

Manage your requirements with RequisitePro

Skill Level: Introductory

[Martin C. Brown \(questions@mcslp.com\)](mailto:questions@mcslp.com)

Programmer

04 Jun 2004

Software is based on requirements -- the requirements of your clients, team, marketing department and other entities, all of whom send you requests about the features that they would like incorporated into the system. But how do you track all of this information and use these requirements to help manage your project? In this tutorial you'll see how to turn the requirements you receive into a project within RequisitePro, which enables you to track the requirements information and use this to produce project plans and requirements documents.

Section 1. Before you start

About this tutorial

One part of the software development cycle that is often overlooked is requirements. These requirements take many different forms, including the requests of users and management, the features and functionality of your competitor's products, and the needs and requirements demanded by your development team to achieve the requirements specified.

If you are the project manager or lead developer of software products then you are probably involved in the specification and management of the requirements during the life of the entire project. Managing and collating that information can be time consuming, unless you have a specialized tool like RequisitePro® to help you with the process of requirements specification and management. RequisitePro is a tool that enables you to specify the requirements and features of an application before the application starts to be developed. This saves time, money, and usually a lot of frustration by giving your developers an initial definition of the application prior to the

actual coding process.

RequisitePro provides a database for storing and tracking requirements. You can add requirements to the database through a dedicated project management-like interface, or you can populate the database while writing the requirements specification within Microsoft Word. This latter method enables you to continue using your familiar requirement specification documents and templates, and even to generate the requirements from within Word, while gaining all the benefits of a managed database of requirements.

In this tutorial you'll see the basic mechanics of the requirements process and how RequisitePro can aid the specification and tracking process. The various requirements are covered in the following sections:

- Plan a software project
- Transfer your requirements into RequisitePro
- Organize project information
- Create requirements within RequisitePro
- Project documents
- Reports and views
- Manage requirements during a project

Prerequisites

To complete the steps in this tutorial, you will need a copy of RequisitePro. [Download a trial version.](#)

You will also need a full working copy of Microsoft® Word 2000, XP (Word 2002) or Word 2003 to get the best benefits from the system.

Section 2. Plan a software project

Starting a project

Let's start by looking at what a requirement really is, how it affects the development process, the effects of the project plan, and how this information can be translated

into the way RequisitePro works.

Software projects get started in a host of different ways: a formal request from a client or another department, a request for a new piece of software to help you in your work, an existing application is split into a number of smaller projects to make them easier to handle.

There are lots of issues to consider when starting the project. You need to select the development team, the development environment languages, and other components. Most important of all though, you need to define what the software is going to do once you have finished the project.

But where do you start? Well, most people start by defining a list of features. It can be short, such as "build a new word processor" or it can be a detailed blow-by-blow account of all the features and functionality. These requirements are, like the original project specification, probably sourced from a number of different groups (or stakeholders) within your organization.

RequisitePro was designed for collating these requests and managing the requests as they go through the development life cycle. It turns a list of requirements from all the different sources and types them into a coherent requirements specification that can be used to initiate the project and, during the development stages, monitor and track the progress of the development compared to the original specification and/or changing requirements. You can even use this requirements specification to develop a project plan through the built-in integration with Microsoft Project.

What is a requirement?

If you look at a formal definition, a requirement is a capability that a system must provide. A typical program can be made up of a number of different requirements, from a few to a few hundred thousand. With any project the most important step is to realize that even a simple application has a number of requirements. For example, if you look at the infamous 'Hello, World!' program you could define its requirement as follows: "To display the string 'Hello, World!' on the screen."

In reality the requirement is rarely that simple. I've always thought that it's possible to define the requirements for an application using a chaos theory. One of the portions of chaos theory is related to the geometry of nature and how the granularity of your view affects what you see.

For example, if you were to measure around the coast of the British Isles using a granularity of 1 mile, you'd arrive at a reasonable approximation of the length of the coast. But it wouldn't be precise because you'd average out all of the twists and turns that were smaller than a mile. Reduce the length to a yard, and you would arrive at a larger value, but a more precise one. Measure it in inches, and again

you'd get a larger value, but possible not a true one.

The same is true with the requirements for an application. Any language can be used to display the simple "Hello, World" message above. What hasn't been discussed about the message above is whether it displays in a window, and if it is in a window, what the window looks like, and if it has any buttons, and what do they look like.

Collating and managing requirements

Its important when building any application that you build the correct application, meaning that the people involved in initiating and requesting the application in the first place end up with the one they expected. In addition, you need to create a specification for a project that enables you to develop the application correctly -- that is, you build in the ability to easily extend, manage and modify the product as the requirements change.

To build the correct application, you must first collect all the requirements from the different people involved in the project. This could include customers, end users, and your marketing department. You might also want to consider the input from partners involved in the overall project or system, and it's always a good idea to incorporate some of the functionality and features of your competitor's products.

Once you have all of the requirements, you need to manage and control them to ensure that they are effectively applied throughout the development of your system. This means tracking their effectiveness and status and prioritizing the requirements so that your developers are devoting their time and energies to the right requirements.

It is also critical to realize that requirements are not static elements -- you don't generate requirements at the beginning of a project and then never review the contents. In the middle of a project it's likely that someone suddenly realizes the potential for a new feature, or that someone recognizes that an existing requirement doesn't quite support all of the functionality required.

Your requirements therefore become a 'living' document of the features of the project -- not a monolithic (and often tiresome) definition.

Using requirements in project management

Many projects fail, stutter, miss deadlines, or completely fail to live up to user expectations because the definition and tracking of the requirements for the system have not been correctly recorded and tracked.

Many factors contribute to these failures. For example, a key deadline can be missed because priorities weren't followed. Alternatively, the needs of the users have changed since the inception of the project in the original requirements specification. The changes to the requirements and the tracking and history of those changes are as important to the requirements specification as the original document.

Getting to know RequisitePro

RequisitePro enables you to collect and manage all of the requirements for a system, track where the requirement request originated, give a requirement a priority and add information useful to developers and project managers about the difficulty and relative workload for each of the requirements in the system.

Use RequisitePro during the development of your application so that you can continue to track the original requirements and their progress during development and add and expand upon those requirements as the project matures.

RequisitePro consists of the main RequisitePro application that provides the core functionality required to build a requirements specification. Behind the scenes, RequisitePro makes use of a database to store all of the requirements information and to track changes and updates to the information.

RequisitePro comes with a version of the Access Jet database built-into the system, but you can also use a RDBMS (Oracle or SQL Server) that is more practical in larger teams or when dealing with larger, more complicated, requirements specifications.

All projects are also securable through a standard user/password combination and this security authorization is based on a separate, project specific authorization database.

RequisitePro and Microsoft Word

In addition to this data store, RequisitePro integrates directly with Microsoft Word. You can create requirements from within a Word document, automatically populating the database within a RequisitePro project. These documents and their contents are tracked and stored along with the project, with changes also being recorded into the project's database. These documents are what you would normally produce during the course of a requirements specification, but when created or imported within RequisitePro they allow you to create the project and integrate the information between the two systems directly.

This integration means that you get all of the control of a database, with the flexibility of an open structure and content in the documentation you produce, with all the

requirement information being automatically updated between the two. For example, if you change the requirement text, the corresponding requirements document that you produce automatically reflects these changes; there is no need to manually update your requirements document with the changes.

RequisitePro is also multi-user aware. Multiple members of the development and project teams can open individual projects so that requirements, progress, and information can be updated. You can also elect to open projects in read-only mode -- useful when you only want to study the information or produce reports -- and exclusive access, which can be used for maintenance or project initiation.

There are numerous different document types provided with RequisitePro that can be included within a RequisitePro project. You can also create and add any document to a project and create your own document types and templates. In each case, the documents are Word documents, with templates being supported through standard Word templates. The Word integration also includes a special toolbar within Word to enable you to embed information from the database into a document directly within Word without having to go through a re-generation process.

Standard outlines and templates for the different documents include:

- **Vision statement:** shows the main characteristics, major features and key requests.
- **Glossary of terms:** describes the different elements within the system.
- **Requirements specifications:** includes sub types for the different requirement types such as traditional, Use-Case and supplementary.
- **Test plans:** helps develop tests for your requirements. Use this to develop test plans within other application such as Rational® TestManager.

This last template also highlights the other integration features built into RequisitePro, as well as using Microsoft Word directly as part of the requirements management process. The information generated can also be used to help you work with other components from the Rational suite.

RequisitePro can also integrate with Microsoft Project if you need to match requirements and development needs into a new or existing project plan as part of a wider system development process.

Requirement types in RequisitePro

Before you look at a specific example of detailing requirements into RequisitePro, consider the type of requirements you are going to track in the system.

RequisitePro supports customizable requirement types, but also includes definitions for the two most common types of requirement: Traditional and Use-Case:

- Traditional requirements follow the format that you might already be using. A simple declaration of a feature or operation within the application. For example, the earlier "Hello, World" requirement was an example of a traditional requirement.
- Use-Case requirements are more familiar to those people who use other Rational products such as ClearQuest and Rational Rose and ultimately familiar with the Unified Modeling Language (UML). Use-Case works on the basis of an example sequence of events that generate a specific result. These are particularly useful for defining functional software systems that use object-based technology.

RequisitePro enables you to create projects based either solely on one of these two types or create a composite document that supports both types. Depending on the type of project you are creating, you also need to populate different types of document within the project. These documents are seen separately as you go through the tutorial.

It doesn't matter -- from a learning perspective -- which system you want to use. The basic principles of the requirements management process apply in both cases. For the purposes of this tutorial, use the traditional format.

Now, let's translate a sample software application project into its requirements, using RequisitePro.

Section 3. Transfer requirements

Recipe database application

To help you through the process of getting to know RequisitePro, you need to populate a sample RequisitePro project with requirements for a sample application. For the purposes of this tutorial you're going to specify the requirements for a Recipe Database.

The overall aim of the application is to allow a user to store recipes in a database. This requires a number of core requirements that have been supplied to you from a client. The main requirements from the original request are listed below:

- The user should have the ability to enter a recipe into the database.
- A recipe consists of the recipe name, a number of servings, a sub title/description, a list of ingredients and a list of methods.
- The user should be able to search for a recipe by name, description, or ingredients.
- The user should be able to change the number of servings for a recipe and have the ingredient quantities recalculated.

The client in this case is a stakeholder in the project. Stakeholders are those people who have contributed features and requirements into the system and include all the parties who are interested in the project. These include your potential users, the project team, and management. Your competitors do not directly provide requirements, but the specification and features of your competitor's projects will influence the features and specification of your own project.

Finally, the developers involved in your project will also supply their own list of requirements and functionality, based on the needs of the rest of the stakeholders. Take the Recipe database application, for example. The fact that you need to store structured information almost certainly implies that you need to create a database suitable for storing the recipe data. This is a developer requirement, identified by the requirements of others.

