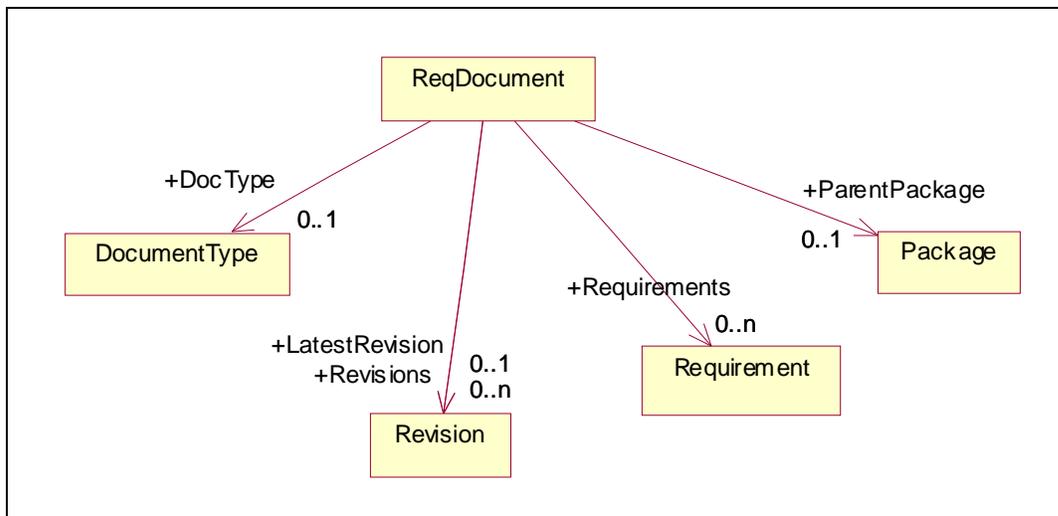
A requirements document created in Microsoft Word or Rational RequisitePro that captures requirements and is used to communicate product development efforts. Each requirements document addresses a particular requirement type, such as product requirements, software specifications, or test plans.

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

### Class Hierarchy: ReqDocument

#### SubClasses of ReqDocument

This class 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 to the directory containing the ReqDocument.

**Relationships Specific to ReqDocument**

<b>Name</b>	<b>Kind</b>	<b>Class</b>	<b>Description</b>
DocType	0..1	DocumentType	Document type associated with a ReqDocument.
LatestRevision	0..1	Revision	Latest Revision artifact associated with a ReqDocument.
ParentPackage	0..1	Package	Parent package associated with a ReqDocument.
Requirements	0..n	Requirement	Requirements associated with a ReqDocument.
Revisions	0..n	Revision	Revision artifacts associated with a ReqDocument.

# Requirement

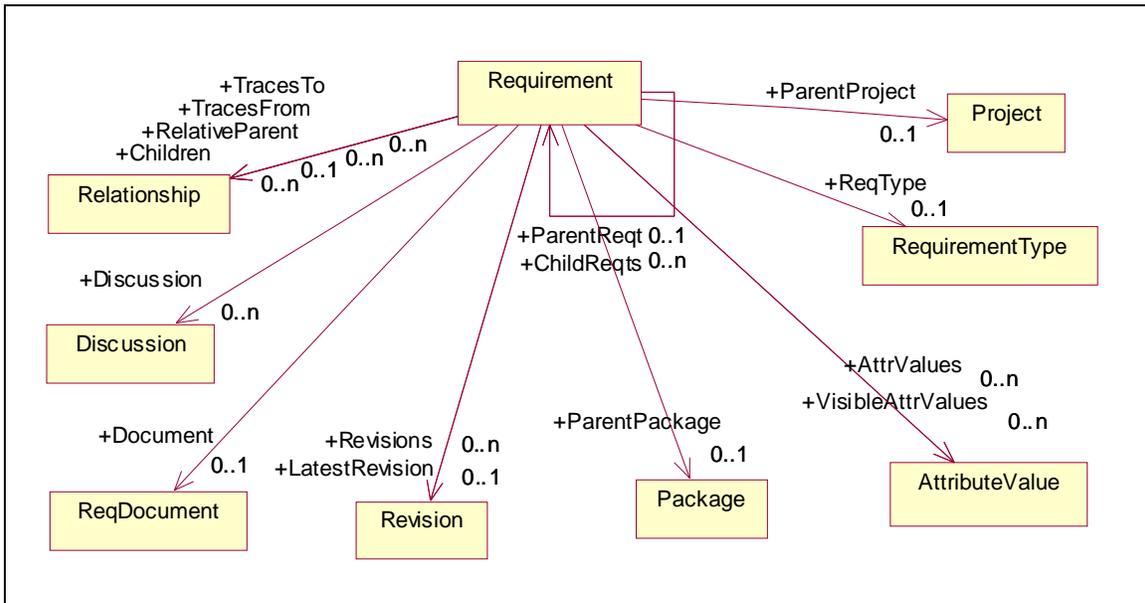
## ReqPro Domain

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: Requirement

#### SubClasses of Requirement

ExtRequirement



### 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.

<b>Properties</b>	<b>Inherited From</b>	<b>Description</b>
FullTag		Full tag of the requirement, such as PR1.
GUID		The GUID 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

<b>Name</b>	<b>Kind</b>	<b>Class</b>	<b>Description</b>
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 ChildReqs relationship.
ChildReqs		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.
ParentPackage	0..1	Package	Parent package associated with a Requirement.
ParentProject	0..1	Project	Project that this Requirement is contained in. This relationship is especially useful when doing cross-project traceability.

