NI Requirements Gateway

Customization Guide

January 2009
372657A-01

Worldwide Technical Support and Product Information

`ni.com`

National Instruments Corporate Headquarters

11500 North Mopac Expressway   Austin, Texas 78759-3504   USA   Tel: 512 683 0100

Worldwide Offices

Australia 1800 300 800, Austria 43 662 457990-0, Belgium 32 (0) 2 757 0020, Brazil 55 11 3262 3599, Canada 800 433 3488, China 86 21 5050 9800, Czech Republic 420 224 235 774, Denmark 45 45 76 26 00, Finland 358 (0) 9 725 72511, France 01 57 66 24 24, Germany 49 89 7413130, India 91 80 41190000, Israel 972 3 6393737, Italy 39 02 41309277, Japan 0120-527196, Korea 82 02 3451 3400, Lebanon 961 (0) 1 33 28 28, Malaysia 1800 887710, Mexico 01 800 010 0793, Netherlands 31 (0) 348 433 466, New Zealand 0800 553 322, Norway 47 (0) 66 90 76 60, Poland 48 22 328 90 10, Portugal 351 210 311 210, Russia 7 495 783 6851, Singapore 1800 226 5886, Slovenia 386 3 425 42 00, South Africa 27 0 11 805 8197, Spain 34 91 640 0085, Sweden 46 (0) 8 587 895 00, Switzerland 41 56 2005151, Taiwan 886 02 2377 2222, Thailand 662 278 6777, Turkey 90 212 279 3031, United Kingdom 44 (0) 1635 523545

For further support information, refer to the *Technical Support and Professional Services* appendix. To comment on National Instruments documentation, refer to the National Instruments Web site at `ni.com/info` and enter the info code `feedback`.

Important Information

Warranty

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

National Instruments respects the intellectual property of others, and we ask our users to do the same. NI software is protected by copyright and other intellectual property laws. Where NI software may be used to reproduce software or other materials belonging to others, you may use NI software only to reproduce materials that you may reproduce in accordance with the terms of any applicable license or other legal restriction.

Trademarks

National Instruments, NI, ni.com, NI TestStand, LabVIEW, and MATRIXx are trademarks of National Instruments Corporation. Refer to the *Terms of Use* section on `ni.com/legal` for more information about National Instruments trademarks.

Other product and company names mentioned herein are trademarks or trade names of their respective companies.

Members of the National Instruments Alliance Partner Program are business entities independent from National Instruments and have no agency, partnership, or joint-venture relationship with National Instruments.

Patents

For patents covering National Instruments products/technology, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your media, or the *National Instruments Patent Notice* at `ni.com/patents`.

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.

(2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

Conventions

The following conventions are used in this manual:

» The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.

💡 This icon denotes a tip, which alerts you to advisory information.

📝 This icon denotes a note, which alerts you to important information.

bold Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names.

italic Italic text denotes variables, emphasis, a cross-reference, or an introduction to a key concept. Italic text also denotes text that is a placeholder for a word or value that you must supply.

`monospace` Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames, and extensions.

`monospace bold` Bold text in this font denotes the messages and responses that the computer automatically prints to the screen. This font also emphasizes lines of code that are different from the other examples.

`monospace italic` Italic text in this font denotes text that is a placeholder for a word or value that you must supply.

Platform Text in this font denotes a specific platform and indicates that the text following it applies only to that platform.

Contents

Chapter 1
Capturing Information from External Products

Chapter 2
Customizing Types and Type Elements

Chapter 3
Using and Testing Regular Expressions

Chapter 4 Customizing Reports

Chapter 5
Using OTScript

Chapter 6
Managing Customized Projects

Appendix A
Technical Support and Professional Services

Index

1

Capturing Information from External Products

NI Requirements Gateway is a requirements traceability solution that links development and verification documents with formal requirements stored in documents and databases. Requirements Gateway improves the quality of the development process by effectively managing requirements traceability and impact analysis throughout the life cycle of a project.

You must first configure a project so that Requirements Gateway can analyze it, and then specify the external data source that you want to analyze, including National Instruments products and other third-party products. Requirements Gateway supports the following National Instruments products:

- LabVIEW VIs
- LabWindows™/CVI™ source code and function panel files
- NI TestStand sequence files and XML reports
- MATRIXx SystemBuild catalogs

Requirements Gateway supports the following third-party products:

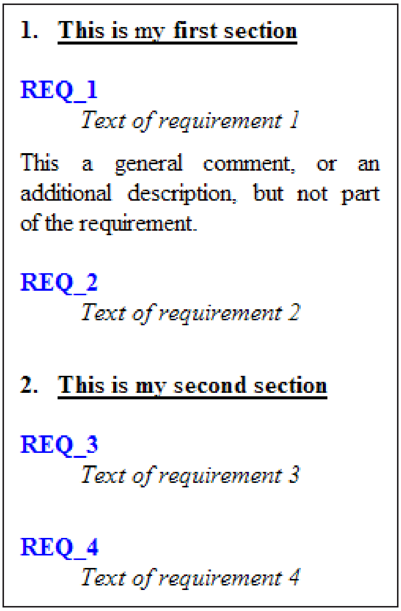
- Microsoft Word documents
- Microsoft Excel spreadsheets
- Microsoft Access database files
- Microsoft Visio project files
- Telelogic DOORS databases
- IBM Rational RequisitePro databases
- Adobe Acrobat PDF files
- Generic text files
- Generic source code files

Requirements Gateway uses intermediate files and document types to process the information in the source files into a form Requirements Gateway can analyze.

Intermediate File

After you configure a project and Requirements Gateway begins to analyze the input documents, a converter transforms the information natively-stored in each input file into an intermediate file that contains the same information in an ASCII or XML format that the engine can analyze.

For example, you might have a Word source file that contains the requirements for the the project, as shown in Figure 1-1.



1. **This is my first section**

REQ_1

Text of requirement 1

This a general comment, or an additional description, but not part of the requirement.

REQ_2

Text of requirement 2

2. **This is my second section**

REQ_3

Text of requirement 3

REQ_4

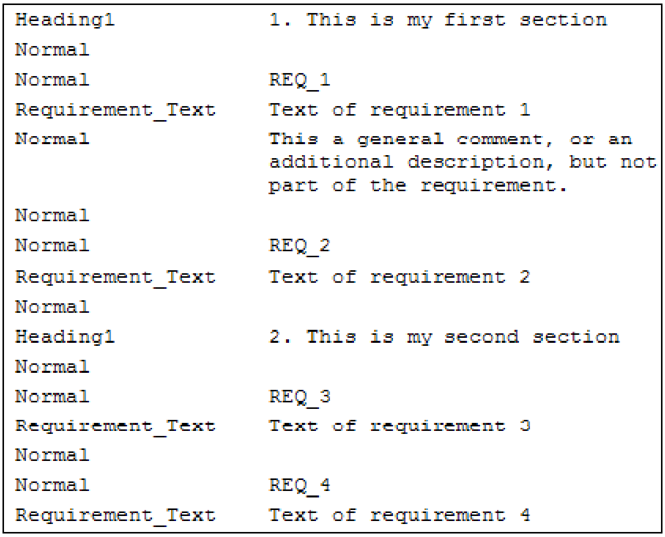
Text of requirement 4

Figure 1-1. Original Microsoft Word Document

In this example, the Word document contains the following discrete elements:

- Section headings
- Requirement IDs
- Requirement text
- General comments

Each element uses a style in the Word source document. Section headings use a Heading1 style, requirement IDs and general comments use a Normal style, and requirement text uses a Requirement_Text style. The converter transforms the textual traceability information into an ASCII intermediate file and includes the Word styles associated with each element, as shown in Figure 1-2. The Word type uses the style tags in the intermediate file and the syntax of the traceability information to capture the information.



```
Heading1            1. This is my first section
Normal
Normal              REQ_1
Requirement_Text    Text of requirement 1
Normal              This a general comment, or an
                    additional description, but not
                    part of the requirement.
Normal
Normal              REQ_2
Requirement_Text    Text of requirement 2
Normal
Heading1            2. This is my second section
Normal
Normal              REQ_3
Requirement_Text    Text of requirement 3
Normal
Normal              REQ_4
Requirement_Text    Text of requirement 4
```

Figure 1-2. Intermediate File Generated from the Microsoft Word Document

The traceability information now exists in a format that is no longer binary or proprietary. However, even though the information is readable by users, Requirements Gateway does not yet know the following relevant information that it must capture:

- How can Requirements Gateway know that `REQ_nn`, where `nn` indicates digits, specifies requirement IDs and must be captured?
- How can Requirements Gateway know that paragraphs that use the Requirement_Text style specify requirement text and must be captured?

Requirements Gateway applies a document type to the intermediate file to capture this information.

Types

A type contains the semantic definition for the traceability information you want to capture from the source file. Each type specifies the expected format for the following traceability elements in each document that uses the type:

- Section
- Macro-requirement
- Requirement
- Entity
- Reference
- Attribute
- Reference attribute
- Link
- Text
- Picture

Refer to Chapter 5, *Customizing Types*, of the *Getting Started with NI Requirements Gateway* manual and to the *NI Requirements Gateway Help* for more information about each traceability element.

Requirements Gateway defines at least one built-in type for each supported external product. The built-in types define a standard syntax for each traceability element in the source file. For example, the built-in Word type recognizes the same syntax for requirements in all Word documents.

You can also create custom types that can capture unique syntax, formatting, or organization of traceability elements in source documents. Use the Types pane of the Configuration dialog box to create custom types. Refer to Chapter 2, *Customizing Types and Type Elements*, of this document and Chapter 5, *Customizing Types*, of the *Getting Started with NI Requirements Gateway* manual for more information about customizing types. Refer to the *NI Requirements Gateway Help* for more information about the Types pane.

Refer to Chapter 6 through Appendix A of the *Getting Started with NI Requirements Gateway* manual for more information about how each external product defines its traceability elements.

2

Customizing Types and Type Elements

If a project document contains traceability information that uses unique syntax that does not adhere to the format a built-in type requires, you can modify the document to conform to the required format or you can create a custom type that processes the format that your source document defines.

Use the Types pane of the Configuration dialog box to create custom types and type elements. Select **File»Edit Types** in the main window, or click the **Edit Types** button, shown at left, on the main window toolbar to launch the Types pane.

Note This chapter describes the options available for configuring each element of a type. Refer to Chapter 5, *Customizing Types*, of the *Getting Started with NI Requirements Gateway* manual for a tutorial on creating a custom type and using the type in a project.

Defining Type Elements

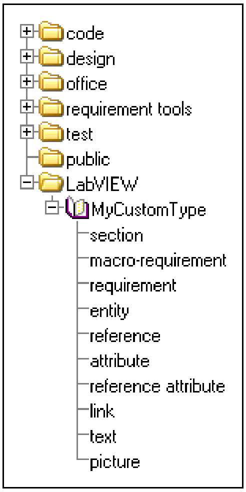
You do not have to define all the elements for a type. You need to define only the elements that are relevant for your needs and to support your requirement standards.

The Types pane contains the Types List, which is a tree view of all the built-in and custom types defined for the project. The Types List groups the types in folders that correspond to the directories in the `<Requirements Gateway>\config\types` directory. The final folder in the list represents the custom types defined for any projects in the directory of the current project file. Select a type in the Types List to populate the other options of the Types pane.

The Types List also contains a public folder. Place custom types and duplicates of the default types in the public folder to make the types available for all projects. Requirements Gateway stores the type definitions for the public folder in `<Requirements Gateway Public>\Config\types\public\public.types`. The `<Requirements Gateway Public>` directory is located by default at `C:\Documents and Settings\All Users\Documents\National Instruments\`

`Requirements Gateway x.x` on Windows 2000/XP and at `C:\Users\Public\Documents\National Instruments\Requirements Gateway x.x` on Windows Vista.

Each type in the list contains its own tree view of all the traceability elements, or type elements, that you can define, as shown in Figure 2-1.



- code
- design
- office
- requirement tools
- test
- public
- LabVIEW
  - MyCustomType
    - section
    - macro-requirement
    - requirement
    - entity
    - reference
    - attribute
    - reference attribute
    - link
    - text
    - picture

Figure 2-1. Types List of the Types Pane

To configure an element of a custom type, select the element under the custom type in the Types List, and use the buttons on the Types pane toolbar to create and delete elements. Use the Analysis, Advanced Options, and Attributes tabs of the Types pane to further configure the element. Refer to the *NI Requirements Gateway Help* for more information about the options on the Types pane toolbar and tabs.

The elements of the built-in types are predefined and you cannot modify them. However, you can duplicate a built-in type and modify the copy, and apply the customized type to a project document. Expand the root node of a built-in type in the Types List and select **Duplicate** from the context menu. Requirements Gateway creates a copy of the built-in type in the last folder of the Types List. The copy contains all the same element definitions as the built-in type, and you can completely modify the copy.