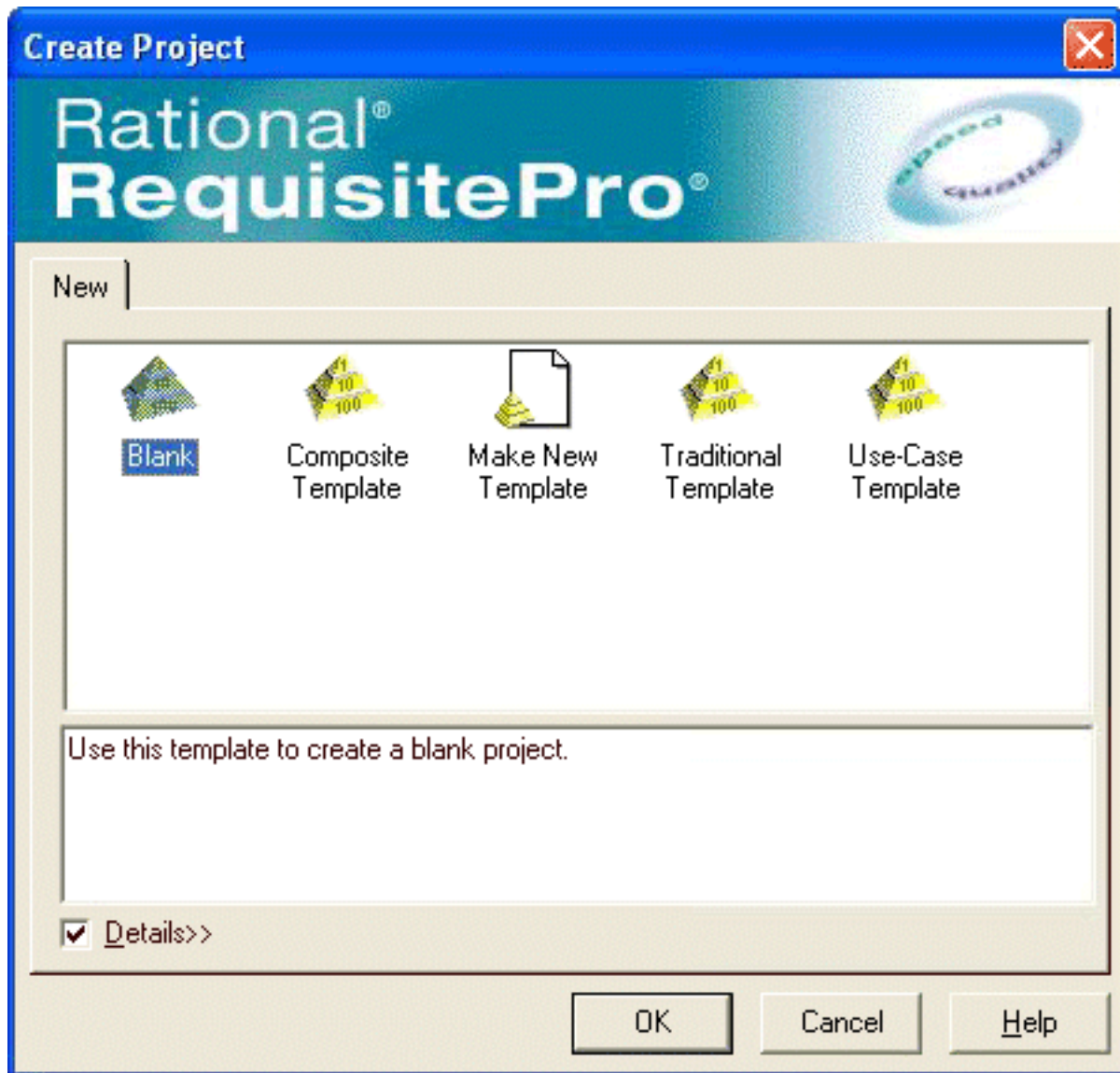
Let's start building your project.

Creating a new project

Creating a new project is similar to creating a new document in any other application. Create a new project to store the requirements for the Recipe database application.

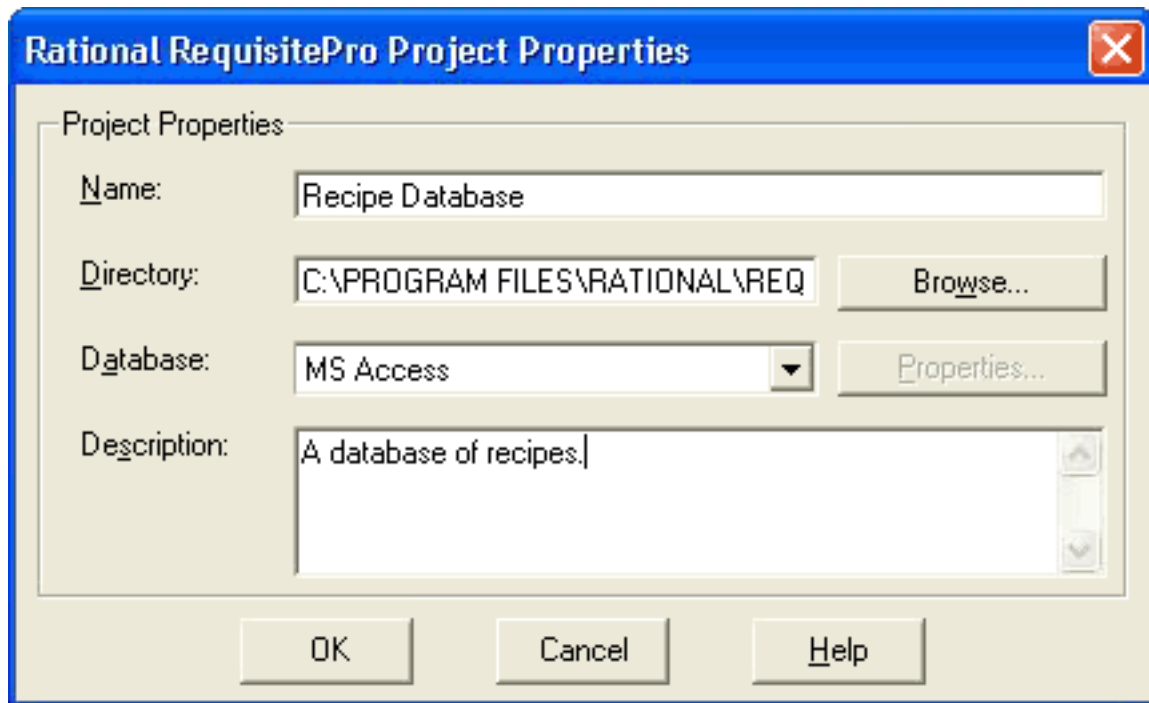
Open Rational RequisitePro and click **File > New**. A list of available project templates, as shown in Figure 1 is displayed.

Figure 1. Choosing a project template



Blank projects allow you to create a structure suitable for your needs, including allowing you to define your own requirement types, documents and packages. The Traditional and Use-Case templates create a standard set of folders and documents applicable to these two requirement types, while a composite template provides a structure that supports the two primary requirement styles. Descriptions of each of the template types and their contents are provided in the details box at the bottom of the window. You can also create a new template. For this project, you're going to use the Traditional Template. Select that template and click **OK**. The Project Properties window opens as shown in Figure 2.

Figure 2. The Project Properties window



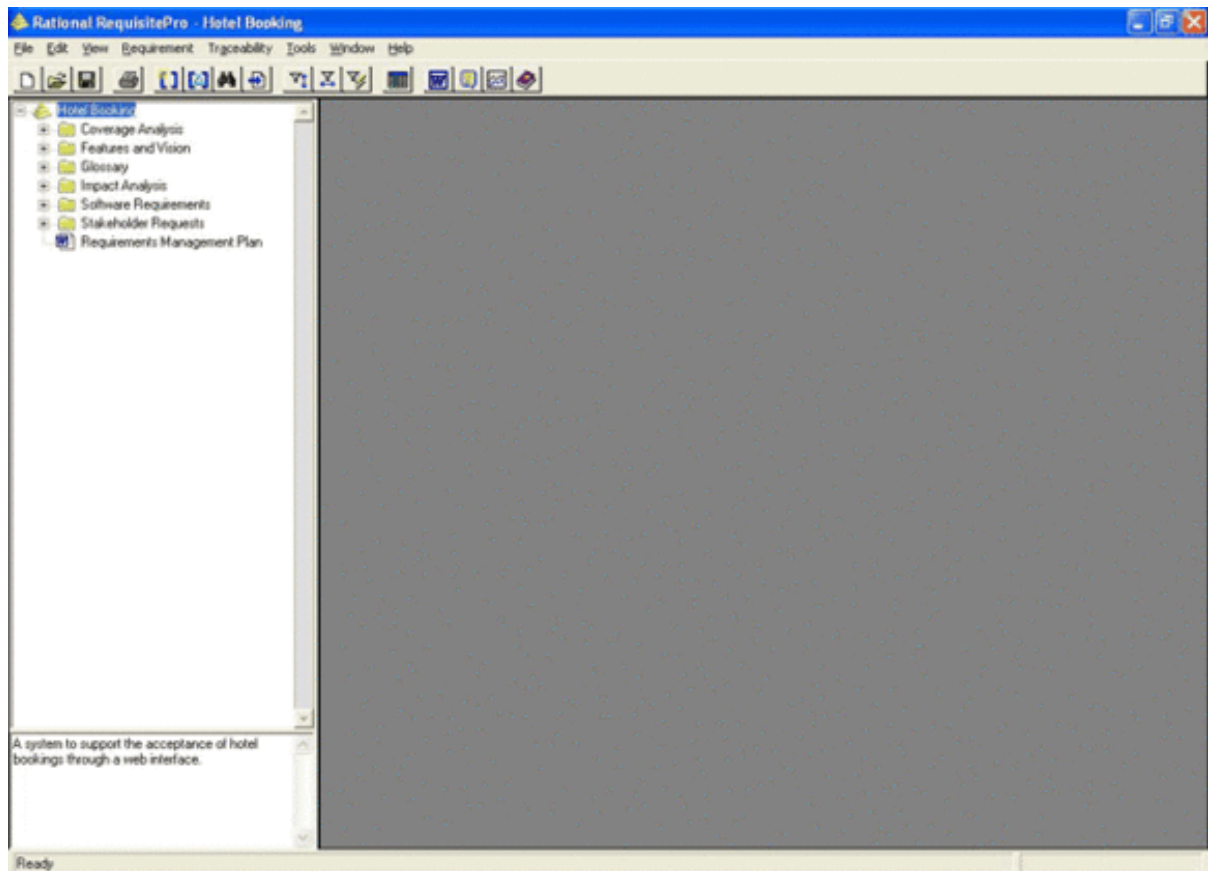
Enter the name of the project, select a directory, select a database, and optionally provide a description. Use the built-in Microsoft Access database. This is also the location where you would select a suitable SQL Server or Oracle solution. Click **Properties** to define the login and database identities.

Packages

Once the project template has been copied into your new document, a window similar to the one shown in Figure 3 below appears. All of the items in the tree view are expanded on the left. Each of these is called a package and it logically groups documents and requirements to help you organize your project.

For example, you'll be using the Software Requirements package to help store the primary list of requirements for the project, but you're also going to create a package to hold requirements generated by developers as they start to develop the system. The specifics of the various stages required in creating the package and the developer requirements are covered as you work through the rest of the tutorial.

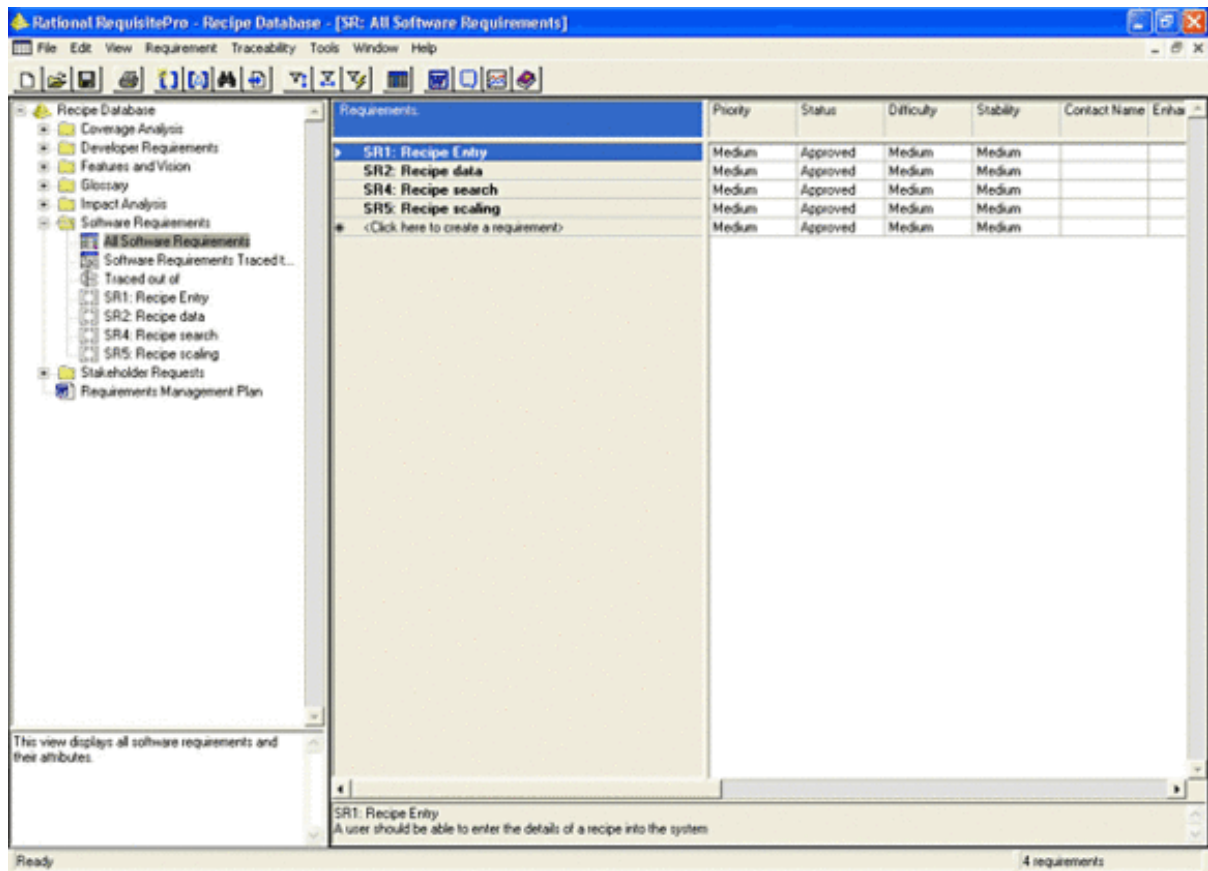
Figure 3. A new project



RequisitePro Explorer window

Let's take a closer look at the main project window within RequisitePro. The figure below shows a document with the software requirements section expanded.