<b>Name</b>	<b>Kind</b>	<b>Class</b>	<b>Description</b>
ParentReq		Requirement	Parent Requirement artifact for this Requirement.
RelativeParent	0..1	Relationship	Parent relationship for a given Requirement. The relationship refers to the parent Requirement artifact returned by the ParentReq relationship.
ReqType	0..1	RequirementType	The type of this requirement.
Revisions	0..n	Revision	The collection of revision artifacts for this 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 Domain

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:** RequirementType

### SubClasses of RequirementType

This class 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

This class has no relationships.

## Response

### ReqPro Domain

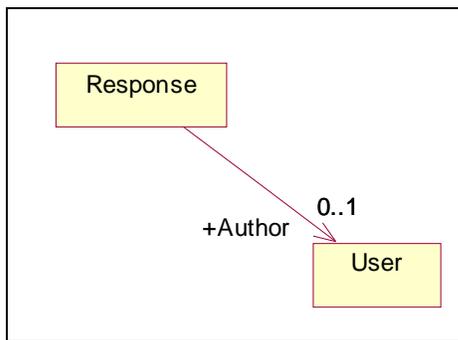
An answer to a discussion item. You can respond to the initial discussion topic or to other, previous responses.

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

**Class Hierarchy:** Response

#### SubClasses of Response

This class 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		Concatenation of the full key of the parent and the key of this response.
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.

# Revision

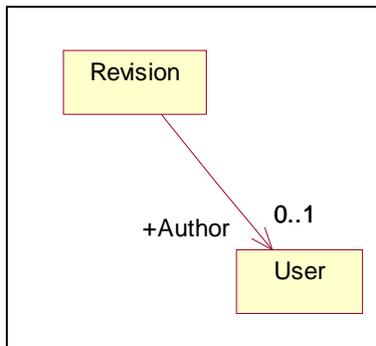
## ReqPro Domain

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:** Revision

### SubClasses of Revision

This class 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**

<b>Name</b>	<b>Kind</b>	<b>Class</b>	<b>Description</b>
Author	0..1	User	User that made the change.

# User

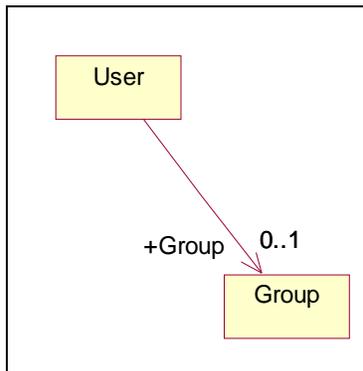
## ReqPro Domain

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:** User

### SubClasses of User

This class 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

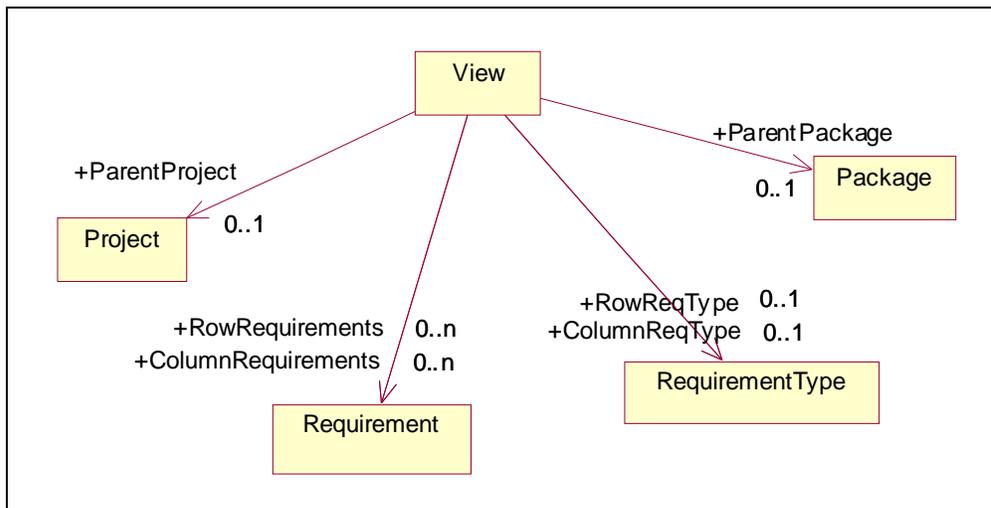
## ReqPro Domain

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:** View

### SubClasses of View

This class 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**

<b>Name</b>	<b>Kind</b>	<b>Class</b>	<b>Description</b>
ColumnReqType	0..1	RequirementType	Requirement type of a column in a View.
ColumnRequirements	0..n	Requirement	Requirements in the columns of a View.
ParentPackage	0..1	Package	Parent package associated with 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.