Note Requirements Gateway uses regular expressions similar to Perl to specify the syntax and parameters for capturing elements. This chapter uses only some examples of regular expressions. Refer to Chapter 3, *Using and Testing Regular Expressions*, for more information about regular expressions.

Section

A section is a hierarchical element that specifies how to identify and display the structure within a document. Examples of sections include the following:

- Headings in a text or Microsoft Word file
- Files in a directory
- Rows in a database
- Sequences, step groups, and steps in a TestStand sequence file
- Front panels, diagrams, controls, and indicators in a LabVIEW VI

You can specify the following options on the Types pane for a section element:

- **Name**—The name of the section element, which is displayed in a tooltip when you hover the cursor over the section of a document in the main window.
- **Display**—The name that Requirements Gateway displays in the project workspace for each section of the document. If this option is left empty, Requirements Gateway uses the section names from the source document as display names.
- **Text display**—Additional text for the section element that Requirements Gateway displays on the Text pane of the following locations when you select a section element in the project workspace:
  - On the Texts and Reference Attributes tab of the Coverage Analysis and Impact Analysis views
  - On the Detailed Information pane of the Graphical and Requirement Details views
- **Identifier format**—The identifier for the section element. The identifier can use information provided by a regular expression. For example, you can enter `\1` and `\2` in this control to refer to the first and second fields of the regular expression.
- **Regular expression**—The regular expression used to capture section elements in the source files. You can use the Regular expression option alone, or you can use a regular expression in tandem with an end regular expression.
  - If the End regular expression control is empty, the Regular expression option defines the entire captured section element.
  - If the End regular expression control is filled, the Regular expression option defines only the expression that represents the opening tag of a section element.

You can also right-click in the Regular expression control to open a context menu that contains the following fields automatically generated according to the number of sub-expressions contained in the Regular expression control:

- **Identifier**—Captures the ID of the element.
- **Label**—Captures any optional text used to notarize the element.
- **Text**—Captures the text of the element.
- **Guid**—Captures an internal ID.
- **Parent**—Captures the ID of the element that is one hierarchical level higher than the currently selected element.
- **Depth**—Captures the depth of a section heading. This option is available only for section elements.

- **End regular expression**—The closing tag of a section element. You can use this option to recursively analyze the document so that you can capture sections from files that use a hierarchy of sections, such as the following structure:

```
[Section]
Info…
    [Section]
    Other info…
    [End of Section]
[End of Section]
```

Note The End regular expression control is predefined for dedicated interfaces. If you need to analyze recursive structures in XML, use XML type definitions. Refer to the *Defining an XML Type* section of this chapter and to the *NI Requirements Gateway Help* for more information about XML types.

Creating Multiple Section Levels

You can create multiple levels of sections with multiple section branches at each level. For example, you can have multiple top-level sections, multiple second-level sections, and so on.

Complete the following steps to create multiple levels of sections.

1. Expand the root node of a custom type in the Types List on the Types pane.
2. Select the **section** branch and click the **Add new type** button, shown at left, on the Types Toolbar to create a new section element under the section branch.
3. Enter `SectionA1` in the **Name** control.
4. To create a second branch, select the parent section branch and click the **Add new type** button again to create a second section element on the same level as the SectionA1 element.
5. Name this section SectionB1.
6. Select the SectionA1 element and click the **Add new type** button to create a new section element under the SectionA1 element. Name this section SectionA2.
7. Select the SectionA2 element and create two subsections, SectionA3-1 and SectionA3-2.
8. Select the SectionA3-1 element and create SectionA4.

   Figure 2-2 shows the completed section hierarchy.

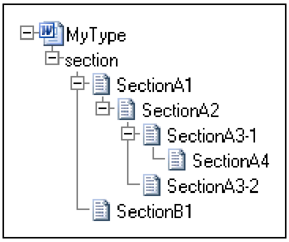


Figure 2-2. Hierarchy of Multiple Section Elements

Creating a Concise Section Expression

When the content of multiple section elements is nearly identical, you can create a concise expression that captures multiple sections simultaneously.

Complete the following steps to create a concise section expression for capturing multiple headings.

1. Expand the root node of a custom type in the Types List on the Types pane.
2. Select the **section** branch and click the **Add new type** button to create a new section element.

3. Name the section, and enter `^Heading (\d+)\s(.+)$` in the **Regular expression** control on the Analysis tab.
4. Right-click the **Regular expression** control and select **Field1»Depth** from the context menu to assign Field1 to Depth.
5. Right-click the **Regular expression** control again and select **Field2» Identifier** from the context menu to assign Field2 to Identifier.

Macro-Requirement

A macro-requirement encapsulates requirements and passes its attributes, text, or links onto those requirements. If a downstream element directly references a macro-requirement, all the requirements that the element contains are considered referenced by the element.

A macro-requirement element specifies how to identify and display macro-requirements for a document. A macro-requirement element defines a starting location and an ending location, where Requirements Gateway associates any requirement element between the locations with the macro-requirement. For example, a macro-requirement element might interpret the syntax `[MacroReq_reqid]`, where reqid is the macro-requirement ID, as a starting location, and the syntax `[End_of_MacroReq]` as an ending location.

You can specify the following options on the Types pane for a macro-requirement element:

- **Name**—The name of the Macro-requirement element, which is displayed in a tooltip when you hover the cursor over the macro-requirement in the main window.
- **Display**—The name that Requirements Gateway displays in the project workspace for each macro-requirement of the document. If this option is left blank, Requirements Gateway uses the macro-requirement names from the source document as display names.
- **Text display**—Additional text for the macro-requirement element that Requirements Gateway displays on the Text pane of the following locations when you select a macro-requirement element in the project workspace:
  - On the Texts and Reference Attributes tab of the Coverage Analysis and Impact Analysis views.
  - On the Detailed Information pane of the Graphical and Requirement Details views.

- **Identifier format**— The identifier for the macro-requirement element. The identifier can use information provided by a regular expression. For example, you can enter `\1` and `\2` in this control to refer to the first and second fields of the regular expression.
- **Regular expression**—The regular expression used to capture the macro-requirement elements in the source files. This expression specifies the opening tag of a macro-requirement element.
- **End regular expression**—The closing tag of a macro-requirement element. If this option is left empty, the Regular expression option defines the starting position of the macro-requirement, and the macro-requirement is considered closed at the end of the source document.
- **Prefix expression**—An additional expression for the macro-requirement that allows you to concatenate the prefix with elements that the regular expression captures to obtain the complete macro-requirement ID.

  For example, you might define requirements in a document as follows:

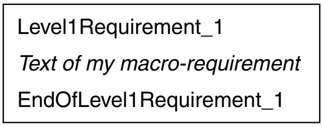
  - `REQ_Product1_`
  - `F1: Max time response shall be 5 ms`
  - `F2: The system shall be active in less than 10 clock periods`

  If you specify `REQ_Product1` as a prefix expression, you can capture requirements as follows:

  - `REQ_Product1_F1: Max time response shall be 5 ms`
  - `REQ_Product1_F2: The system shall be active in less than 10 clock periods`

Defining a Macro-Requirement

Consider the following source file example, as shown in Figure 2-3, as you complete this tutorial for capturing macro-requirements.



Level1Requirement_1

Text of my macro-requirement

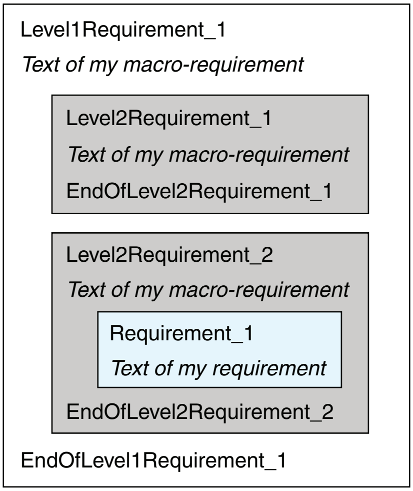
EndOfLevel1Requirement_1

Figure 2-3. Example of Macro-Requirements in a Source File

Complete the following steps to define a macro-requirement element to capture the macro-requirement shown in Figure 2-3.

1. Expand the root node of a custom type in the Types List on the Types pane.
2. Select the **macro-requirement** branch and click the **Add new type** button on the Types Toolbar to create a new macro-requirement element.
3. Enter a name for the macro-requirement element in the **Name** control.
4. Enter `\s+(Level\d+Requirement_\S*)` in the **Regular expression** control on the Analysis tab.
5. Enter `\s+EndOf(Level\d+Requirement_\S*)` in the **End regular expression** control.
6. Click **Apply** to apply your changes and reload the project.

You can define recursive definitions for macro-requirement elements. For example, you can use the regular expression defined above to capture macro-requirements from the file structure shown in Figure 2-4.



Level1Requirement_1

Text of my macro-requirement

Level2Requirement_1

Text of my macro-requirement

EndOfLevel2Requirement_1

Level2Requirement_2

Text of my macro-requirement

Requirement_1

Text of my requirement

EndOfLevel2Requirement_2

EndOfLevel1Requirement_1

Figure 2-4. Macro-Requirement Hierarchy

Use the **Regular expression** and **End regular expression** controls to recursively capture this requirement hierarchy.

Requirement

A requirement expresses a need or constraint, such as a technical constraint, cost, or deadline.

A requirement element specifies how to identify and display requirements for a document. For example, a requirement element might interpret `[REQ: reqid]`, where `reqid` is the requirement identifier, as a requirement.

You can define the following options on the Types pane for a requirement element:

- **Name**—The name of the requirement element, which is displayed in a tooltip when you hover the cursor over that element in the main window.
- **Display**—The name that Requirements Gateway displays in the project workspace for each requirement element. If this option is left empty, Requirements Gateway uses the requirement names from the source document as display names.
- **Text display**—Additional text for the requirement element that Requirements Gateway displays on the Text pane of the following locations when you select a requirement element in the project workspace:
  - On the Texts and Reference Attributes tab of the Coverage Analysis and Impact Analysis views.
  - On the Detailed Information pane of the Graphical and Requirement Details views.
- **Identifier format**—The identifier for the requirement element. The identifier can use information provided by a regular expression. For example, you can enter `\1` and `\2` in this control to refer to the first and second fields of the regular expression.
- **Regular expression**—The regular expression used to capture requirement elements in the source files. This expression specifies the opening tag of a requirement element.
- **Sub-regular expression**—An additional regular expression that captures a second level of analysis to process the result of the preceding regular expression.

- **End regular expression**—Specifies the closing tag of a requirement element. If this option is left empty, the Regular expression option defines the starting position of the requirement, and the requirement is considered closed at the beginning of the next captured requirement.
- **Prefix expression**—An additional expression for the requirement element that allows you to concatenate the prefix with elements that the regular expression captures to obtain the complete requirement ID.

Table 2-1 shows a standard requirement statement in a source document that contains a requirement ID and requirement text.

Table 2-1. Example of a Standard Requirement

Requirement ID	Requirement Text
REQ1	This is the requirement text.

You can capture the requirement ID and text using the following expressions:

- **Requirement**—`(REQ\d+)`
- **Text**—`\|2 ([^\|]+)`

You can also use a single regular expression `(REQ\d+)\|2 ([^\|]+)` that defines two fields, `(REQ\d+)\|2` and `([^\|]+)`. You can right-click the **Regular expression** control and select **Field 1»Identifier** from the context menu to assign the first field to the identifier, and right-click again and select **Field 2»Text** from the context menu to assign the second field to the text.

Typically, you capture a requirement ID first, and then the requirement text. However, you might specify the requirement text before the ID, as shown in Table 2-2.

Table 2-2. Example of an Inverse Requirement

Requirement Text	Requirement ID
This is the requirement text.	REQ1

You can capture such a requirement by using the expression `\|1 ([^\|]+)\|2 (REQ\d+)` and assigning the first field to the text and the second field to the identifier.

Entity

An entity is an element that must contain a reference to a requirement, derived requirement, or macro-requirement and cannot reference itself. An entity element is similar to a section element, except that Requirements Gateway displays an error message if a reference element does not follow an entity element.

You can specify the following options on the Types pane for an entity element:

- **Name**—The name of the entity element, which is displayed in a tooltip when you hover the cursor over an entity element in the main window.
- **Display**—The name that Requirements Gateway displays in the project workspace for each entity of the document. If this option is left empty, Requirements Gateway uses the entity names from the source document as display names.
- **Text display**—Additional text for the entity element that Requirements Gateway displays on the Text pane of the following locations when you select an entity element in the project workspace:
  - On the Texts and Reference Attributes tab of the Coverage Analysis and Impact Analysis views.
  - On the Detailed Information pane of the Graphical and Requirement Details views.
- **Identifier format**—The identifier for the entity element. The identifier can use information provided by a regular expression. For example, you can enter `\1` and `\2` represent in this control to refer to the first and second fields of the regular expression.
- **Regular expression**—The regular expression used to capture the entity elements in the source files. This expression specifies the opening tag of an entity element.