Figure 4. The main RequisitePro Explorer window



The usual menu bar is at the top and a button toolbar beneath it. Hover over the buttons to get tool tips. This section describes the buttons to create a new requirement, how to show the properties for a requirement, and how to search for a requirement and a button to open the document in which the requirement is specified in detail. The next batch provides filtering tools (queries), a topic discussed in the Views section of this tutorial.

The main part of the window is split into three panels. The top-left panel shows the structure of the project. The 'root' of the structure is the main project title. The rest of the structure works just like a standard tree view of a directory, as in Explorer. In fact, the basic structure of the whole project follows the same structure as a directory and can consist of a combination of documents and further folders. To open any of the items, as opposed to the folders, double-click on the information.

The panel at the bottom-right shows summary information for a package, requirement or document. The large panel on the right side of the window details the requirements list, alternative view or query on this information.

The core data you store in the project is the requirement -- all of the other information in the document is typically either based on this information, or provides an alternative view on the information.

Section 4. Organize project information

Project packages

The first stage to documenting your requirements is to organize your project and create the necessary packages that help you to manage the requirements in your system. For the sample project, one of the pre-existing project templates, which incorporates a number of standard packages, requirement types, and associated Word documents that make up a typical RequisitePro project was used. The basic structural component of a project is the package.

The packages within a RequisitePro project are used to organize the different components in the project. Packages are really just a logical grouping for the different components within the system. They allow you to organize these components and to create queries and views that generate links between these items. For example, in the template there is a Stakeholder Requests package and a main Software Requirements package. These software requirements can be traced back to one of the stakeholder requests.

Packages can contain one or more of three different types of information:

- **Requirements** -- the main data type in the project.
- **Documents** -- based on a template, a bare document or the requirements list.
- **Views** -- views, based on separate queries, of the list of requirements.

The packages are important because it is through the combination of package and the requirements that you are able to track the different requirement types. This is often used to help organize the sources of feature requests, client requests, stakeholder requests, and developer requirements based on the attributes of the requirement or its' requirement type.

For example, in the traditional project template there are three packages used to record different requirement information:

- **Software requirements** -- the final list of requirements that are used for the project.
- **Features and vision** -- the list of features for the project.

- **Stakeholder requests** -- the list of requests for features from stakeholders.

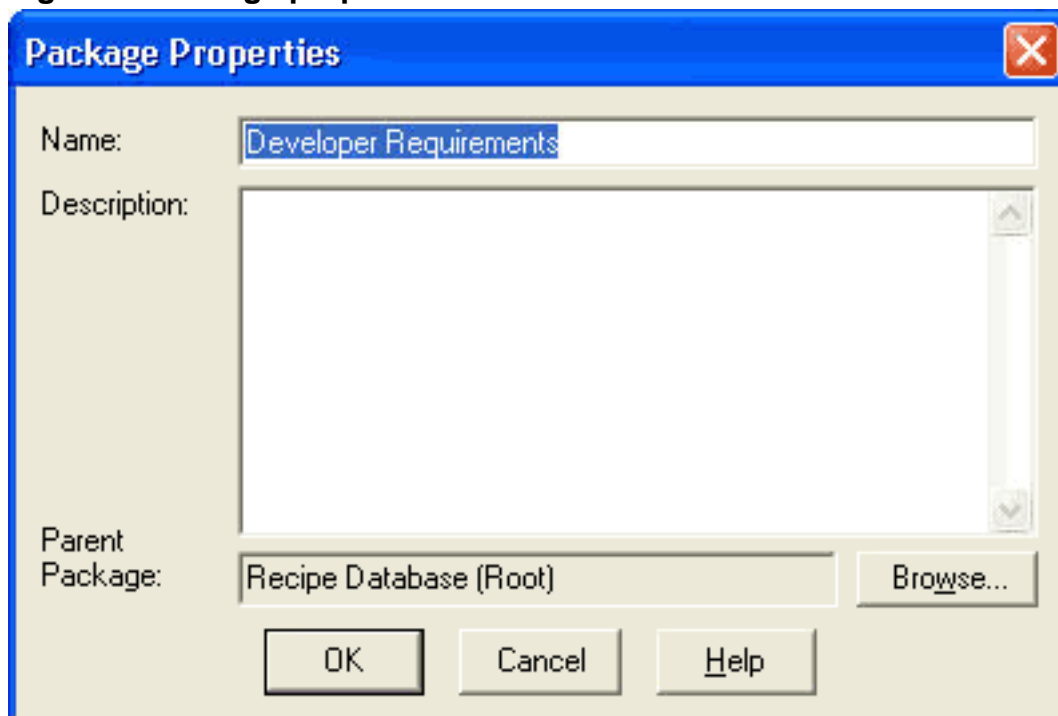
There's also a package used to record the glossary of terms to be used when referring to components of a project, called Glossary.

Creating a new package

To create a new package to help track developer requirements in this project:

1. Right-click in the explorer and select **New > Package** or select the same option from the File menu. A window opens, as shown in Figure 5.
2. Give the package a name. You're creating a Developer Requirements section so enter `Developer Requirements` into the Name field. In the Description field, enter `Requirements specified by the developer based on stakeholder and user requirements.`

Figure 5. Package properties



The Parent Package defines the location of this package within the overall project. Because a package is technically just another name for a folder, nest packages within other packages or the root projects, just as you can nest folders within folders on a file system. Create this package in the root of the project because you'll be

tracking developer requirements across the whole gamut of requirements in the project. You can create packages within each of the existing requirements packages to hold developer requirements related to each parent -- i.e., developer requirements within the software requirements -- and then use a separate view at the root of the project to collate all the requirements into one list. Views and how to query and display information are described later.

Now that you have a developer requirements package, it's time to start adding requirements into the project. There are two ways to do that: through the RequisitePro explorer interface or while creating a specification within Word. The former method is described first.

Section 5. Create requirements

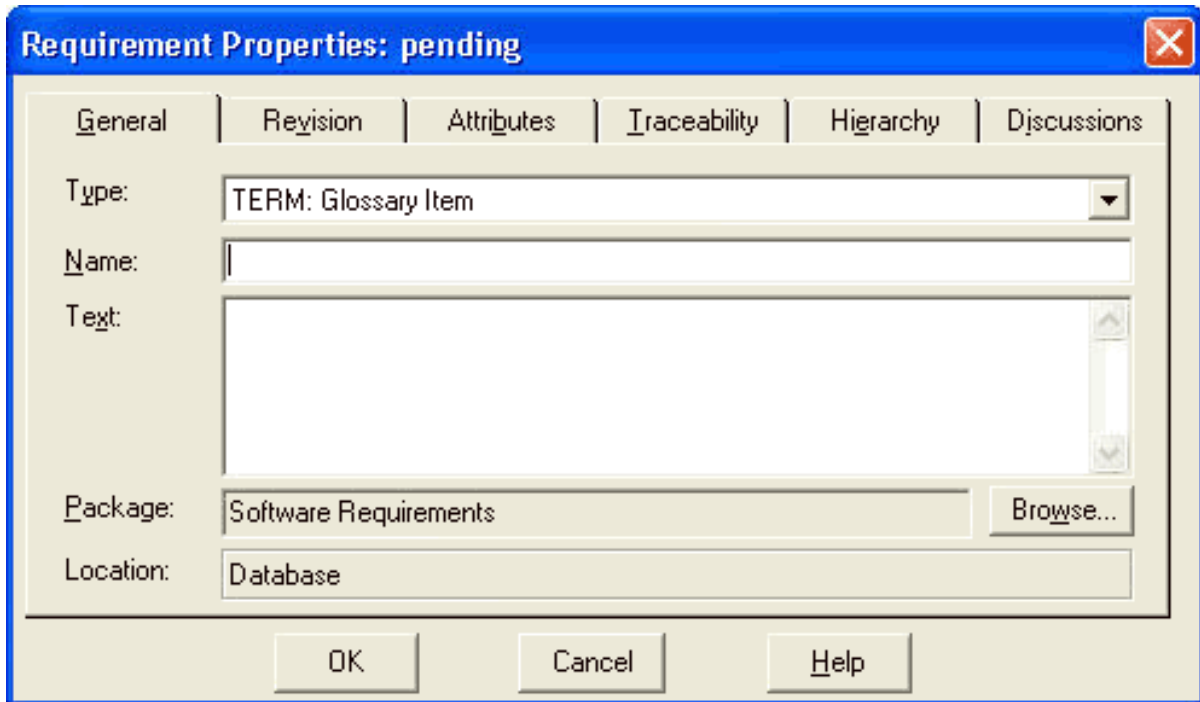
Creating requirements within RequisitePro

Start by building a list of requirements. The steps below show you how to add the first of the recipe database requirements to the project; add the other three using the same method.

To create the first recipe database requirement:

1. Right-click within the tree and select **New > Requirement**, or select the same menu option from the main File menu. A window like the one in Figure 6 opens. Populate the fields with information from the requirements introduced earlier in this tutorial.
2. Specify the requirement type. This is a notional marker that highlights the requirement as a specific type but has no bearing on where the requirement is stored. The type affects what attributes are stored with the attribute and ultimately that information is used in queries and views to link and trace the sequence and source of information through the system. For the purposes of this walk through, use the Software Requirement type.
3. Give the requirement a name. This should be enough to identify the item within the system. Enter `Recipe Entry` in the Name field. You can also attach a more detailed description in the Text field. Cut and paste the text `The user should have the ability to enter a recipe into the database into this field.`

Figure 6. Requirement properties



All requirements are recorded within a folder. This can either be the root folder (not generally recommended) or one of the packages. Because you are creating basic software requirements at this stage, click **Browse** and select the Software Requirements folder.

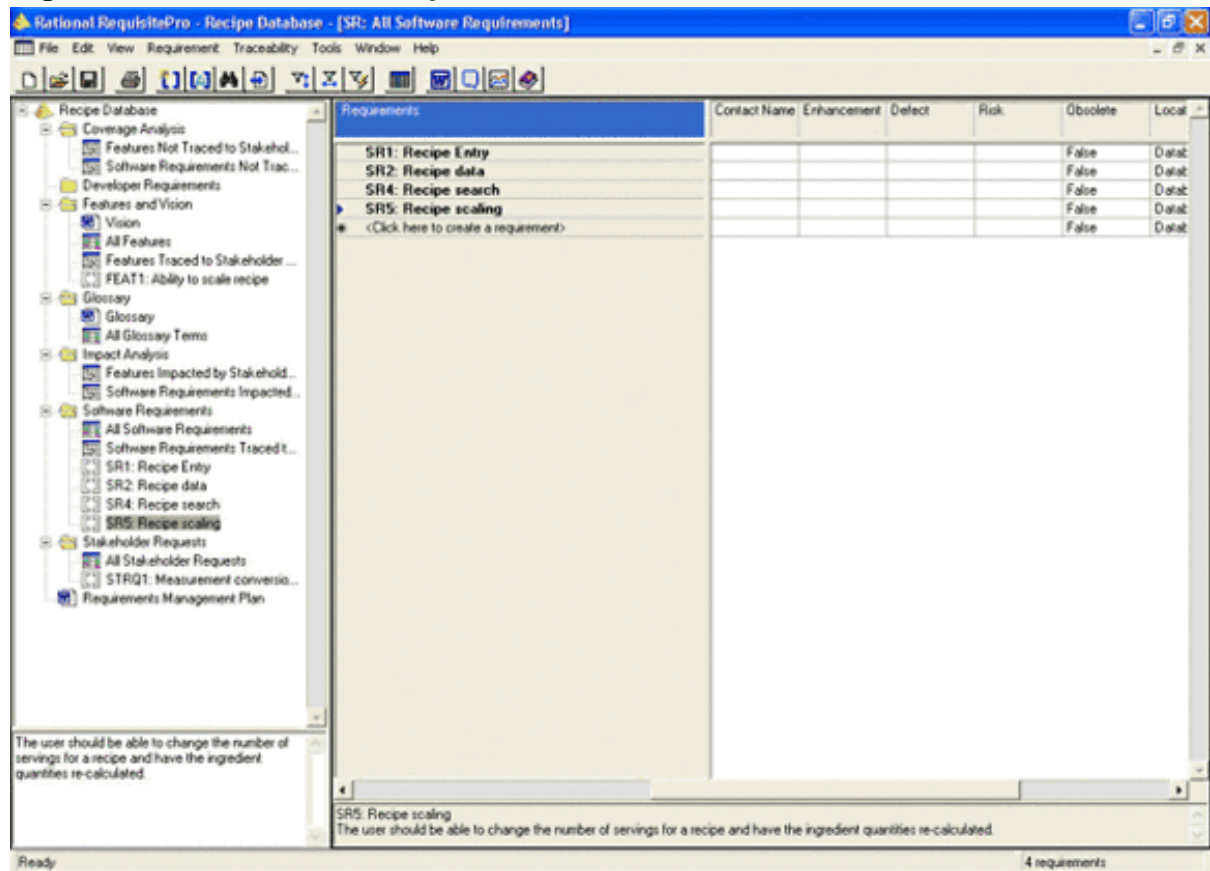
Adding further requirements

Repeat this sequence with the remaining requirements from the list, as outlined in this table. All the requirements should be stored within the same Software Requirements folder and all should be of the Software Requirement type.