Reference

A reference indicates the coverage of a requirement. A reference points to a requirement, derived requirement, or macro-requirement.

A reference element specifies how to identify references to requirements for a document. Requirements Gateway does not display reference elements but associates the reference with a preceding section or entity element. For example, an element might interpret `[Covers: reqid]`, where `reqid` is the ID of the covered requirement, as a reference.

You can specify the following options on the Types pane for a reference element:

- **Name**—The name of the reference element, which is displayed in a tooltip when you hover the cursor over the reference in the main window.
- **Display**—The name that Requirements Gateway copies in addition to the requirement ID when you select **Copy For** from the context menu in the main window to copy the reference information for a document type. Refer to the *Copy For* section for more information about using the Copy For option.

  The Display control for a reference element does not specify the name that Requirements Gateway displays in the project workspace, since Requirements Gateway does not display reference elements in the project workspace.
- **Text display**—Additional text for the reference element that Requirements Gateway displays on the Text pane of the Graphical view when you select a reference element.
- **Identifier format**—The identifier for the reference element. The identifier can use information provided by a regular expression. For example, you can enter `\1` and `\2` in this control to refer to the first and second fields of the regular expression.
- **Regular expression**—The regular expression used to capture the coverage information in the source files. This expression specifies the opening tag of a reference element. For example, the expression `Covers_(REQ\d+)` captures only the referenced ID.

  You can right-click the Regular expression control to launch a context menu that contains a list of fields automatically generated according to the number of sub-expressions contained in the Regular expression control. You can assign each field as the following:

  – **Target**—Captures the target requirement identifier.

  – **Source**—Captures the source requirement identifier.

  – **Target document**—Captures the target document identifier.
- **Sub-regular expression**—An additional regular expression that captures a second level of analysis to process the result of the preceding regular expression. This is helpful when the reference information contains references to several requirements.

  For example, you might have the following reference element:

  ```
  CoveredRequirements:REQ1, REQ2, REQ3
  ```

You can specify a regular expression and a sub-regular expression to capture this information.

- **Regular expression**—`CoveredRequirements:(.+)`
- **Sub-regular expression**—`(REQ\d)`

- **Inverse regular expression**—A regular expression used to capture references from an upstream document instead of a downstream document. Refer to the *Using Inverse Regular Expressions* section for more information about using inverse regular expressions.

Using Inverse Regular Expressions

Typically, a downstream document contains references to requirements in an upstream document. Requirements Gateway uses the regular expression defined for a reference element to capture covering statements in the downstream document as references. However, you can specify an inverse regular expression to capture reference information from an upstream document instead.

Figure 2-5 shows an example of a project configuration that contains a standard coverage relationship.

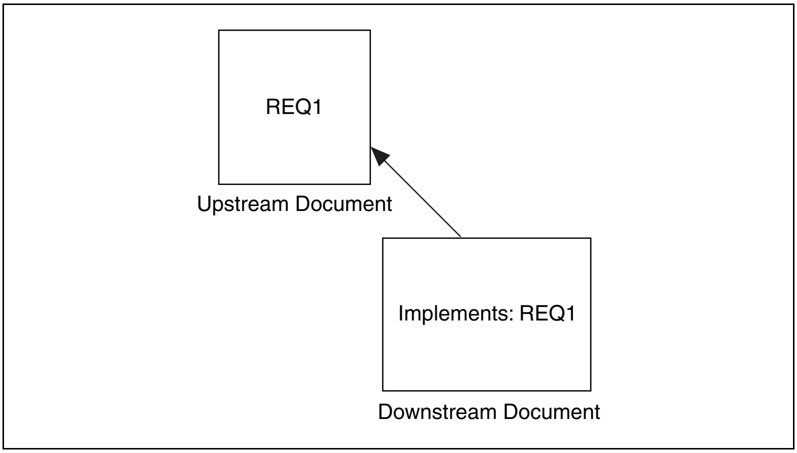


Figure 2-5. Standard Coverage Relationship

The type applied to the upstream document defines `HL_REQ1` as a requirement, and the type applied to the downstream document defines `Implements:` as a reference. Requirements Gateway captures `HL_REQ1` as an upstream requirement and `Implements: HL_REQ1` as the downstream reference.

In an inverse coverage relationship, the downstream document contains the requirement, and the upstream document contains the reference information, as shown in Figure 2-6.

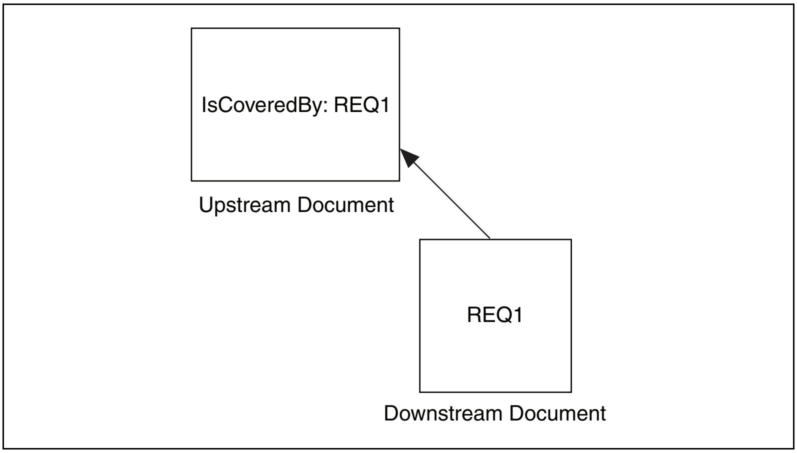


Figure 2-6. Inverse Coverage Relationship

The type applied to the downstream document defines `REQ1` as a requirement, and the type applied to the upstream document defines `IsCoveredBy:` as a reference. Thus, Requirements Gateway captures `REQ1` as a downstream requirement and `IsCoveredBy: REQ1` as an upstream reference.

Copy For

You can create references to the requirements in a single upstream document from multiple downstream documents using the Copy For submenu.

Consider you have an upstream document, Specifications, that you want to implement and verify using two downstream documents, as shown in Figure 2-7. One downstream document, Development, specifies the implementation phase and defines an Implementation reference element with the syntax `Refers to : (.+)?`. The other downstream document, Test Plan, specifies the verification phase and defines a Verification reference element with the syntax `\|3 Normal\t(.+?)\|`, from which `(REQ_\d+)` is extracted.

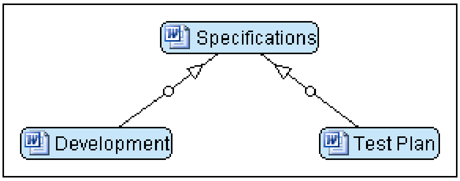


Figure 2-7. Project Configuration with Two Downstream Documents

Right-click a requirement in the upstream document and select **Copy For** from the context menu, which contains the reference elements defined for each downstream document that covers the upstream requirement. In this example, the Copy For submenu contains the reference element names, Implementation and Verification, that are defined for the types applied to the Development and Test Plan documents, respectively.

After you select an option in the Copy For submenu, you can paste the copied reference information into the downstream document. You can define the text pasted in the downstream document using the **Display** control on the Types pane for the reference element. You can use the variables `$i` and `$l` in the Display control to refer to the identifier and label, respectively, of the selected requirement. For example, if you enter `Refers to : $i` in the Display control of the Implementation reference element of the type applied to the Development document, and then right-click a requirement in the upstream document and select **Copy For»Implementation** from the context menu, Requirements Gateway copies `Refers to : REQ_1` to the clipboard so that you can paste the reference statement in the downstream document.

Attribute

An attribute describes a requirement. An attribute has a name and a Boolean, numeric, or string value. A requirement can only have one attribute with a specific name. You can use attributes and their values when you define filters, which specify the requirements to analyze or display in the main window.

An attribute element specifies how to identify and display attributes of requirement elements for a document. An attribute element might define the priority or owner of a requirement. When you select an attribute element in the Types List, the Types pane contains an Analysis tab and an Attributes tab.

You can specify the following options on the Types pane for an attribute element:

- **Name**—The name of the attribute element, which is displayed in a tooltip when you hover the cursor over an attribute in the main window.
- **Display**—The name that Requirements Gateway displays in the project workspace for each attribute element. If this option is left empty, Requirements Gateway uses the attribute names from the source document as display names.
- **Text display**—Additional text for the attribute element that Requirements Gateway displays on the Text pane of the following locations when you select an attribute element in the project workspace:
  - On the Texts and Reference Attributes tab of the Coverage Analysis and Impact Analysis views
  - On the Detailed Information pane of the Graphical and Requirement Details views
- **Identifier format**—The identifier for the attribute element. The identifier can use information provided by a regular expression. For example, you can enter `\1` and `\2` in this control to refer to the first and second fields of the regular expression.
- **Regular expression**—The regular expression used to capture the attribute information in the source files. This expression specifies the opening tag of an attribute element.
- **Sub-regular expression**—An additional regular expression that captures a second level of analysis to process the result of the preceding regular expression.

Refer to the *NI Requirements Gateway Help* for more information about the options on the Attributes tab.

Example of Using Attributes

You might define a Priority attribute with possible values of Low, Medium, and High for your requirement standards.

You can capture the attribute value using the regular expression `Priority=(.+)`. Requirements Gateway captures anything inserted after the `Priority=` syntax, so Requirements Gateway might also capture typing errors or erroneous values.

Use the regular expression `Priority=(Low|Medium|High)` to capture the value of the Priority attribute only if the value matches the accepted values of Low, Medium, or High.

You can also define Low, Medium, High as acceptable attribute values on the Attributes tab of the Types pane. In this case, Requirements Gateway still captures anything inserted after the `Priority=` syntax, but if the captured string does not match the accepted values, Requirements Gateway prompts you with a warning message that the string does not match. National Instruments recommends that you define the list of acceptable attribute values if you are using enumerated attributes.

Reference Attribute

A reference attribute is added to a reference to define the type of coverage for the reference, such as partial or provisional. A reference attribute has a name and a Boolean, string, or numeric value. A reference can only have one attribute with a specific name.

A reference attribute element specifies how to identify and display attributes of reference elements for a document. When you select a reference attribute element in the Types List, the Types pane contains an Analysis tab and an Attributes tab.

You can specify the following options on the Types pane for a reference attribute element:

- **Name**—The name of the reference attribute, which is displayed in a tooltip when you hover the cursor over a reference attribute in the main window.
- **Display**—The name that Requirements Gateway displays in the project workspace for each reference attribute element. If this option is empty, Requirements Gateway uses the reference attribute names from the source document as display names.
- **Text display**—Additional text for the reference attribute element that Requirements Gateway displays in the Text pane of the following locations when you select a reference attribute element in the project workspace:
  - On the Texts and Reference Attributes tab of the Coverage Analysis and Impact Analysis views
  - On the Detailed Information pane of the Graphical and Requirement Details views

- **Identifier format**—The identifier for the reference attribute attribute. The identifier can use information provided by a regular expression. For example, you can enter `\1` and `\2` in this control to refer to the first and second fields of the regular expression.
- **Regular expression**—The regular expression used to capture the reference attribute information in the source files. This expression specifies the opening tag of a reference attribute element.
- **Sub-regular expression**—An additional regular expression that captures a second level of analysis to process the result of the preceding regular expression.

Refer to the *NI Requirements Gateway Help* for more information about the options on the Attributes tab.

Link

A link specifies a non-covering reference to a requirement, or a reference to a section or entity.

A link element must refer to the identifier of the requirement, section, or entity. You can use the link element to navigate from the link element to the requirement, section, or entity element.

After you specify a link element of a type on the Types pane of the Configuration dialog box, you can create links from a document of the type in the project workspace.

You can specify the following options on the Types pane for a link element:

- **Name**—The name of the link element, which is displayed in a tooltip when you hover the cursor over that element in the main window.
- **Display**—The name that Requirements Gateway displays in the project workspace for each link element. If this option is left empty, Requirements Gateway uses the link names from the source document as display names.
- **Text display**—Additional text for the link element that Requirements Gateway displays on the Text pane of the following locations when you select a link element in the project workspace:
  - On the Texts and Reference Attributes tab of the Coverage Analysis and Impact Analysis views
  - On the Detailed Information pane of the Graphical and Requirement Details views

- **Identifier format**—The identifier for the link element. The identifier can use information provided by a regular expression. For example, you can enter `\1` and `\2` in this control to refer to the first and second fields of the regular expression.
- **Regular expression**—The regular expression used to capture the link information in the source files.

  You can right-click the Regular expression control to launch a context menu that contains a list of fields automatically generated according to the number of sub-expressions contained in the Regular expression control. You can assign each field as the following:

  – **Target**—Captures the target requirement identifier.

  – **Source**—Captures the source requirement identifier.

  – **Target document**—Captures the target document identifier.
- **Sub-regular expression**—An additional regular expression that captures a second level of analysis to process the result of the preceding regular expression.

Creating a Link

Consider that you have a string `See_REQ_Specs1_(Specifications)` that you want to capture as a link.

Complete the following steps to create a link element.

1. Expand the root node of a custom type in the Types List on the Types pane.
2. Select the **link** branch and click the **Add new type** button on the Types Toolbar to create a new link element.
3. Enter a name for the link element in the **Name** control.
4. Enter `See_(REQ_\S*)_\(([^\s\#]*)\)$` in the **Regular expression** control.
5. Right-click the Regular expression control and select **Field 1»Target** from the context menu to assign Field1 to target.
6. Right-click the Regular expression control and select **Field 2» Target Document** from the context menu to assign Field2 to target document.
7. Click **Apply** to apply the changes and reload the project.

Text

Text is the descriptive wording associated with a section, entity, requirement, or attribute.

You can specify the following options on the Types pane for a text element:

- **Name**—The name of the text element, which is displayed in a tooltip when you hover the cursor over a text element in the main window.
- **Regular expression**—The regular expression used to capture the text information in the source files.
- **Sub-regular expression**—An additional regular expression that captures a second level of analysis to process the result of the preceding regular expression.

For requirements, you can allocate the fields of a regular expression to capture text. Refer to the *Requirement* section for more information about allocating fields of a regular expression.

Picture

A picture element specifies how to identify image files associated with a preceding section or entity element. Requirements Gateway 1.1 supports images with BMP, PNG, Scalable Vector Graphics (SVG), and Windows Metafile (WMF) file formats.

You can specify the following options on the Types pane for a picture element:

- **Name**—The name of the picture element, which is displayed in a tooltip when you hover the cursor over a picture element in the main window.
- **Regular expression**—The regular expression used to capture the picture information in the source files.

Defining an XML Type

You can create an XML-based type to capture traceability information from an XML intermediate file. The type elements of an XML type capture information from specific XML tags that you define on the Analysis tab of the Types pane.

To create a new XML type, click the **Add XML type** button, shown at left, on the Types Toolbar. You can also duplicate an existing XML type and edit it. Select an element branch of the XML type tree and click the **Add XML type** button again to create an element for the XML type.

When you create an XML type, the Analysis tab of the Types pane contains the following options for defining the XML syntax you want to analyze.

- **Tag**—Specifies if the XML structure you define to capture the selected type element refers to an XML tag. The root of the XML structure you define can only be a tag.
- **Attribute**—Specifies if the XML structure you define to capture the selected type element refers to an XML attribute.
- **Name**—Name of the selected tag or attribute.
- **Identifier**—Specifies that the captured information is the ID of the selected type element.
- **Label**—Specifies that the captured information is the label of the selected type element.
- **Text**—Specifies that the captured information is the text of the selected type element.
- **Guid**—Specifies that the captured information is an optional Globally Unique Identifier (GUID) for the element that must be unique within the document. The value cannot be displayed within Requirements Gateway but can be used to identify the element when reanalyzing the document or performing a navigation.
- **Parent**—Specifies that the captured information is the parent of the selected type element.
- **Image**—Specifies that the captured information is the picture of the selected type element.
- **Condition**—Specifies that the captured information is a value of an XML attribute.
- **Value**—The expected value of an XML attribute. When you select the **Condition** option, you can enter a regular expression in the **Value** control for the condition.
- **Add an XML tag or attribute**—Adds a new XML element below the currently selected XML element in the XML tree view. After you add the XML element to the tree view, select **Tag** or **Attribute** to specify the role of the new XML element.
- **Remove an XML tag or attribute**—Deletes the currently selected XML element.

Excluding Analysis Information

You can exclude text areas or entire elements from analysis.

Excluding Text Areas

You can exclude parts of a document from the Requirements Gateway analysis. For example, you might want to exclude some information because it contains content that Requirements Gateway might capture as traceability information, even though you do not want the information to be captured.

Complete the following steps to exclude some text in a document.

1. Select the root node of a custom type in the Types List of the Types pane and click the **Advanced Options** tab.

»

2. Click the submenu arrow, shown at left, to the right of the **Expressions to exclude text** control.
3. Select the **Add a new expression** menu item to add a new default expression to the **Expressions to exclude text** control.
4. Edit the default expression in the free text control just below the **Expressions to exclude text** field.

   For example, the expression `<<<[\S\s]*?>>>` excludes part of the document located between <<< and >>> tags. When Requirements Gateway analyzes the input documents with the custom type applied, Requirements Gateway ignores the text located between the <<< and >>> tags.

You can delete an expression by right-clicking the expression in the **Expressions to exclude text** control and selecting **Delete expression** from the context menu.

Excluding Elements

You can also exclude entire elements from the analysis. For example, you might want to keep obsolete requirements in your documents, but you do not want Requirements Gateway to capture the obsolete requirements during analysis, even if the obsolete requirements use the same syntax as current valid requirements.

Complete the following steps to exclude an entire element.

1. Select the root node of a custom type in the Types List of the Types pane and click the **Advanced Options** tab.
2. Enter an expression in the **Expression to delete an element** control. For example, the expression `(REQ_\d+)\s*\(Obsolete\)` ignores all the requirements with an identifier `REQ_nn` followed by `(Obsolete)`.

If you exclude elements from analysis, Requirements Gateway displays a message in the following locations of the main window:

- Messages tab of the Coverage Analysis and Impact Analysis views
- Messages pane of the Graphical and Requirement Details views

If you do not correctly define the expression and Requirements Gateway captures an element that does not correspond to a project element, Requirements Gateway activates the **`Impossible to delete`** error message. Refer to the *NI Requirements Gateway Help* for more information about error messages.

Note If you exclude requirement elements, Requirements Gateway also ignores all elements attached to requirements, such as attributes and text.

3

Using and Testing Regular Expressions

Requirements Gateway uses regular expressions similar to Perl to capture traceability information in the intermediate files. Regular expressions allow Requirements Gateway to manage a wide set of requirement standards and define multiple ways to identify requirements and other elements.

Expressions might look complex or difficult to read, but you can simplify the process of writing expressions by first drafting them in natural language. Then, you can use the regular expression reference in this chapter to transpose the natural language keywords into expressions that Requirements Gateway can understand.

You can also use the Expressions pane of the Configuration dialog box to test your regular expressions.

Components of Regular Expressions

Regular expressions can contain multiple components that capture specific information from intermediate files.

Matching Explicit Strings

You can match a fixed string by writing the string itself in the expression. For example, if you want to analyze `REQ_1`, `REQ_2`, the corresponding fixed string is `REQ_`.

Matching Character Sets

Use the following character escapes to match variable characters:

- **\d**—Corresponds to a digit character
- **\w**—Corresponds to alphanumeric word characters
- **.(dot)**—Corresponds to any single character except line break characters

Evaluating Occurrences

The following standard quantifiers recognize the different occurrences of characters or sub-expressions:

- *****—Indicates zero or more occurrences
- **+**—Indicates one or more occurrences
- **?**—Indicates zero or one occurrences

Specifying Location

The following boundary indicators identify the location of a matched sub-expression:

- **^**—Indicates the beginning of the line
- **$**—Indicates the end of the line

Specifying Alternatives

Use the vertical line (`|`) character as an alternate operator to match one of a set of options. For example, if you specify `TestResult=(OK|KO)`, then `(OK|KO)` would match either `OK` or `KO`.

You must use grouping with the alternate operator because it has the lowest precedence of any regular expression operator.

Identifying Separators

Requirements Gateway recognizes the following separator characters:

- **\s**—Matches a white space character. This is helpful for Word documents and is more generic than simply typing a space character, because Word suggests, and often automatically applies, several types of spaces.
- **\S**—Matches a non-white space character and specifies that you are not attempting to use a separator.
- **\t**—Matches a tab character. The converter for Word documents automatically inserts a tab character in the intermediate file between the name of the Word style and the text, so the `\t` character helps separate the Word styles.

Matching Special Characters

Some characters have special meanings. Thus, in order for Requirements Gateway to recognize the characters `(`, `)`, `^`, `|`, `+`, `*`, `[`, and `]` as inputs instead of special characters, you must escape them.

Use the back slash (\) character before special characters to escape the meaning of the special characters. For example, to match an opening bracket, use `\(`.

Excluding Matching of Some Characters

Use a caret (^) character immediately after the opening of an expression to negate the character set and match all characters except those that the expression contains.

For example, to specify a formalism such as a fixed string `REQ_` followed by one or more characters except a space or a comma, use the `(REQ_[^\s\,]+)` expression.

Capturing Sub-expressions

You must group sub-expressions between brackets if you want to capture the information sub-expressions define.

Use ellipses `(…)` to delimit a group. Requirements Gateway numbers captured groups by counting the opening parentheses from left to right.

In Requirements Gateway, a group is also called a field. You can define expressions that contain several fields. You can also analyze sub-expressions of groups. Refer to the *Section* section of Chapter 2, *Customizing Types and Type Elements*, for more information about sub-expressions.

Matching Information from a Table

When Requirements Gateway converts tables for intermediate files, the exact syntax of the conversion varies according to the source files.

Converting Word Tables

When Requirements Gateway converts a table in Word, the intermediate file formats the table information as |`<column number> <word style> <Column content>`, where the vertical line (|) character indicates the beginning of a table description.

For example, if Word outputs an intermediate file that reads `|1 Normal REQ_1 |`2…, you can specify a regular expression, `\|1 Normal (REQ_\d+)`, where the user wants to capture the fixed string `REQ_`followed by one or more digits in the first column.

Refer to the *Coupling Microsoft Word with NI Requirements Gateway* document for more information about converting Word tables.

Converting Excel Tables

When Requirements Gateway converts a table in Excel, the intermediate file formats the table information as `|<column number> <Column content>`, where the vertical line (`|`) character indicates the beginning of a table description.

For example, if Excel outputs an intermediate file that reads `|1 REQ_1 |2…`, you can specify a regular expression, `\|1 (REQ_\d+)`, where the user wants to capture the fixed string `REQ_` followed by one or more digits in the first column.

Refer to the *Coupling Microsoft Excel with NI Requirements Gateway* document for more information about converting Excel tables.

Capturing Multiple Lines

Use the characters `[\s\S]*` to capture information from multiple lines. In some default types, Requirements Gateway uses this expression to capture text.

Using Question Marks

Use the question mark (`?`) character to express an alternative without capturing the expression itself as a result and to manage the scope of analysis.

Expressing an Alternative without Capturing Expression Itself

Use brackets to capture groups or specify alternative groups. For most of the alternative groups, you do not want to capture the alternative as a result. Use the `?:` characters to avoid this capture.

For example, if you have the expression `TestDuration=(\d+) ms - Result=(?:Passed|OK)`, you want to capture the test duration value if the test result value is `OK` or `Passed`. You can express this alternative condition using `(OK|Passed)`, but capturing this sub-expression is not very useful. Use `?:`, as in `(?:Passed|OK)`, to avoid this result.

If you have the expression `(REQ_(?:FUNCT|PERF)_\d+)`, you want to capture only Functional requirements `REQ_FUNCT_xx` and Performance requirements `REQ_PERF_xx`. You do not want to capture only `FUNCT` or `PERF`.

Note The second example is for instructive purposes only. National Instruments recommends that you define two kinds of requirement elements in your custom type of analysis. Create one requirement element for functional requirements, using an expression such as `(REQ_FUNCT_\d+)`, and create another for performance requirements, using an expression such as `(REQ_PERF_\d+)`.

Managing the Scope of Analysis

During analysis, you might find several occurrences of a given regular expression. You can use a question mark character to restrict the scope of analysis. Table 3-1 shows an example of an Excel table that you must analyze.

Table 3-1. Sample Excel Table

Requirement	Value	Priority Level	Comment
REQ: Max Response Time	10 ms	A	—
REQ: Max Output Level	6 V	A	5 V +/–20%

The requirement formalism is defined as `REQ:` followed by a label in natural language. The only formalism you can specify is that the information is in the first column, as the expression `|1 REQ: Max Response Time |2 10 ms |3 A` specifies.

If you use the expression `\|1 (REQ:.+)\|` to restrict the analysis to only the requirement and value, but the scope remains too large because the analyzed information contains several occurrences of the vertical line (`|`) character. You can use the expression `\|1 (REQ:.+?)\|` to limit the analysis to only the first occurrence of `|1 REQ: Max Response Time |`.

Use brackets to specify that you want only `REQ: Max Response Time` from this string.