Name	Text
Recipe data	A recipe consists of the recipe name, a number of servings, a subtitle/description, a list of ingredients and a list of methods.
Recipe search	The user should be able to search for a recipe by name, description or ingredients.
Recipe scaling	The user should be able to change the number of servings for a recipe and have the ingredient quantities re-calculated.

The list should look like the one shown in Figure 7.

Figure 7. The final list of requirements



You can see from the figure that each requirement is given a unique reference number. These numbers are used to help track and trace requirements through the system. In the above figure, the numbers are not sequential because a mistake was made in an earlier example. This is perfectly normal and nothing to worry about; there is no way to change the numbers or "fill-in" the gaps.

Alternative requirement types

The project template you are using includes a number of pre-defined requirement types. Requirement types define the attributes and other information that is stored with a requirement in the database.

Typically, you create a different requirement type within a project according to the needs of the project. For example, you might want to individually track standard software requirements, stakeholder requests, and user requests within the system. You can also use requirement types to enable the system to track other types of data. For example, the Glossary requirement type is used to define terms for different components in the system. Strictly it's not a requirement, but the information

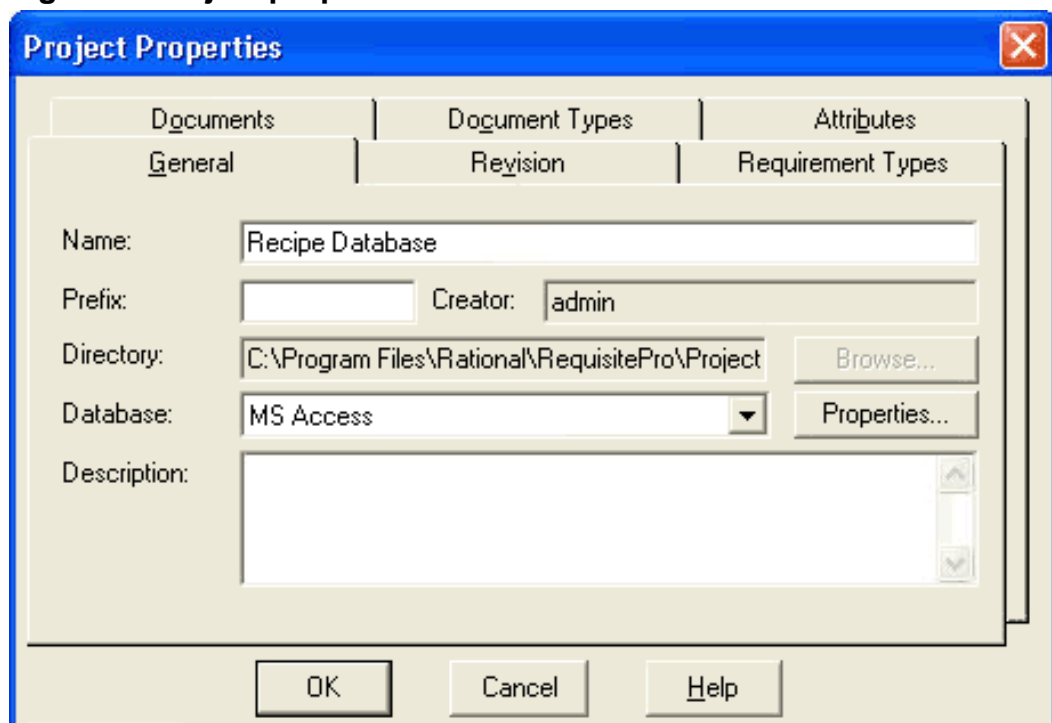
is stored within the requirement database. Because you can create different requirement types, it doesn't matter how the information is stored -- just whether you can recover the information.

All requirements have a prefix that is used to help identify the requirement when it is created. For example, Software Requirements typically have the prefix SR and requirements are numbered sequentially according to this prefix. When you created the first recipe database requirements earlier, you should have created requirements with IDs like SR1, SR2 and so on. You'll use these when you start managing the project.

Requirement types are configured on a project-by-project basis or within a project template. To add or change the type definitions:

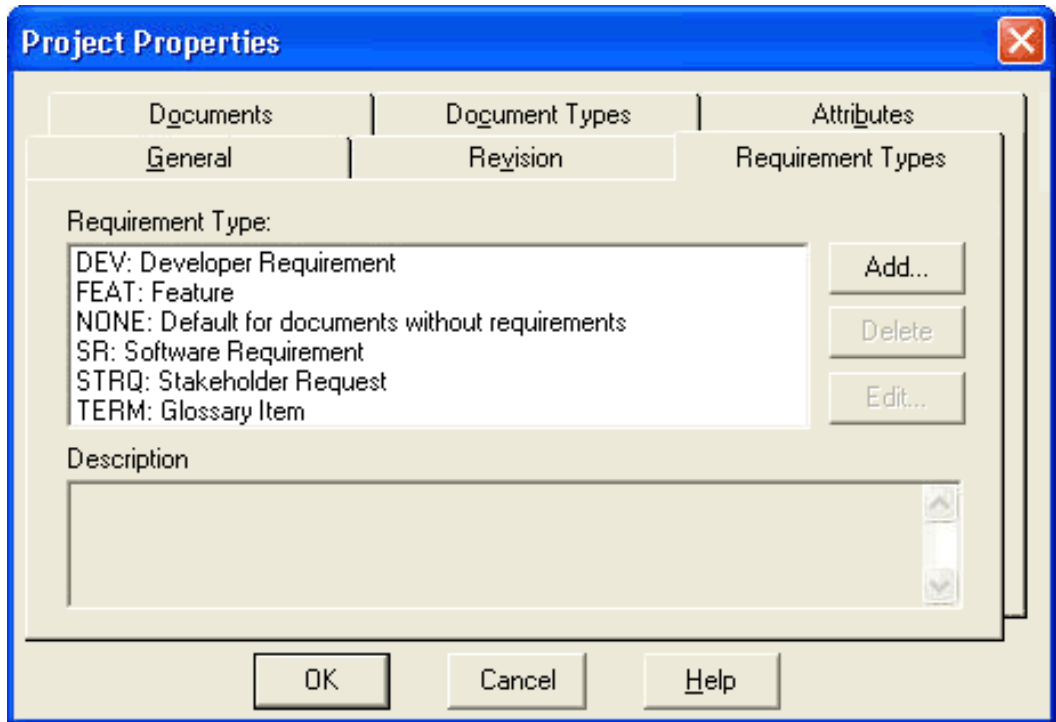
1. Open the properties for the project either by right-clicking on the project and choosing Properties or by double-clicking on the root project. A project properties window appears like the one shown in Figure 8.

Figure 8. Project properties



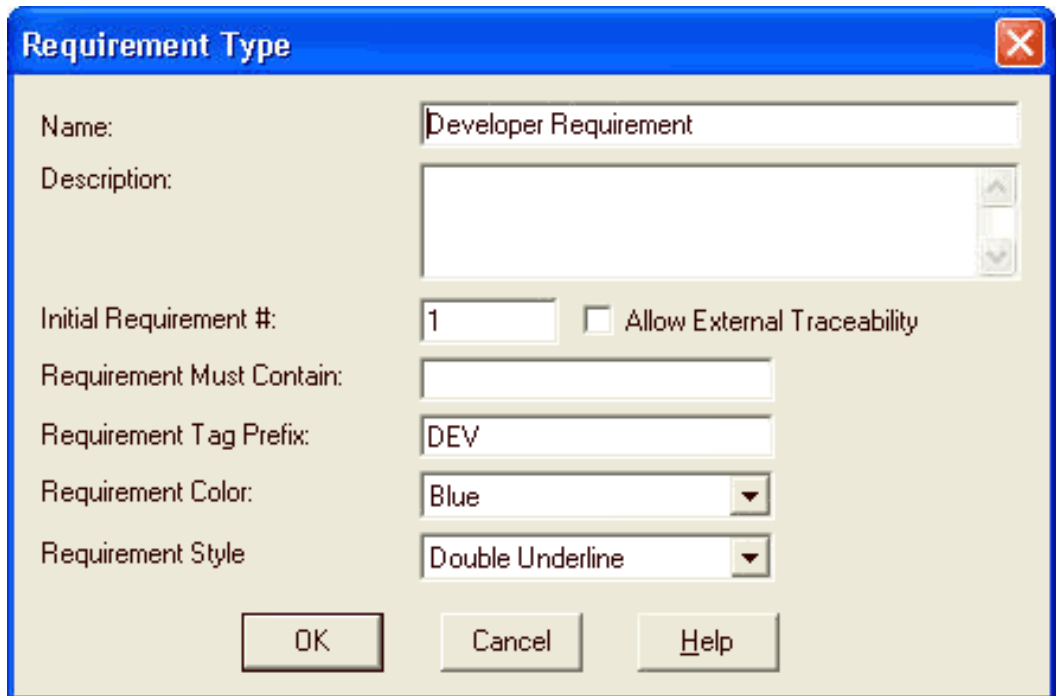
2. Click the Requirement Types tab.

Figure 9. Requirement Types tab



- To add a new type, click the **Add** or edit an existing type. You can see from Figure 9 above that a Developer Requirement type has been added. The basic properties for this type are shown in Figure 10 below.

Figure 10. The Requirement Types tab in Project Properties



4. Duplicate the information in this window to create the Developer Requirement type. The color and style are used when viewing the requirements in reports and documents so you can use this to help identify the type.

Requirement attributes

Requirements in the system also have a number of additional attributes that are used during the management of a project. These help to further define the requirement in the system so that you can use them to sort, order, and prioritize requirements during the life of the project. In nearly all cases these attributes have their own pop-up lists of suitable values. Most of these items should be self-explanatory and as with other components of the system the actual list of attributes and values are highly configurable.

For reference however, following is a list of those attributes defined within your project for the Software Requirement:

- **Priority** -- High, Medium or Low.
- **Type** -- the type of requirement (usability, performance, etc).
- **Status** -- whether the requirement has been approved or incorporated into the project.
- **Difficulty** -- a rough guide to the difficulty that would be experienced to achieve the requirement.
- **Stability** -- a guide to the stability of the requirement within an active project.
- **Risk** -- a guide to the risk associated with implementing the project. For example, there might be a high developmental risk if the requirement would require changes to other parts of the system (thereby reducing stability).
- **EnhancementRequest** -- the information sourced from a Rational project if this RequisitePro project is part of a Rational project.
- **Defect** -- generated by a Rational project during defect management/testing.
- **Contact Name** -- for the source of the requirement.
- **Obsolete** -- an indication of whether the requirement is obsolete.