Note You can also use the expression `\|1 (REQ:[^\|]+)` to capture the information you need. In this expression, `\|1` specifies the first column, the string `REQ:` followed by any character except the `[^ ]` characters specifies the mark for the next column `\|`, and the plus (+) character specifies multiple times.

Frequently Used Expressions

This section summarizes the most commonly used regular expressions. The first column of each table lists the text you want to capture. The second column proposes regular expressions or sub-expressions that can match the text.

Table 3-2 shows common expression for capturing fixed strings, such as requirement IDs.

Table 3-2. Frequently Used Expressions for Fixed Strings

Fixed Strings	Matching Regular Expression
REQ_1, REQ_2 (fixed string followed by digits)	(REQ_\d+)
REQ_FUNCT_1 (fixed string followed by digits) REQ_PERF_1	Create two requirement elements for the type and use the following regular expressions: • (REQ_FUNCT_\d+) • (REQ_PERF_\d+)
Other fixed string followed by digits. Because IDs can indicate the concept of categories, you might consider several kinds of requirements. You might want to capture categories as additional information.	You can define an attribute for requirements by using an alternate operator (|) to capture the information included in the ID, as shown in the following examples: • Attribute Category using the expression REQ_(FUNCT|PERF)_\d+ • Attribute Functional using the expression REQ_(FUNCT)__\d+ • Another attribute Performance using the expression REQ_(PERF)_\d+
SRS_LED-AA-Product_123 (fixed string SRS_ terminated by a _nn, where nn indicates digits, but with the undefined characters between SRS_ and _nn)	If you know that no space or no separator exists in these characters, you can use the expression (SRS_\S+_\d+). If you do not know whether any spaces or separators exist in these characters, you can use the expression (SRS_.+?_\d+). You can also use a general expression, such as (SRS_\w+), if you know that SRS_ is followed by digits, letters, or an underscore (_), but not by a space or special character.

Table 3-3 shows common expressions for capturing attributes.

Table 3-3. Frequently Used Expressions for Capturing Attributes

Attributes Examples	Matching Regular Expression
Extract from a Test log file: • `Test: Scenario1 - Result = OK` • `Test: Scenario 2 - Result = KO`	An attribute Result with `Result = (OK | KO)`, or an attribute Passed with `Result = (OK)` and an attribute Failed with `Result = (KO)`
Extract from a Specifications document • `Allocation = xxx` • `Priority = High`	An attribute Allocation with `Allocation = (.+)` and an attribute Priority with `Priority = (High|Medium|Low)`. For the attribute Priority you can also use multiple attribute values. Refer to the *NI Requirements Gateway Help* for more information about attribute values.

Table 3-4 shows common expressions for capturing references.

Table 3-4. Frequently Used Expressions for Capturing References

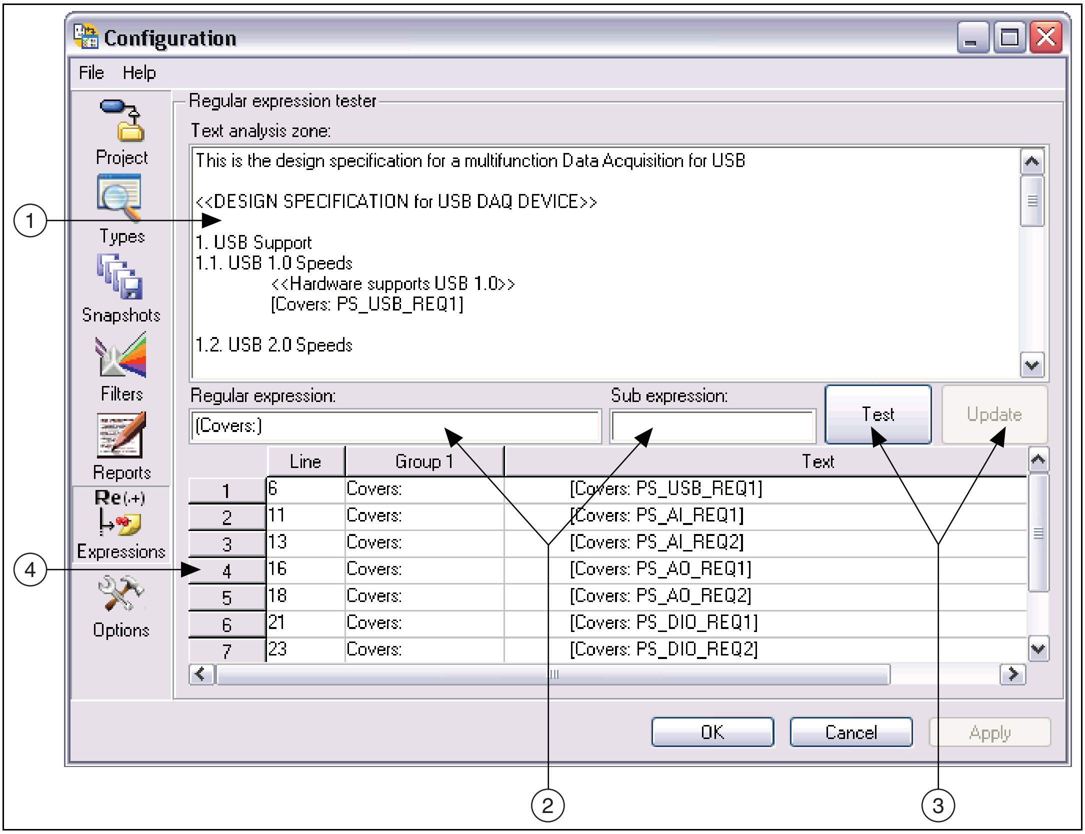
Reference Examples	Matching Regular Expression
[Covers *<OneReqID>*]	`\[Covers ([^\]]+)`
`Covered Requirements: REQ1, REQ2, ...`	`Covered Requirements: (.+)`, and a sub-regular expression `(REQ\d+)`
`Covered Requirements: REQ1, REQ_A_2, ...`	`Covered Requirements: (.+)`, and a sub-regular expression `(REQ\w+)`

Testing Regular Expressions

Use the Expressions pane of the Configuration dialog box to help you test and validate the regular expressions you define for the types you create on the Types pane. Testing ensures that the expressions accurately capture the appropriate elements before you use the expressions live in a project.

Requirements Gateway does not provide a button or menu item in the main window to access the Expressions pane of the Configuration dialog box. Instead, you must use the options on the main window toolbar or the options in the main window File menu to first launch the Configuration dialog box, and then click the **Expressions** button to access the Expressions pane.

You can also click the **Test regular expression** button, shown at left, next to the regular expression controls on the Types pane to launch Expressions pane with the controls populated with the defined regular expression, as shown in Figure 3-1.



Text analysis zone:

This is the design specification for a multifunction Data Acquisition for USB

<<DESIGN SPECIFICATION for USB DAQ DEVICE>>

1. USB Support
1.1. USB 1.0 Speeds
<<Hardware supports USB 1.0>>
[Covers: PS_USB_REQ1]

1.2. USB 2.0 Speeds

Regular expression: (Covers:)

Sub expression:

	Line	Group 1	Text
1	6	Covers:	[Covers: PS_USB_REQ1]
2	11	Covers:	[Covers: PS_AI_REQ1]
3	13	Covers:	[Covers: PS_AI_REQ2]
4	16	Covers:	[Covers: PS_AO_REQ1]
5	18	Covers:	[Covers: PS_AO_REQ2]
6	21	Covers:	[Covers: PS_DIO_REQ1]
7	23	Covers:	[Covers: PS_DIO_REQ2]

1 Text Analysis Zone	2 Regular Expression Controls	3 Toolbar	4 Results Table

Figure 3-1. Expressions Pane of the Configuration Dialog Box

The Expressions pane contains the following sections:

- **Text Analysis Zone**—A free text control in which you can enter a body of text that you can use to test your regular expressions.
- **Regular Expression Controls**—The regular expression and sub-expression that you want to test in the body of text.
- **Toolbar**—Buttons for testing and updating the regular expressions.
- **Results Table**—The results of the tested expressions in the source text.

Refer to the *NI Requirements Gateway Help* for more information about the options on the Expressions pane.

Populating the Text Analysis Zone

Use the Text analysis zone to check your regular expressions against existing source text to verify that the expressions capture the appropriate information. You can manually enter text in the Text analysis zone, or you can import the text from a source file. Importing converts the source file to a temporary intermediate file so you can test how your expressions behave in a project analysis before you use them in a live project.

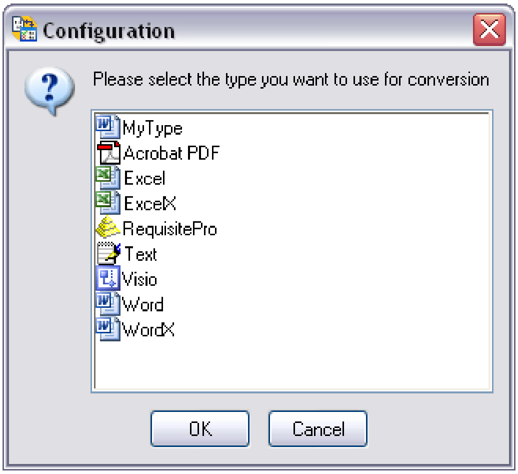
Use the following options in the File menu on the Expressions pane to import source text:

- **Import file**—Imports source information from a file such as a Word, document or Excel spreadsheet.
- **Import directory**—Imports source information from multiple documents in the same directory. Types such as Code and MultiWord support multiple documents.
- **Import intermediate file**—Imports the source information from an existing intermediate file, usually from databases such as RequisitePro.

Importing a File

Complete the following steps to import a single file and populate the Text analysis zone with the source file text.

1. On the Expressions pane, select **File»Import file** to launch the Configuration dialog box for importing files, as shown in Figure 3-2.



Configuration

Please select the type you want to use for conversion

MyType
Acrobat PDF
Excel
ExcelX
RequisitePro
Text
Visio
Word
WordX

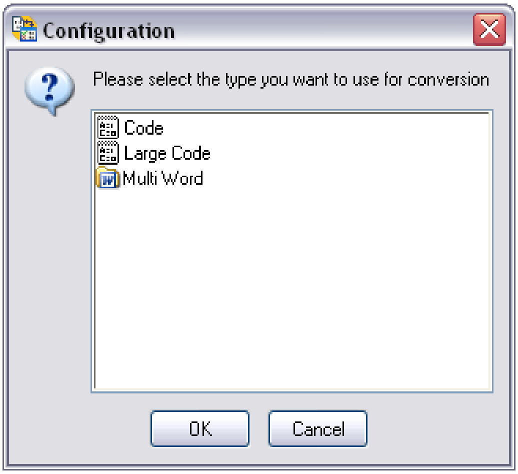
Figure 3-2. Import File Dialog Box

2. Select the type of document you want to convert to an intermediate file and import. You can select from all the built-in and custom types that support individual files.
3. Click **OK** to apply the type and launch the Open dialog box. After you select the type, the Open dialog box filters all the source files in the project directory to show only the files of the type you select.
4. Browse to the file you want to analyze, and click **Open**. The Text analysis zone populates with the intermediate file converted from the selected file.

Importing a Directory

Complete the following steps to import source text from multiple documents in the same directory and populate the Text analysis zone with the combined text from all documents in the directory.

1. Select **File»Import directory** to launch the Configuration dialog box for importing files from a directory, as shown in Figure 3-3.



Please select the type you want to use for conversion

Figure 3-3. Import Directory Dialog Box

2. Select the type of documents that you want to convert to a single intermediate file. You can select from all the built-in types that support multiple documents, including Code, Large Code, and MultiWord.
3. Click **OK** to apply the type and launch the Browse for Folder dialog box, which contains a tree view of all the directories on the computer.
4. Select the directory that contains the files you want to analyze, and click **OK** to launch a second Configuration dialog box that contains all the files of the selected type in the selected directory.
5. Select the file you want to analyze, and click **OK**. The Text analysis zone populates with the intermediate file converted from the selected file.

Importing an Intermediate File

Complete the following steps to import an existing intermediate file and populate the Text analysis zone with the intermediate file text.

1. Select **File»Import intermediate file** to launch the Open dialog box. You can select only intermediate files with the `.txt` extension.
2. Browse to the file you want to analyze, and click **Open**. The Text analysis zone populates with the text of the intermediate file.

Note You cannot import types that do not support text-based documents, such as LabVIEW, TestStand, Access, and DOORS. If you want to test regular expressions for such types, you must create an intermediate file for the types and use the **File»Import intermediate file** option.

Testing Expressions

After you import source information into the Text analysis zone, you can test regular expressions against the content of the imported file.

Complete the following steps to test a regular expression against the imported source file.