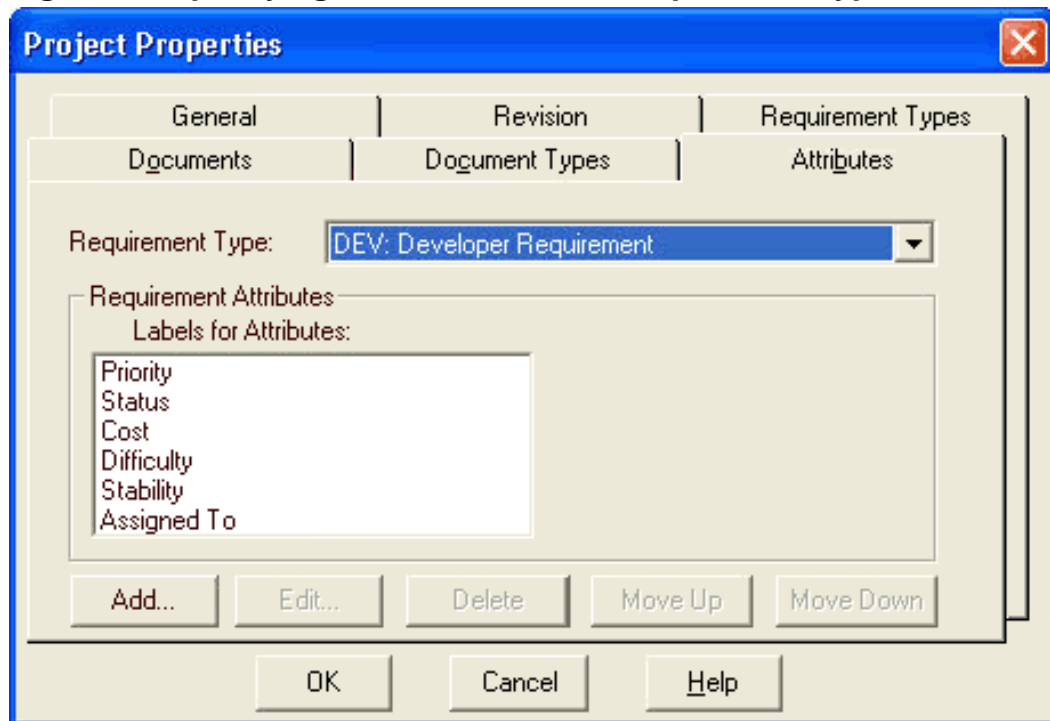
To adjust the attributes assigned to a requirement type, open the attributes tab of the

project properties.

To help with your project, add a development team field to the Developer Requirement type:

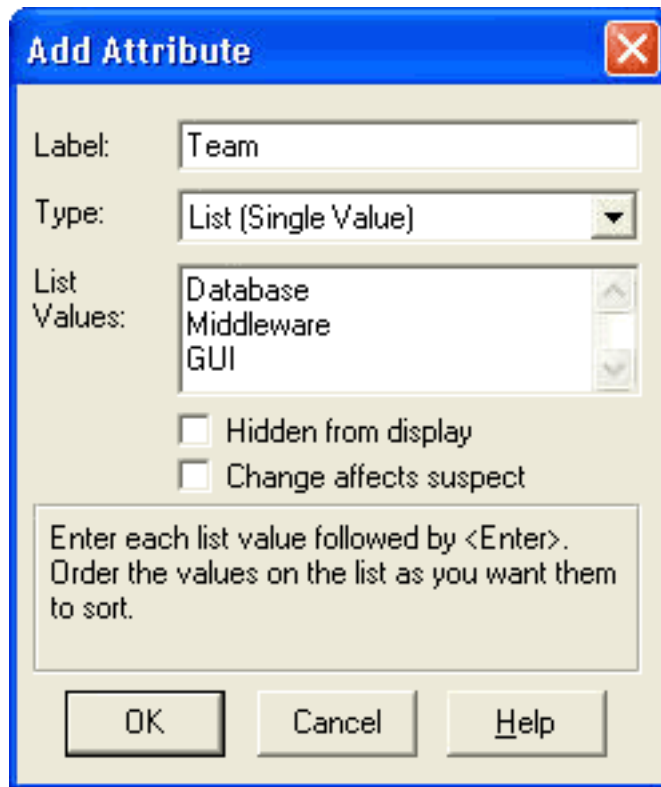
1. Open the project properties.
2. Change to the Attributes tab (shown in Figure 11) and then select the Developer Requirement type from the window. A list of the attributes currently defined for the type is shown in Figure 11.

Figure 11. Specifying the attributes for a requirement type



3. Click **Add**. Attributes have a label (the field name you'll populate when you create a requirement), a type, and an optional list of values that are used within a popup or list. The type is a field type just as in a database and can be an integer, floating point, text string, date, time, URL or a list, either single or multiple selection. Select the List (Single Value). Populate the list using the information shown in Figure 12. Use Return to separate the values.

Figure 12. Adding an attribute to a requirement type



Additional requirement properties

The remaining properties for a requirement are used during the management of a project. You'll see more detail on those in the discussion of managing a requirement project within RequisitePro.

The other tabs in the requirements properties include:

- Revision: shows the history of changes to a requirement
- Traceability: governs the relationship between requirements
- Hierarchy: shows which maps the parents and/or children of a requirement
- Discussions: enables you to discuss changes and requirements with other members of your team using the information embedded into the RequisitePro project.

Before you move on to the management and reporting facilities within RequisitePro, look at the alternative (and optional) method of creating requirements from within Word while creating a requirements specification.

Section 6. Project documents

Documents in a project

Let's set aside the recipe database discussion for a moment and take a look at the significance of documents within RequisitePro. These are an optional component of the system that some people find more convenient to work with than the pure database structure of the Explorer interface.

Use RequisitePro to manage and control the requirements on an existing project. Most project teams use Word to create their requirements specification as a standard Word document, but cutting and pasting the specification from the Word document into RequisitePro can be time consuming. Even for new projects, some people can be more comfortable with using Word than the heavily regimented structure of the RequisitePro system.

For this reason, RequisitePro allows you to create requirements in the internal RequisitePro database from within Word through the use of a special Word extension accessible through a toolbar and/or menu.

As with other parts of the system, RequisitePro includes a number of standard document types and templates that can be used to specify and document different types of information in the project. Also, all documents that are included as part of the project are subject to the same revision history system as requirements stored in the database.

Built-in templates exist for some key requirements components. There's a Vision template used to define the original problem, overall vision, and a definition of the stakeholder needs and features. There are also templates for the Requirements Management plan that can be used to set up the parameters of the project, the people involved, and how the definition and development of the requirements are developed.

RequisitePro includes a number of requirement specification templates that you can use to build your requirements specification. These documents provide a structure which, in combination with the integration between Word and RequisitePro, enables you to create requirements both in a Word document and in the requirements database. This section describes the use of these documents and the other parts of the Word integration process.

Creating requirement documents

If you select to generate the requirements in Word, it is probably a good idea to generate and record all requirements directly within Word, although it's certainly not a requirement. This is because once the link between the Word document and the Rational database has been created, further updates to either the database or the Word document will result in updating the corresponding information. For example, if you change the description text for a requirement in the RequisitePro explorer, the information is updated in the Word document.

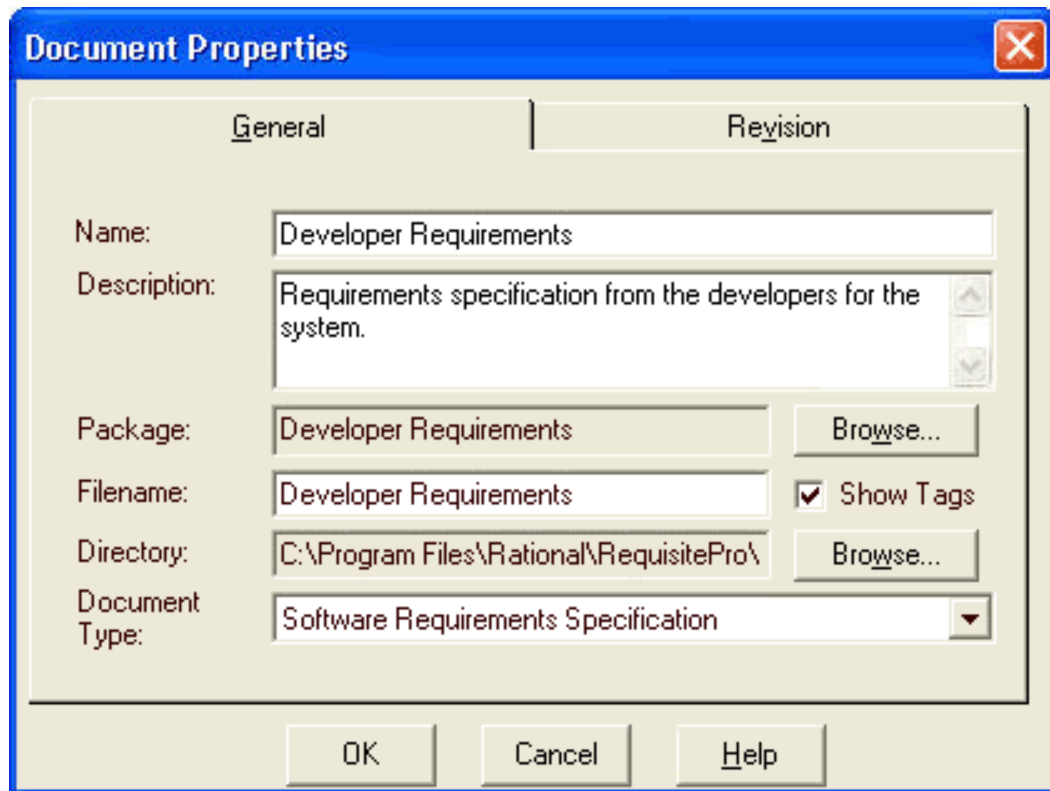
Although it's possible to achieve the same result in Word from an existing project and requirements list by using cut and paste, it's a time consuming process with larger requirements specifications and certainly not recommended unless you have a lot of time.

The benefits of using Word are that you can add and incorporate additional pieces of information into the requirements specification document that are not recorded and stored within the requirements database.

To generate requirements from within a Word document:

1. Generate a new document, or open an existing document if one was created as part of an existing project template.
2. To create a new document, select **File > New > Document** or right-click on the tree view of the project and select the same options. A window opens as shown in Figure 13.

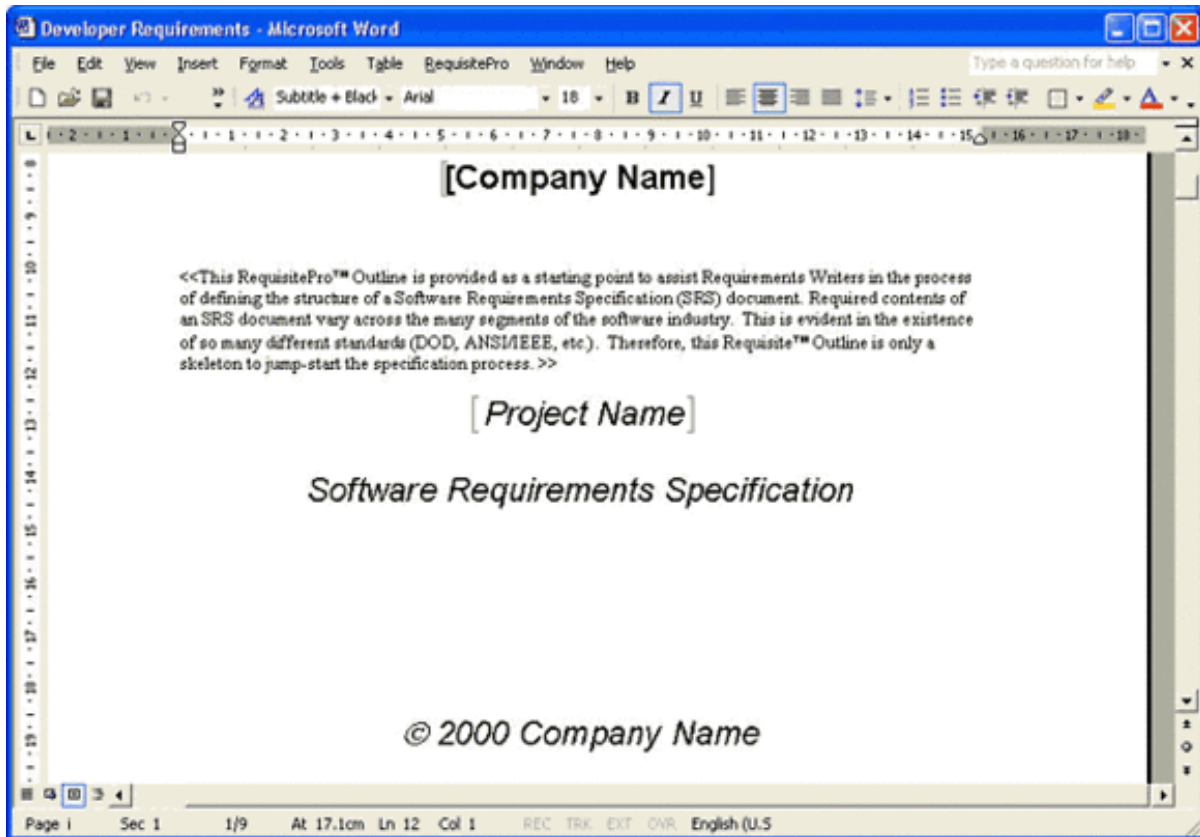
Figure 13. Document properties window



3. Enter a name for the document (as it appears in the project) and provide a description of what the document contains. You'll be using a document for your Developer Requirements specification so name the document accordingly. As you can see from the sample, this document was created in a new package that has not been created yet, so create it in the root package.
4. For the document type, select the Software Requirements specification. This creates a document using the corresponding Word template.
5. Click **OK** to create the document and open the template within Word.

Now that you have the document open (see the sample shown in Figure 14), it's time to start populating the document with information and create requirements.

Figure 14. Populating the document



Adding requirements from within Word

The default document templates provide a reasonable walk-through of the expected content of the requirements specification and it's up to you to fill out the information such as the project name, company, and the outline details and contents of the specification and system.

A new toolbar is available within Word when you are editing a RequisitePro document (see Figure 15 below). You'll need this -- or the RequisitePro menu -- as you create requirements directly in the document.

Figure 15. RequisitePro toolbar menu



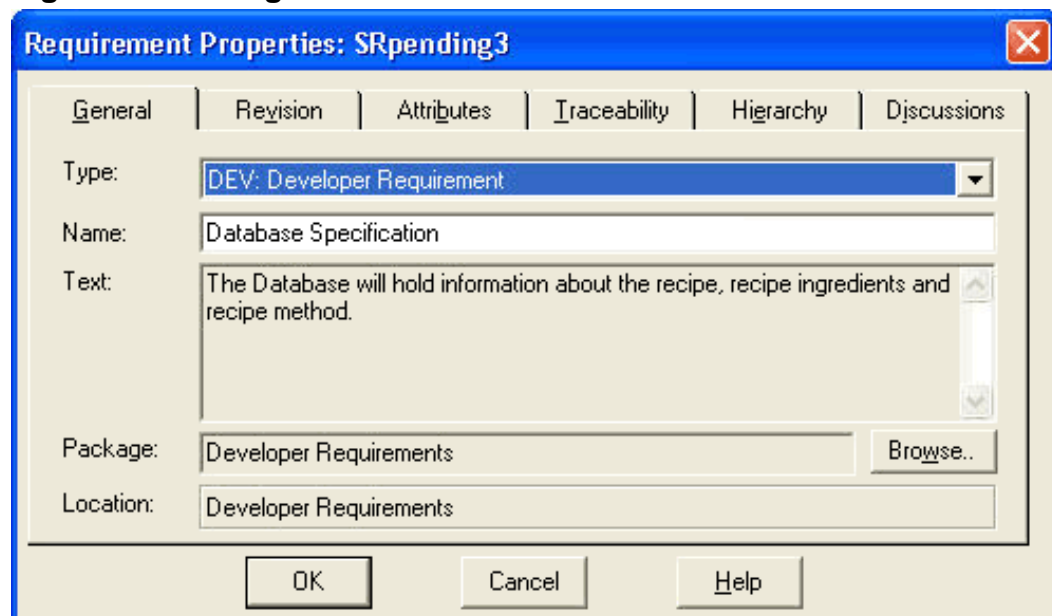
To create a requirement within the document:

1. Enter the text description for the requirement directly into the Word

document. Enter The Database will hold information about the recipe, recipe ingredients and recipe method for this example.

2. Select the text you just typed into the document and then click **New Requirement** (the first square bracket ([]) button), or select **New Requirement** from the RequisitePro menu.
3. Enter a name for the requirement, specify the requirement type, and ensure that the Package and Location information within the requirement properties are correct (see Figure 16 below). These last two properties should automatically be populated based on the location of the document you are editing.

Figure 16. Package and location information



4. The document text you selected is replaced with the requirement text - specially tagged within the document so that it can be updated by RequisitePro. Meanwhile, the text you selected is submitted into the RP database. Note that the item is marked as pending until you save and close the document within Word. Part of the save process imports all of the information into the RequisitePro database.

Importing an existing requirements document

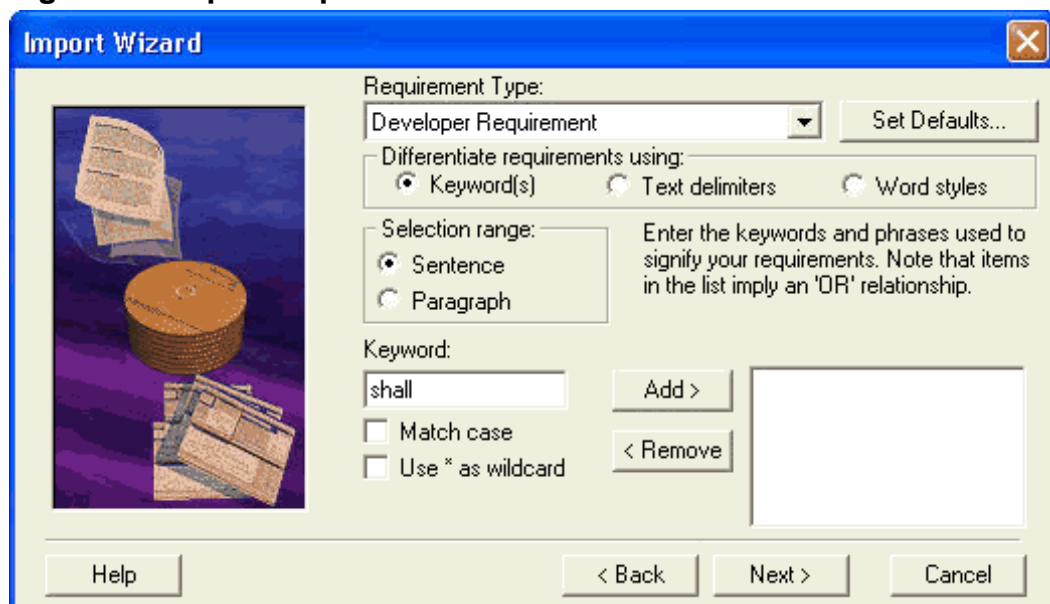
If you already have requirements documents written in Word, import the requirements specification into a RequisitePro project in one of two ways. You can either try a direct import, which attempts to determine the requirements from the

Word document using text parsing and delimiting techniques, or you can perform the operation using the method described above after you have imported the document into the RequisitePro project.

To import an existing document into a project:

1. Click **File > Import** in RequisitePro.
2. Select the Microsoft Word Document radio button and use the browser to locate the document you want to import.
3. To parse the document contents and import requirements into the project, select **Requirements only**. To import the requirements and then add the document as a document within the project, select **Requirements and document**. Alternatively, use RequisitePro to trace and track changes to any Word document that might be related to the project. Either way, select **Document only**.
4. Click **Next**.
5. If you've chosen either of the Document options, then you need to specify the same document options as if you were creating a new document, such as the document name, package location, and document type. Click **Next** to continue the import process.
6. If you have opted to import requirements and parse the contents, the Import Wizard window appears as shown in Figure 17.

Figure 17. Import requirements



7. Specify how the system parses the content. Use keywords to identify paragraphs and sentences that define requirements and text delimiters, or use Word styles to identify the requirements in the source document.
8. Once you've selected the system to use, click **Next**. You'll be asked to approve each matched requirement. Select **Yes to all** to accept all of the matching requirements once you have verified the basic layout.

It is also possible to import requirements from a Comma Separated Values (CSV) text file, which can be useful if you used Excel Microsoft Project or other tabular/database tools to define your requirements.

Exporting requirements into a Word document

If you chose to define all of your requirements and information directly within the RequisitePro explorer, export those requirement details into a Word document for distribution at a meeting or as part of an approval process. You can, of course, do this for any requirements in the database, including those you create from within other Word documents.

To generate a Word document based on the database contents, click **File > Export**. You must first choose a View from the explorer to select which requirements are exported into the document. Views are discussed later in this tutorial.

Note that this process creates a standalone document. There is no linking between the exported requirements and database as there is with requirements generated from within Word. The format is also less than ideal, but it should enable you to quickly dump the contents into a file for a report.

Mixing source and generated documents

There is no reason why you can't mix and match the documents that are stored in a RequisitePro project. Many projects consist of the document types created when requirements were defined along with the vision and any other elements, with those that you have generated directly from the database.

There is no limit to the number of documents or their structure and how they are stored and organized within the system.

Section 7. Reports and views

Views

Now that you have a list of requirements for your recipe database in the system, you need to view and report on those requirements so you can make decisions about what steps are required in the development process to achieve those goals. In RequisitePro, those reports are called Views and they are a powerful tool to use during the development of the software project plan.

Views in RequisitePro use standard query techniques on the database of requirements to show information and relationships. Each view is composed of the query that generates the information to be displayed and the view type. There are four predetermined formats for viewing the information as follows:

- **Attribute Matrix** -- a simple table of requirements and their attributes.
- **Traceability Matrix** -- a matrix table, showing the relationship between two requirements lists. For example, you might want to show the relationship between the stakeholder requests and the main software requirements.
- **Traceability Tree (Traced into)** -- a tree showing how requirements relate to the specified requirement. For example, you might want to show a tree that describes how stakeholder, developer, and other requirements related to the main software requirements.
- **Traceability Tree (Traced out of)** -- a tree showing the requirements traced from the specified requirement type. For example, you might want to show all the requirements traced from a stakeholder request.

It is through the various views that you manage, control, and monitor the information in your requirements specification. Views are used throughout the life of a requirements project.

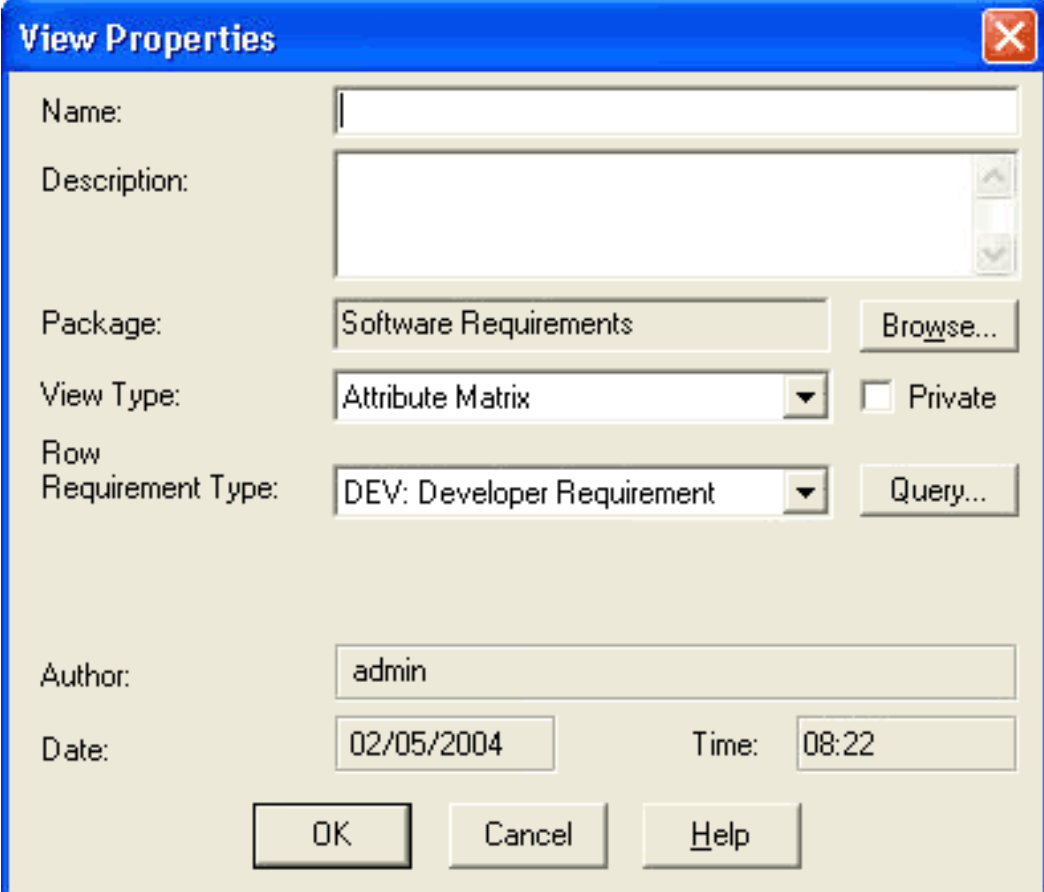
Creating a new View

Views can be created anywhere within the project structure, but typically they are created to show specific information within a package. These packages can either be those relating to specific requirement types, as in Stakeholder Requests or Developer Requirements, or they can be packages specially designed to collate and view information.

To create a new View:

1. Right-click anywhere in the tree view and click **New > View** or select the same option from the File menu (see Figure 18).