1. Enter a regular expression in the **Regular expression** control and, if necessary, a sub-regular expression in the **Sub expression** control. You can automatically populate the expression controls by clicking the **Test regular expression** button next to the regular expression options on the Types pane.
2. Click the **Test** button.

   Requirements Gateway tests the expressions you entered against the imported source file. If the expression does not match anything in the imported text, the message **`Text doesn't contain any correspondence with the regular expression`** displays just below the regular expression controls.

   If the expression matches elements of the analyzed text, the tester displays the results in the Results Table at the bottom of the Expressions pane. The table contains the following information:

   - **Line**—The line number in the imported file where Requirements Gateway detects the matching text. You can click the line number row header to highlight the corresponding line in the source text in the Text analysis zone.

- **Group1**—The first captured field that corresponds to the first part of the regular expression you defined between brackets.
- **Text**—The full text that the regular expression captures.

Updating Expressions

If you test a regular expression and do not match any information in the source file, modify the tested expressions in the **Regular expression** and **Sub expression** controls until you obtain the result you want.

If you clicked the **Test regular expression** button on the Types pane to launch the Expressions pane with the expression controls already populated, and then modified those expressions to correctly capture information, you can click the **Update** button on the Expressions pane to replace the expressions you imported from the Types pane with the correct versions.

Note The Update button is available only when you click the Test regular expressions button on the Types pane.

4

Customizing Reports

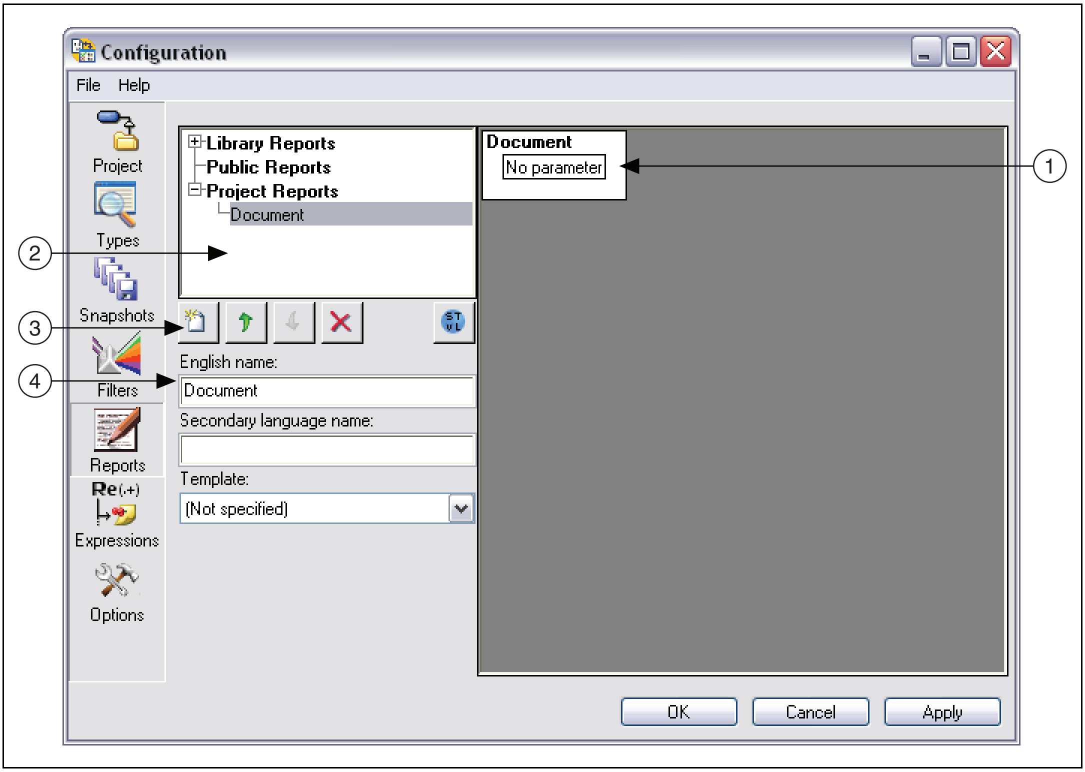
Requirements Gateway can generate Rich Text Format (RTF), HTML, PDF, and PostScript (PS) traceability reports. Use the Reports pane of the Configuration dialog box to construct models of customized reports that you can generate from the main window.

The report structure is an organization of formatting elements and Requirements Gateway traceability elements, such as attributes, requirements, and references.

Requirements Gateway saves the default read-only report structures as XML files in the `<Requirements Gateway>\config\doc_models` directory. You can use the options on the Reports pane to create custom reports from scratch or duplicate and modify existing reports, and to make them available for all projects or for a specific project.

Note This chapter discusses how to customize the report structure using the Reports pane. Refer to Chapter 4, *Generating Reports*, of the *Getting Started with NI Requirements Gateway* manual for tutorials on generating reports. Refer to the *NI Requirements Gateway Help* for more information about the content and purpose of each default report.

Select **Reports»Edit Reports** in the main window to launch the Reports pane, as shown in Figure 4-1.



English name: Document

Secondary language name:

Template: (Not specified)

1	Report	2	Report List	3	Toolbar	4	Properties

Figure 4-1. Reports Pane of Configuration Dialog Box

The Reports pane contains the following sections:

- **Report**—A graphical structure of the active report selected in the Report List.
- **Report List**—Lists all the available library, project, and public reports.
  - **Library Reports**—Default read-only report types available for all projects, stored in `<Requirements Gateway>\config\doc_models\Library Reports.xml`.
  - **Public Reports**—Custom reports available for all projects, stored in `<Requirements Gateway Public>\Config\doc_models\Public Reports.xml`.
  - **Project Reports**—Custom reports available only for a specific project, stored in the same directory as the project in `doc_models\Project Reports.xml`.

- **Toolbar**—Contains buttons to create, rearrange, and delete reports from the Report List. In addition, the **Report elements** button, shown at left, toggles the controls below the Report List to display the properties of the selected item in the report or the available elements to insert in the report. You can also double-click the Report elements button to launch a modal Report Elements window.
- **Properties**—Contains the following panes that display individually when you toggle the Report elements button:
  - **Selection Properties**—Options for configuring the item you select in the Report List or in the graphical report structure. The options available depend on the element you select.
  - **Report Elements**—Structures, data blocks, and other parameters that you can insert into the report. When you select an element of the report, valid elements you can insert into the selected element are highlighted with bold text.

Refer to the *NI Requirements Gateway Help* for more information about the options on the Reports pane.

Using the Graphical Report Structure

The right side of the Reports pane contains a graphical structure of the current report. You can drag and drop elements from the Report Elements pane to the desired location in the report structure. You can rearrange, modify, and delete report elements.

Note If the report is read-only, you cannot drag and drop elements into the report structure.

Figure 4-2 shows an example of a report as it appears in the graphical structure.

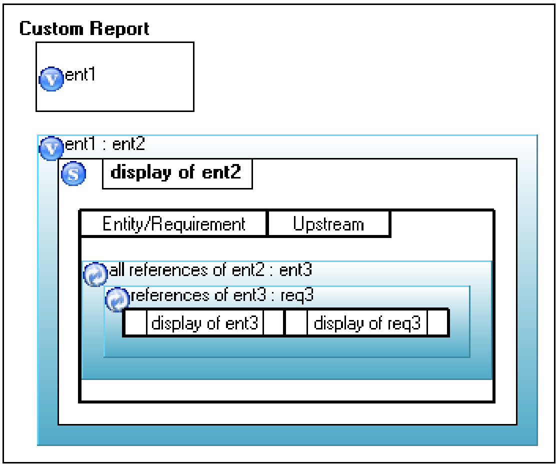


Figure 4-2. Example of a Custom Report in the Graphical Report Structure

When you select an element in the graphical structure, the relevant element is highlighted with hatching and Requirements Gateway automatically refreshes the options available on the Selection Properties pane.

Creating Reports

Requirements Gateway distinguishes between library reports and project reports. Library reports are the default report types that are available for all projects. Project reports are custom reports that you create from scratch or by duplicating and modifying an existing report.

Creating a New Report

Complete the following steps to create a new project report from scratch.

1. On the Reports pane, click the **New report** button, shown at left, on the toolbar to create a new blank report at the end of the Report List. The new untitled report appears in the graphical structure.
2. Enter a name for the report in the **English name** control. This is the name that appears as an available report in the Reports menu in the main window.

Duplicating an Existing Report

If the report you want to create is similar to an existing library report, you can duplicate the library report and modify only the elements you want to change. The library reports are read-only, so you can only modify the library reports by duplicating them.

Complete the following steps to duplicate an existing library report.

1. Expand the **Library Reports** node in the Report List and select the report you want to duplicate.
2. Right-click the selected report and select **Duplicate** from the context menu to create a new report at the end of the Report List. The new report has the same content as the built-in report, but appears under the Project Reports folder because you can modify the duplicate.
3. Enter a unique name for the duplicated report in the **English name** control.

Report Elements

When you edit the structure of a report, you can add structural elements, data, and parameters. Click the **Report elements** button to switch from the Selection Properties pane to the Report Elements pane, which contains the following tabs:

- **Structures**—Contains elements that define the structure of the report, such as text, paragraph, list, or table.
- **Data**—Contains elements associated with the object you select in the report, such as the project, a document, or a requirement.
- **Parameters**—Contains data type elements you can pass to a structure element.

Structures

Structures are formatting elements that you can use to build the overall report structure. Use the Structures tab of the Report Elements pane to drag and drop structures into a report.

You can insert the following structures:

- Anchor
- Code
- Generation
- Include File

- List
- Table row
- Cross reference
- List item
- Image
- Paragraph
- Table
- Section
- URL
- Text
- Code OTScript

Inserting a Structure into a Report

Complete the following steps to insert a structure into a report.

1. If the Report Elements pane does not display, click the **Report elements** button to switch from the Selection Properties pane to the Report Elements pane.
2. Click the **Structures** tab.
3. Drag a structure element from the Structures tab into the graphical report structure.

   🛇 If you drag an element to a position that is not permitted for the selected element, Requirements Gateway displays a forbidden symbol, shown at left, to indicate that you cannot insert the element in the current position.

   ⊞ If you drag an element to a position that is permitted for the selected element, Requirements Gateway displays a plus icon, shown at left, to indicate that you can add the element in the current position.

You cannot insert structure elements anywhere you want. You must nest some structures within other structures to use them. Table 4-1 summarizes the locations allowed for inserting each structure element.

Table 4-1. Locations Allowed for Inserting Structures

Structure	Can be inserted in
Code	Section
Include file	Paragraph
List	Document, section
Table row	Table
Cross reference	Paragraph
List item	List
Image	Document, section
Paragraph	Document, section, list item
Table	Document, section, table row
Section	Document, section
URL	Paragraph
Text	Cross reference, URL, paragraph, include file, section, title, title header of a table
Code OTScript	Cross reference, URL, paragraph, include file, section, title, title header of a table

When you select an element in the graphical structure, Requirements Gateway highlights the element with hatching and the Report Elements pane shows the elements available to drop into the selected element with bold.

Paragraph

To insert a paragraph into a document, drag a **paragraph** element from the Structures tab to the desired position in the report. A paragraph cannot contain another paragraph.

Section

Complete the following steps to insert a section in a report.

1. Drag a **section** element from the Structures tab to the desired position in the report. You can only add a section to the root document element. A section can contain another section.
2. To name the section, drag a **text** element into the `No title` area of the section.
3. Click the **Report elements** button to switch to the Selection Properties pane and enter a name for the section in the **English text** control.

   Figure 4-3 shows an example of a Section element named `Section Title`.

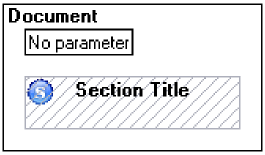


Figure 4-3. Section Element with an Entered Title

Refer to the *Text* section for more information about inserting text.

Table

Complete the following steps to insert a table in a report.

1. Drag a **table** element from the Structures tab to the desired position in a document or section. By default, a table contains two columns. You can define a different number of columns on the Selection Properties pane.
2. Click the **Report elements** button to switch to the Selection Properties pane and specify the number of columns you want in the table in the **Number of columns** control.
3. Next, to add rows to the table, click the **Report elements** button again to return to the Report Elements pane and click the **Structures** tab. Drag a **table row** element from the Structures tab to the inserted table. You can add as many rows as you want.

Figure 4-4 shows an example of a table with three columns and two rows.

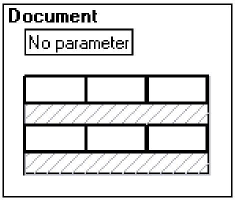
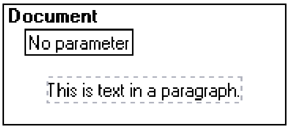


Figure 4-4. Table with Three Columns and Two Rows

Text

Complete the following steps to insert text in an applicable structure element.

1. Drag a **text** element from the Structures tab to the desired position in a relevant structure element, such as a paragraph.
2. Click the **Report Elements** button to switch to the Selection Properties pane.
3. Enter the text you want to display in the **English text** control. Figure 4-5 shows an example of text entered for a paragraph element.



This is text in a paragraph.

Figure 4-5. Text Added to a Paragraph Element

Image

Complete the following steps to insert an image in a report.

1. Drag an **image** element from the Structures tab to the desired position in a document or section.
2. An image element expects image data. Click the **Data** tab and drag an image data type, which uses the icon shown at left, into the image element.

   Refer to the *Data* section for more information about adding data to a report.

List and List Item

Complete the following steps to insert a list of items into a report.

1. Drag a **list** element from the Structures tab to the desired position in a document or section.
2. To add an item to the list, drag a **list item** element from the Structures tab to the list, as shown in Figure 4-6. You can add as many items as you want.

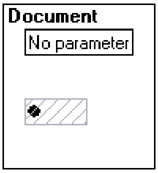


Figure 4-6. List Item Element Added to List Element

You cannot create lists with multiple levels. To add information to a list item, you must add a paragraph to the list item, and include the information to the paragraph element. Refer to the *Paragraph* section for more information about adding paragraph elements.

Code

A code element is a paragraph that appears in Courier New font in the generated report. To insert a code element into a document, drag a **code** element from the Structures tab to the desired position in a section, as shown in Figure 4-7.

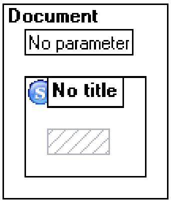


Figure 4-7. Code Element Added to a Section Element

OTScript Code

Complete the following steps to insert an OTScript code into a report.

1. Drag a **code OTScript** element from the Structures tab into a paragraph.

2. Click the **Report Elements** button to switch to the Selection Properties pane.
3. Enter the OTScript code you want Requirements Gateway to evaluate when generating the report in the **Code** control. If the OTScript you enter is erroneous, the Code control displays the code in red, and a message appears below the Code control that explains the error.

Refer to Chapter 5, *Using OTScript*, for more information about using OTScript code.

Include File

Complete the following steps to insert an include file into a report so you can include the content of one document in a different document.

1. Drag an **include file** element from the Structures tab into a paragraph.
2. Click the **Data** tab and drag a data element into this area, which corresponds to the path of a file that will be included. The master document must be able to support the included document.

   In the generated report, the included file appears as a hyperlink.

URL

A URL element appears as a hyperlink in the generated report. Complete the following steps to insert a URL in a report.

1. Drag a **URL** element from the Structures tab into a paragraph.
2. Click the **Report Elements** button to switch to the Selection Properties pane.
3. Enter the URL you want to insert into the report in the **Url** control.

Cross Reference

Complete the following steps to insert a cross reference into a report.

1. Drag a **cross-reference** element from the Structures tab into a paragraph.
2. Click the **Report Elements** button to switch to the Selection Properties pane.
3. Select a target element for the cross reference from the **Target** ring control. The options available in the ring control vary according to the elements you have placed in the report.

Data

For some structure elements, you must insert specific data to precisely define the content of the element. Requirements Gateway interprets the data while generating the report. Use the Data tab of the Report Elements to drag and drop data elements into a report.

Processing Data

The Data tab of the Report Elements pane shows a tree view in which each node gathers its authorized shared sub-data into the following categories, which vary according to the selected node:

- Computation
- Contents
- General information
- Identification information
- Impact matrix
- Misc
- Project links
- Type

Requirements Gateway specifies terminal data and non-terminal data. Terminal data is represented by subordinate components of the tree view, and non-terminal data is represented by nodes in the tree view. When you insert non-terminal data into the report, Requirements Gateway automatically creates a variable or iterator in the Data list. The iterator name is generated from the data element name. For example, an iterator created from a cover link data element is `covn`, where `n` is a number greater than zero. The iterator appears as a new node in the Data list and includes its own categories of sub-data.

In addition, each data element has a data type associated with it. Table 4-2 shows the different data types, specifies whether the type is terminal or non-terminal, and specifies additional information about the type.

Table 4-2. Types of Data You Can Add in a Report Structure

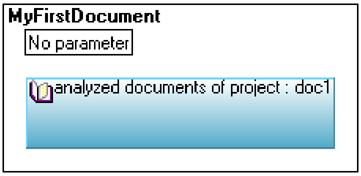
Data Icon	Data Type	Description
t	Text	Terminal element. You can drop a text data type only in a structure element that accepts a Text structure element. Use the text data type to insert text information into the report.
🖼	Image	Terminal element. You can drop an image data type only in an image structure element. Use the image data type to insert an image into the report.
	Attributes	Non-terminal element. The attribute data type represents attributes in Requirements Gateway.
	Cover Links	Non-terminal element. The cover links data type represents references in Requirements Gateway.
	Links	Non-terminal element. The links data type represents links in Requirements Gateway.
📖	Documents	Non-terminal element. The documents data type represents documents in Requirements Gateway.
	Filters	Non-terminal element. The filters data type represents filters in Requirements Gateway.
	Entities	Non-terminal element. The entities data type represents entities in Requirements Gateway.
	Requirements	Non-terminal element. The requirements data type represents requirements in Requirements Gateway.
📄	Sections	Non-terminal element. The sections data type represents sections in Requirements Gateway.
↻	Loop	Non-terminal element. Use the loop data type to insert and loop elements in addition to the elements already listed.

Example of Creating Data

This tutorial builds a document that contains a list of all project requirements to help you visualize how Requirements Gateway creates data for a report.

Complete the following steps to add data iterators to a new report.

1. Create a new report from scratch, as described in the *Creating a New Report* section.
2. On the Selection Properties pane, name the report `MyFirstDocument` in the **English name** control.
3. Click the **Report elements** button to switch to the Report Elements pane, and click the **Data** tab.
4. Expand the **Project»Contents** node and drag an **analyzed requirements** element to the report, as shown in Figure 4-8.



MyFirstDocument

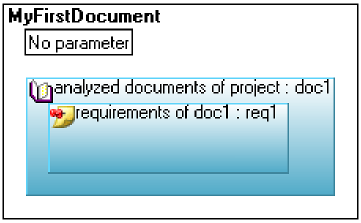
No parameter

analyzed documents of project : doc1

Figure 4-8. Analyzed Documents Element Added to MyFirstDocument

A new iterator **doc1** appears at the top of the tree on the Data tab.

5. Expand the **doc1»Contents** node and drag a **requirements** element to the **analyzed documents** element, as shown in Figure 4-9.



MyFirstDocument

No parameter

analyzed documents of project : doc1

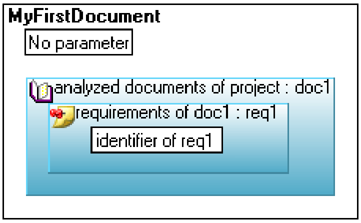
requirements of doc1 : req1

Figure 4-9. Requirements Element Added to the Analyzed Documents Element

A new iterator **req1** appears at the bottom of the tree on the Data tab.

6. Click the **Structures** tab and drag a **paragraph** element to the **requirements** element.

7. Click the **Data** tab and expand the **req1»Identification information** node. Drag an **identifier** element to the **paragraph** element, as shown in Figure 4-10.



MyFirstDocument

No parameter

analyzed documents of project : doc1

requirements of doc1 : req1

identifier of req1

Figure 4-10. Identifier Element Added to the Requirements Element

Parameters

Parameters correspond to traceability and project elements. Use the Parameters tab of the Report Elements pane to drag and drop parameters into a report. You can only add parameters to the parameter area in the root document element of the report. The parameter area displays `No parameter` until you add parameters to it.

You can insert the following parameters:

- Attribute
- Document
- Document/folder
- Element
- Entity
- Entity/section
- Folder
- Link
- Link attribute
- Macro-requirement
- Project
- Requirement
- Requirement/macro-requirement
- Section
- String

Inserting a Parameter into a Report

Complete the following steps to insert a parameter in a report structure.

1. If the Report Elements pane does not display, click the **Report elements** button to switch from the Selection Properties pane to the Report Elements pane.
2. Click the **Parameters** tab.
3. Drag a parameter element from the Parameters tab into a parameter area in a report element. The name of the parameter replaces the default `No parameter` text.
4. Click the **Report elements** button to switch to the Selection Properties pane and enter a message, if you want, in the **English message** control. The message appears when you generate the report and Requirements Gateway prompts you for parameter values.

Note You can drop only one parameter element into the parameter area, regardless of whether the parameter represents one element or multiple elements.

To delete a parameter from the parameter area, right-click the parameter and select **Delete** from the context menu.

Customizing Report Elements

Depending on the element you select in the graphical structure, some additional controls become available on the Selection Properties pane that you can use to improve the appearance of the generated report.

Report Element Options

Table 4-3 shows the options for structure or data elements that are available on the Selection Properties pane to parameterize the contents of the generated report in specific conditions.

Table 4-3. Report Element Options

Option	Available for	Purpose
With title	Document	Displays the report title in the generated report.
Loop when absent	Loop (data)	Displays the content of the loop even when no iteration element for the loop exists.
Display blank when multiple	Table row	Displays an empty table cell if an element is repeated from the previous line.
Non produced when empty	List, table, image, or section	Does not display an empty structure in the generated report.

Using Conditions

Most structure elements include a Condition control on the Selection Properties pane, which you can use to enter a condition written in OTScript so you can restrict a set of elements when generating the report.

Refer to Chapter 5, *Using OTScript*, for more information about writing OTScript conditions.

Creating a Complete Custom Report

This tutorial creates a restricted traceability matrix report for selected documents. The traceability matrix report describes the relationship between the requirements and coverage information for the requirements in the selected documents. Refer to the *NI Requirements Gateway Help* for more information about a traceability matrix report.

Complete the following steps to create the complete the report.

1. Click the **New Document** button to create a new report under the Project Reports folder.
2. Specify a name for the report, such as `MyNewDocument`, in the **English name** control.
3. Click the **Report elements** button to display the Report Elements pane.

4. Click the **Parameters** tab and drag a **Document/folder** parameter to replace the `No parameter` text that appears in the parameter area of the report by default with an **abstr1** parameter, as shown in Figure 4-11.

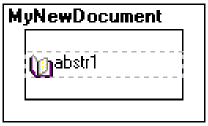


Figure 4-11. Document/folder Parameter Added to MyNewDocument

The new parameter allows you to analyze only specific documents in the generated report.

5. Click the **Report elements** button to switch to the Selection Properties pane, and rename the parameter as `Docs` in the **Parameter name** control.

Tip You can also double-click the **Report elements** button to launch a modal Report Elements window that contains the same options as the Report Elements pane so you do not have to click the Report elements button back and forth to toggle the options.

6. Switch back to the Report Elements pane and click the **Data** tab. A new corresponding iterator **Docs** appears at the top of the tree on the Data tab.

7. Expand the **Docs»Project links** node and drag a **document** element to the report, as shown in Figure 4-12.

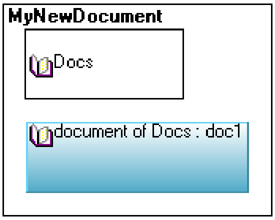
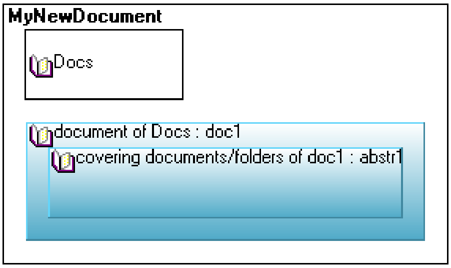


Figure 4-12. Document Element Added to MyNewDocument

A new iterator **doc1**, as defined on the document element, appears at the top of the tree on the Data tab.

8. Expand the **doc1»Project links** node and drag a **covering documents/folders** element to the **document** element, as shown in Figure 4-13.



MyNewDocument

Docs

document of Docs : doc1

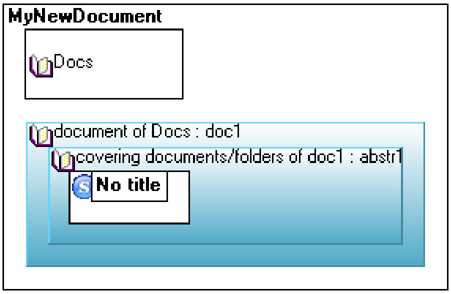
covering documents/folders of doc1 : abstr1

Figure 4-13. Covering Documents/Folder Element Added to the Document Element

A new iterator, **abstr1**, as defined on the documents and folders element, appears at the top of the tree on the Data tab.