Figure 18. Creating a new View



The screenshot shows a 'View Properties' dialog box with the following fields and values:

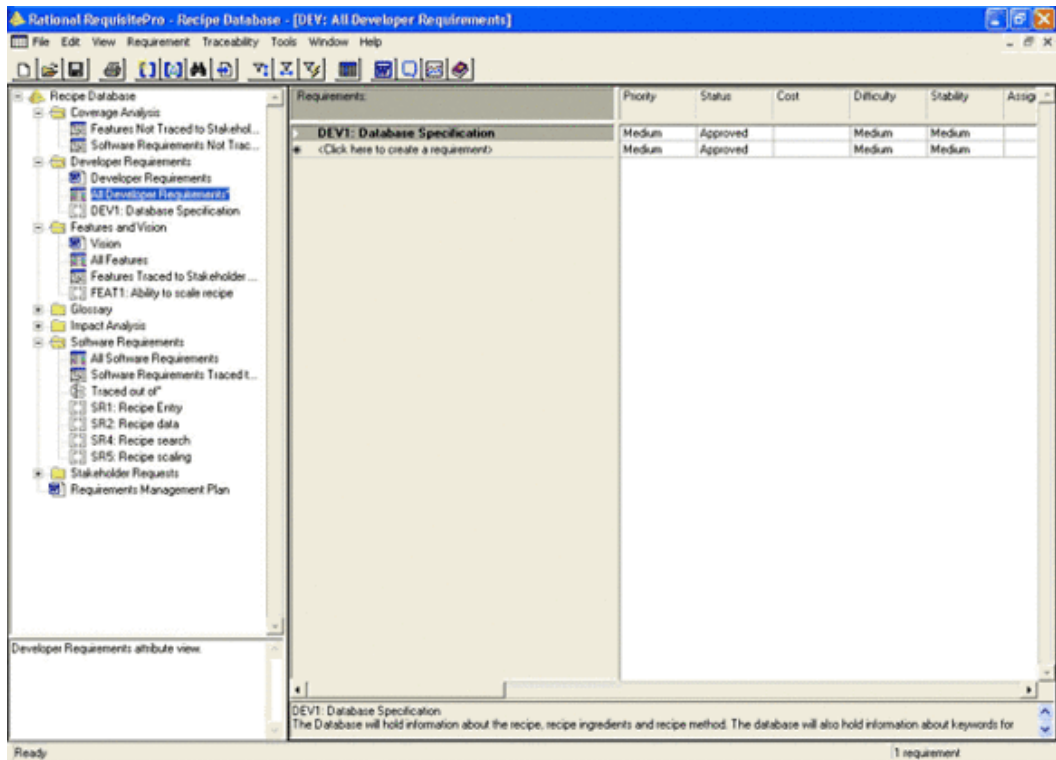
- Name: (empty)
- Description: (empty text area)
- Package: Software Requirements (with a 'Browse...' button)
- View Type: Attribute Matrix (with a 'Private' checkbox)
- Row Requirement Type: DEV: Developer Requirement (with a 'Query...' button)
- Author: admin
- Date: 02/05/2004
- Time: 08:22

Buttons at the bottom: OK, Cancel, Help.

2. You're creating a simple Attribute Matrix view for the Developer Requirements type created earlier in this tutorial, so name the view `All Developer Requirements` and for the description, enter `Developer Requirements attribute view`.
3. Make sure the Developer Requirements package has been selected.
4. Select **Attribute Matrix** as the View Type.
5. Select **Developer Requirement** for the Row Requirement Type. This option selects which requirement types to show within the view.
6. Click **OK**.

- Once the view has been created, RequisitePro explorer changes to this view as shown in Figure 19.

Figure 19. The RequisitePro explorer



This is just a simple view. You can create more powerful or specific views by creating a query that selects specific requirements from the database. That's next.

View queries

Queries within a view allow you to customize the information that is being displayed. For example, you might want to create a view that shows all of the high-priority, but low-risk stakeholder requirements in a project so that you can organize the main software requirements.

The role of the query is a critical part of the requirements process. The query system allows you to monitor and track requirements through the system, report on relationships between requirements, and generate reports and documents based on the results.

Queries are generated as part of creating a view by clicking on the Query button. If you are familiar with other database systems such as Access or Structured Query Language (SQL) based queries, then the basic structure should be familiar.

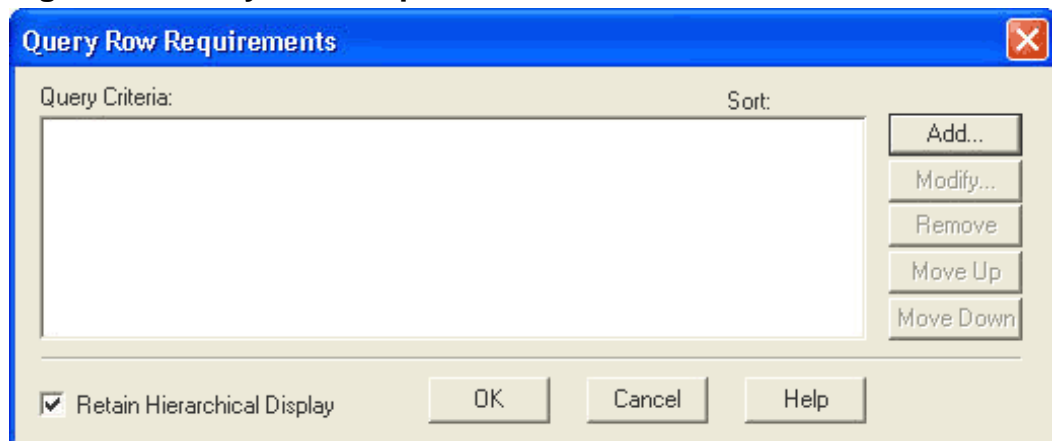
Alternatively, if you have experience using the filtering system for email messages in Microsoft Outlook or another mail client, the basic principles remain the same.

In practice, the queries work like filters selecting requirements from the list. All queries have an order. The higher the order, the earlier the filter is processed. This is a bit like performing a logical AND on each query.

To create a view that displays only the high-priority developer requirements:

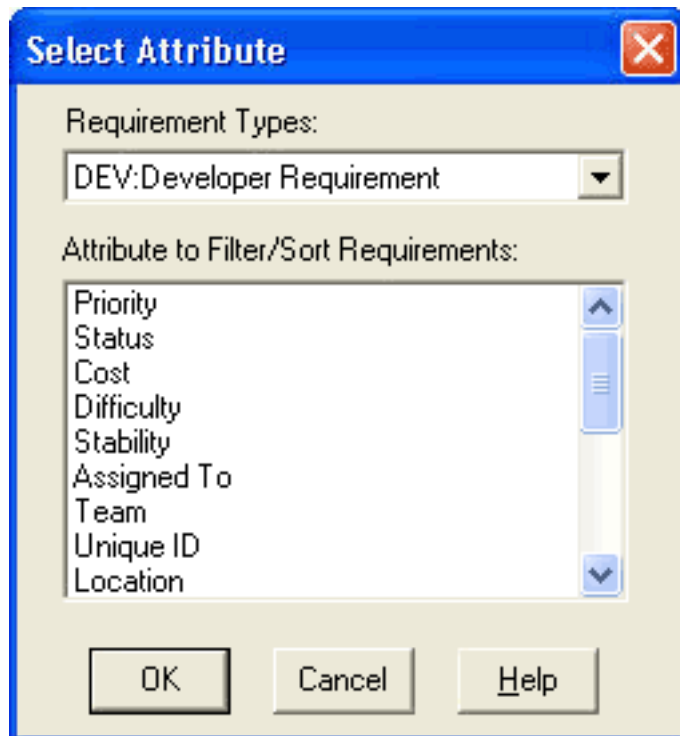
1. Right-click anywhere in the tree view and select **New > View** or choose the same option from the File menu.
2. Name the view `High Priority Developer Requirements` (ignore the description for the moment).
3. Make sure the Developer Requirements package has been selected.
4. Select **Attribute Matrix** as the View Type.
5. Select **Developer Requirement** for the Row Requirement Type.
6. Click **Query** to get the query builder window. Click **Cancel** in the initial window (see Figure 20) to focus on the query building process.

Figure 20. Query Row Requirements window



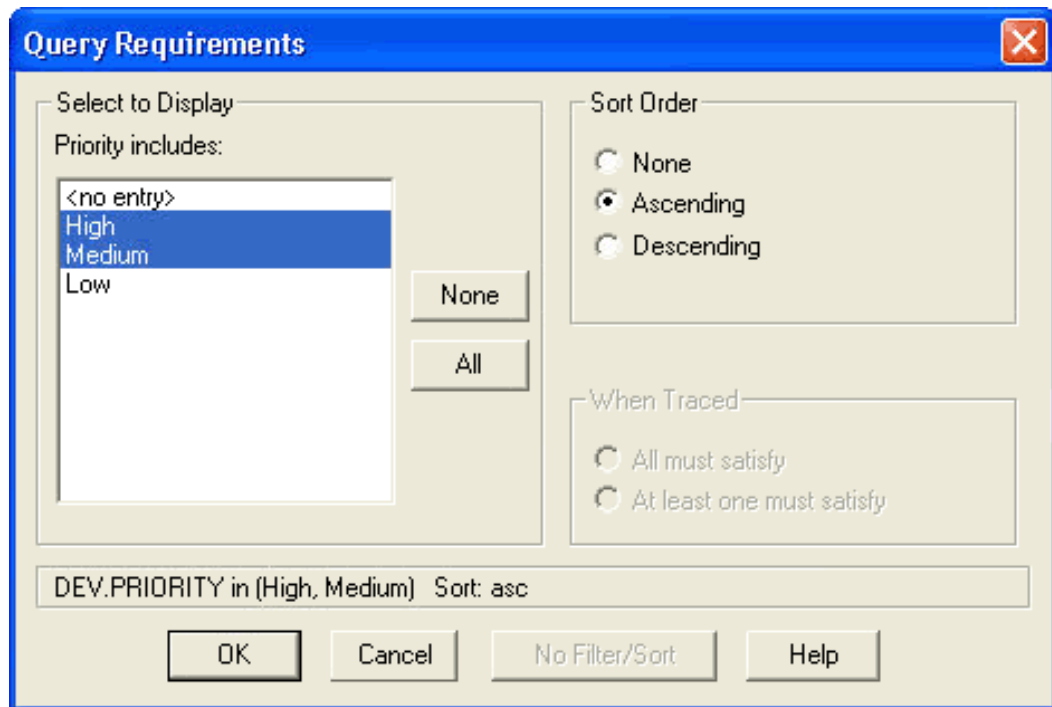
7. Add new criteria to the query by clicking **Add** and selecting the requirement type and attribute to filter. Make sure Developer Requirement is selected and then choose the Priority attribute from the list (see Figure 21).

Figure 21. Selecting the requirement type and attribute to filter



8. Specify the expected value. The Priority attribute has a fixed list of values, so you are provided with the list of these values. You can select **All** or **None** of these, or make multiple selections by clicking on each option as shown in Figure 22. Make sure only High is selected. You can also select a sort order (only applicable if you are selecting multiple list values or working with a value attribute such as text, integer, or date).

Figure 22. List of values for the Priority attribute



9. Click **OK**.

You can add further queries to the list. For example, add further queries to only show those from a specific group of people or only those due by a certain date.

Section 8. Manage requirements

Managing priorities, goals, and other attributes

With the requirements for your recipe database application in the system, you need to start making decisions about the importance and priority of the various requirements so the development team can focus their energies on the requirements with the highest priority. For example, there is no point in developing a scaling system for your recipe if you have no recipe database with which to scale any ingredients. This phase of the requirements process is about the management of requirements and using the properties to help sort, prioritize and categorize your requirements.

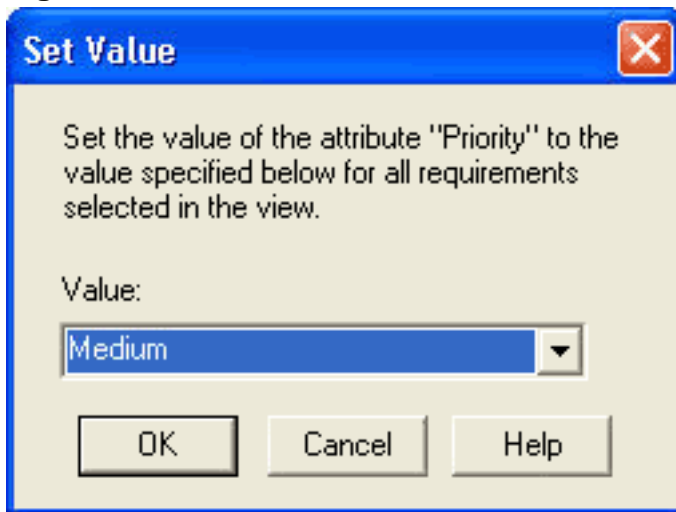
Different attributes on the requirements enable you to set different parameters to

help you organize and prioritize the items. To modify and edit the attributes of a requirement, edit the properties for the requirement.

You can also edit the attributes for requirements through a Matrix View, which shows the value of all the attributes for the matching requirements as part of its display.

Finally, you can 'bulk' set attributes on a range of requirements by clicking the **Set Attributes** button in the toolbar. This sets the attribute value for the selected attribute column on all the requirements in a given view to a single value (see Figure 23).

Figure 23. Set value window



For example, use this to set the initial priority of all attributes to Medium at the start of a project.

Set other attributes values that you have created in the same way to set and organize your project, including setting up expected dates, durations, complexity, source and even the person the requirement to whom the management of the development has been assigned. Remember that attributes are completely free form so you can create as many attributes in a variety of different types as you need to help manage your system.

Collating requirements into a final list

Different requirements from the various packages in your project need to be collated into the final list of software requirements for the overall project. You can do this manually, but the better alternative is to use the tracing functionality within RequisitePro to create links between requirements and then create reports and queries that summarize this information.

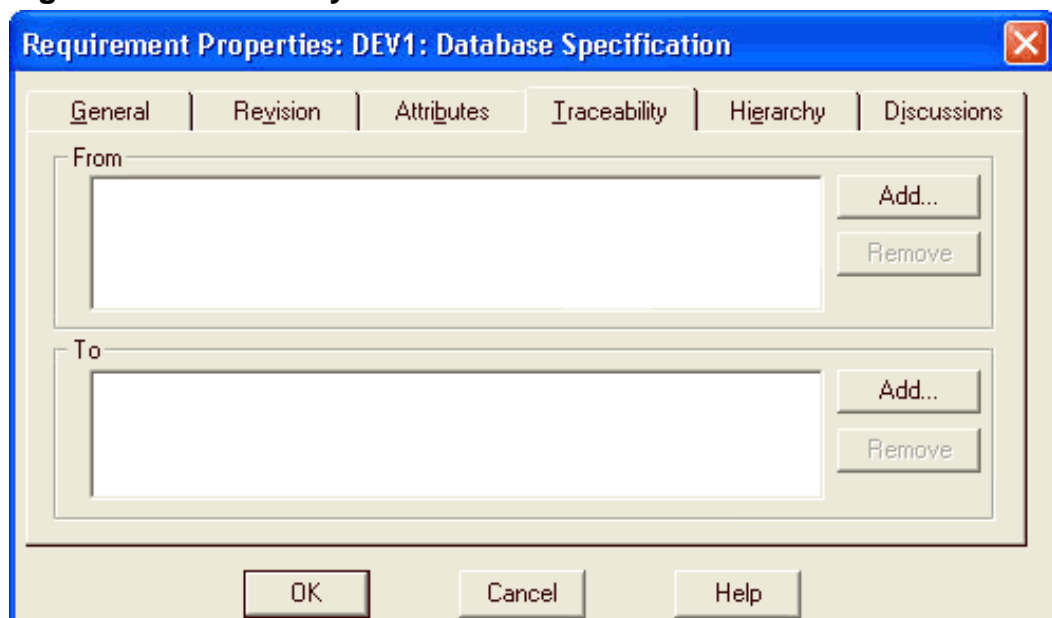
As part of this process you might find that you are creating requirements that refer back to earlier requirements of the same or different types. These relationships are called traceability and can occur in either direction.

For example, you created a Software Requirement called Recipe data and you also created a Database Requirement that contained the database specification. These two requirements are related. The software requirement for a recipe database implies the requirement in developer requirements. In this case, the software requirement traces to the developer requirement.

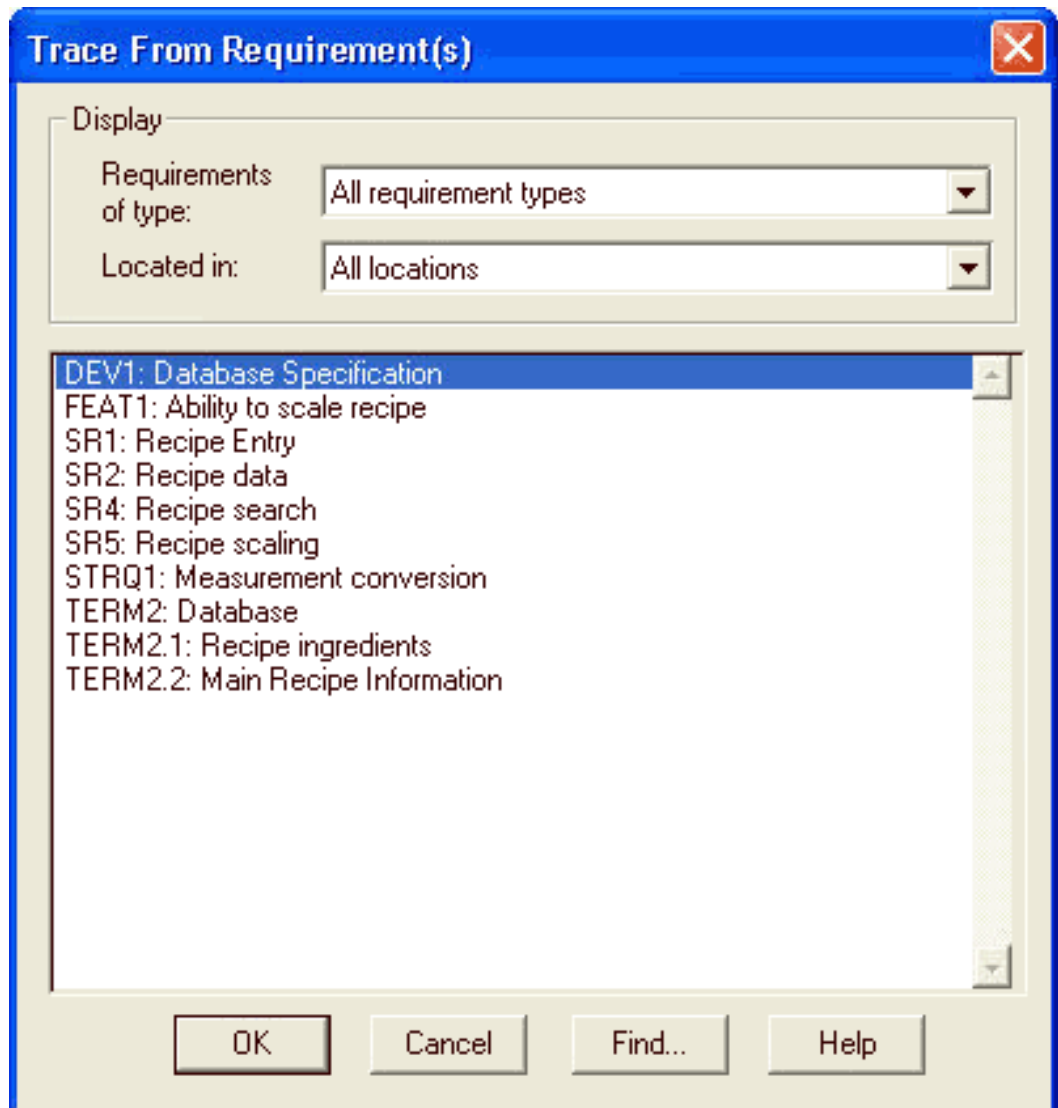
To set up the link between these two requirements, edit either requirement. Let's change the developer requirement.

1. Open the properties for the Database Specification Developer Requirement.
2. Click the Traceability tab as shown in Figure 24.

Figure 24. Traceability tab



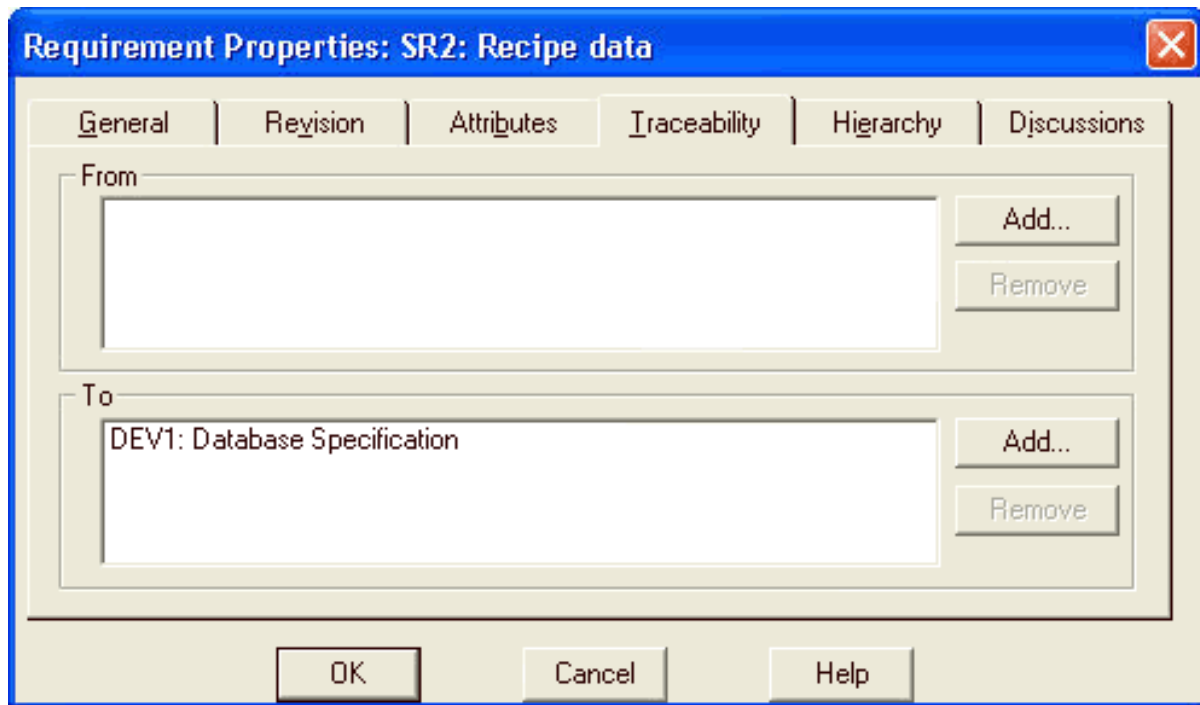
3. Click **Add** on the Traceability tab to create a link back to the software requirements. The following window appears:
Figure 25. Trace from Requirement(s) window



4. Select the Recipe data Software Requirement (probably with the ID SR2). Click **OK**.

You've now created a link between the two. If you open up the properties for the Recipe data software requirement, the trace to the developer requirement is displayed as shown below.

Figure 26. Recipe data software requirement properties



Now have a good idea of how to leverage the power of RequisitePro to specify and manage requirements, thus better enabling the development of a successful application.

Section 9. Wrap up

Summary

In this tutorial you saw the basic mechanics of the software requirements process by using a recipe database as the target application. You used the requirements for the recipe database to understand how to begin to model and record the various requirements within RequisitePro. The software allows you to create requirements either directly into the database, or from within a Word document and supports a full range of tracking history facilities in combination with a flexible database structure. It is this flexibility that enables you to record additional information about the various requirements, such as their source, status or projected cost.

You also used the relationship between the original software requirements and developer requirements to identify how to relate and trace requirements through the life of the project. These links and relationships between the various requirements

can be used to trace the origin and therefore the importance of each requirement in the final system.

Resources

- [Participate in the discussion forum for this content.](#)
- Find more information on the entire suite of [Rational products](#).
- The [developerWorks Rational site](#) is a portal for more information, tutorials and articles on the Rational environment.
- Download a [trial version](#) of RequisitePro.

About the author

Martin C. Brown

Martin C. Brown, a Studio B author, is a former IT Director with experience in cross-platform integration. A keen developer, he has produced dynamic sites for blue-chip customers, including HP and Oracle and is the Technical Director of Foodware.net. Now a freelance writer and consultant, MC, as he is better known, works closely with Microsoft as an SME, is the LAMP Technologies Editor for LinuxWorld magazine, a core member of the AnswerSquad.com team and has written a number of books on topics as diverse as Microsoft Certification, iMacs and open source programming. Despite his best attempts, he remains a regular and voracious programmer on many platform and numerous environments. MC can be contacted at: questions@mcslp.com, or through the [MCslp Web site](#).