Because you want to display only the downstream documents that cover the selected documents to which the report applies, add a condition to the covering documents/folders element that filters only the relevant documents. You can write this condition in OTScript for the covering documents/folders loop.

9. With the covering documents/folder element selected, switch to the Selection Properties pane.
10. Enter `abstr1 $< Docs` in the **Condition** control. The `$<` symbols specify belonging to a list.
11. Switch to the Report Elements pane and click the **Structures** tab. Drag a **section** element to the covering documents/folders element, as shown in Figure 4-14.



MyNewDocument

Docs

document of Docs : doc1

covering documents/folders of doc1 : abstr1

Figure 4-14. Section Element Added to the Covering Documents/Folders Element

The section element structures the report. Each created section displays the coverage analysis of the selected documents one by one.

12. Click the **Data** tab and expand the **doc1»Identification information** node. Drag a **name** element to the section title. The text `name of doc1` appears in the section title.

13. Expand the **abstr1»Identification information** node and drag the **name** element to the right of the section title, as shown in Figure 4-15.

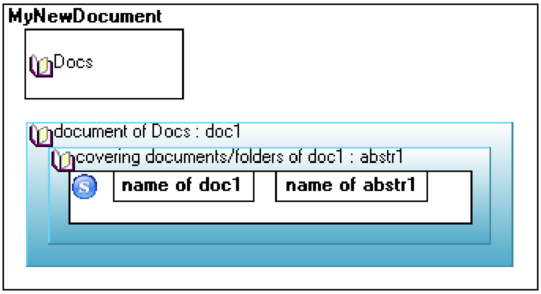


Figure 4-15. Names of doc1 and abstr1 Elements Added to Section Element

14. Click the **Structures** tab and drag a **text** element between the `name of doc1` and `name of abstr1` elements, as shown in Figure 4-16.

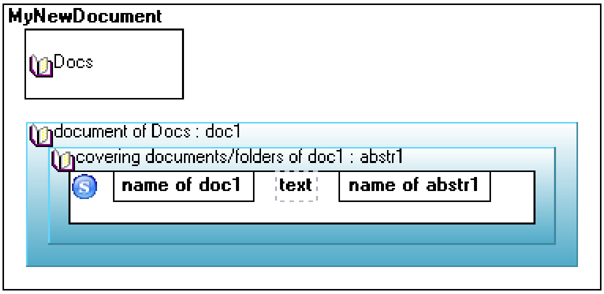


Figure 4-16. Text Element Added between Name Elements in Section Title

15. Switch to the Selection Properties pane and enter `is covered by` in the **English text** control.

16. Switch to the Report Elements pane and click the **Structures** tab. Drag a **paragraph** element to the section element below the name and text elements you added.

17. Drag a **text** element to the paragraph element. Switch to the Selection Properties pane and enter `Coverage ratio` in the **English text** control.

18. Switch to the Report Elements pane and click the **Data** tab. Expand the **doc1»Computation** node, and drag the **cover ratio (abstr1)** to the right of the Coverage ratio text element, as shown in Figure 4-17.

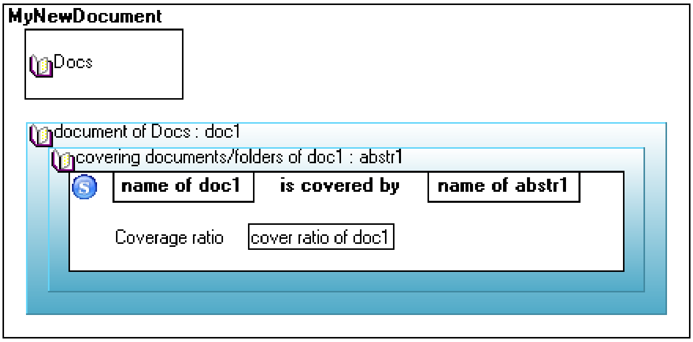


Figure 4-17. Coverage Ratio Information Added to the Section Element

19. Click the **Structures** tab and drag a **table** element to the section element below the Coverage ratio text element. Switch to the Selection Properties pane and enter `4` in the **Number of columns** control.
20. Switch to the Report Elements pane and click the **Structures** tab. Drag a **text** element to each cell of the table header.
21. Switch to the Selection Properties pane and enter `Upstream`, `Description`, `Downstream`, and `Description`, respectively, in the **English text** control for each text element.
22. Switch to the Report Elements pane and click the **Data** tab. Expand the **doc1»Contents** node and drag the **requirements and macro-requirements** element to the table, as shown in Figure 4-18.

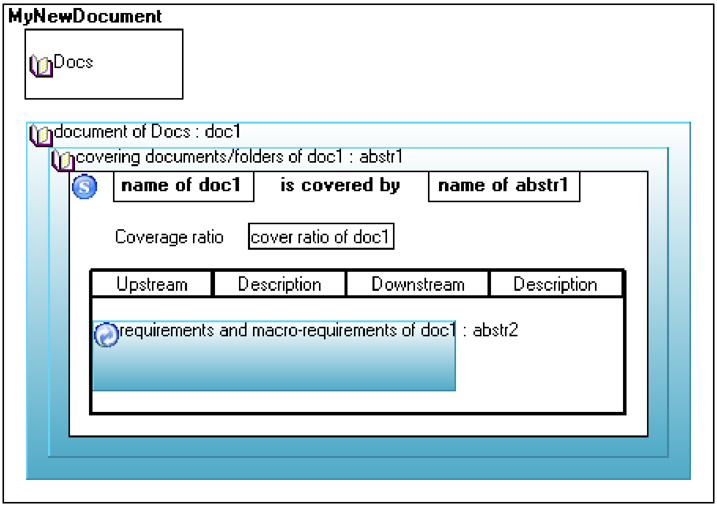


Figure 4-18. Requirements and Macro-Requirements Element Added to Table

A new iterator, **abstr2**, as defined in the requirements and macro-requirements element, appears at the bottom of the tree on the Data tab.

23. With the requirements and macro-requirements element selected, switch to the Selection Properties pane and rename the **abstr2** variable as `Reqs` in the **Variable name** control. This also renames the abstr2 node on the Data tab of the Report Elements pane as Reqs.

24. Now, we want to display the references that cover the requirements of the selected documents. Switch to the Report Elements pane and click the **Data** tab. Expand the **Reqs»Project links** node and drag the **all references** element to the requirements and macro-requirements element.

    A new **abstr2** iterator, as defined in the all references element, appears near the top of the tree on the Data tab.

25. With the all references element selected, switch to the Selection Properties pane and rename the **abstr2** variable as `Refs` in the **Variable name** control. This also renames the abstr2 node on the Data tab of the Report Elements pane as Refs.

26. To catch only references from the selected documents, enter the following OTScript condition in the **Condition** control: `(document = doc1) OR (root = doc1)`. Refer to Chapter 5, *Using OTScript*, for more information about writing OTScript conditions.

27. Switch to the Report Elements pane and click the **Structures** tab. Drag a **table row** element to the all references element.
28. Click the **Data** tab. Complete the following steps to add information to the table row.
    a. Expand the **Reqs»Identification information** node and drag the **identifier** element to the first cell and the **text** element to the second cell.
    b. Expand the **Refs»Identification information** node, and drag the **identifier** element to the third cell and the **text** element to the fourth cell.

    Figure 4-19 shows the completed report structure.

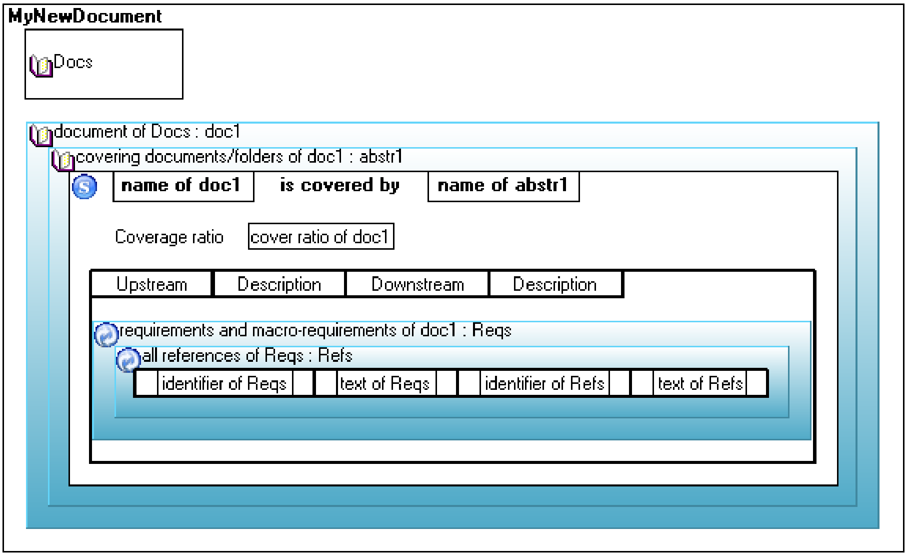


Figure 4-19. Completed Report Structure

29. Click **OK** to save the report.

5

Using OTScript

OTScript is a proprietary object-oriented language based on classes. When an object receives a message, a method associated with the message is executed. The result of the execution is returned as an answer of the object to the message. You typically use OTScript in Requirements Gateway to customize the elements you add to reports.

OTScript has very concise syntax, list manipulation capabilities, and native handling of binary associations, and includes a built-in rule checker and functions for describing a user interface. OTScript also heavily relies on types, so that the type of each expression must be known at compile time. The language provides some predefined types.

OTScript is a rich language, but you can use only a sub-part of this language in Requirements Gateway. Because OTScript is a proprietary language, this chapter provides only an overview of its capabilities.

General Syntax

OTScript is case-sensitive and supports the following primitive data types:

- Integer
- Real
- Boolean
- String

Reserved Words

OTScript considers the following words reserved:

- `True`
- `False`

Predefined Variables

The only predefined variable for OTScript in Requirements Gateway is `EACH`, which specifies the current receiver.

For example, `EACH.requirements` returns the list of all the requirements of a document when a document is selected.

Operators

You can use the following operators on integer and real elements:

- **Arithmetic**—Addition, subtraction, multiplication, division (`/`)
- **Comparison**—`<`, `>`, `<=`, `>=`, `=`, `/=` (not equal)

You can use the following operators on Boolean elements:

- **Comparison**—`=`
- **Boolean**—`AND`, `OR`, `NOT(bool)`

  For example:

  ```
  NOT(V)
  (if V=True then NOT(V) => False
  if V=False then NOT(V) => True)
  ```

You can use the following operators on string elements:

- **Concatenation**—`+`
- **Comparison**—`<`, `>`, `<=`, `>=`, `=`, `/=` (not equal)

Testing

OTScript provides functions that determine whether an object belongs to a class or list, if a list is empty, and if one string matches another.

Belonging to a Class

Use the function `ISA` between an object and a class, as in `obj1 ISA class1`, to test whether the object belongs to the class. The `ISA` function returns `True` if the object belongs to the class and `False` if not. Examples of classes include `Folder`, `Section`, and `Requirement`.

Belonging to a List

Use the function `$<` between an object and a list, as in `obj1 $< list1`, to test if the object belongs to the list. The `$<` function returns `True` if the object belongs the list and `False` if not.

Emptiness of a List

The function `$NO(list1)` tests if a list of a current document is empty, and returns `True` if the list is empty and `False` if not.

The function `$SOME(list1)` tests if a list of a current document is not empty, and returns `True` if the list is not empty and `False` if the list is empty.

String Matching

The function `STRMATCH(string1, string2)` tests if one string matches another. Use an asterisk (`*`) for 0-n characters and a pound sign (`#`) for one character. For example, `STRMATCH(doc1.name, "TestCard*")` tests if the name of `doc1` begins with the string `TestCard` string.

Control Structures

Use a period (`.`) between expressions to evaluate the first expression on the second expression, as in `<expr1>.<expr2>` . OTScript applies each expression to its preceding element. For example, `coveredDocuments.requirements.addedText` returns the text elements added to the requirements of the currently selected covered documents.

Use brackets (`[]`) around a condition following an expression, as in `<expr>[<cond>]`, to evaluate the expression and select only the elements that verify the condition. For example, `requirements[STRMATCH(label,"DS_*")]` returns the requirements whose label matches the string `DS_`.

Frequently Used Statements

Use the `$SOME` function followed by brackets (`[]`) around an expression to express a concept of existence and condition. For example, `$SOME(documents.leafRequirements[isUncovered])` returns `True` if documents exist that contain uncovered leaf requirements.

Use a combination of the `EACH` and `ISA` functions to express a concept of belonging. For example, `EACH ISA Folder` returns `True` only for an element that is a folder.

Writing OTScript Conditions for Custom Reports

In Requirements Gateway, use OTScript to specify conditions for the elements that you add to a custom report structure on the Reports pane of the Configuration dialog box. A condition is a Boolean, numeric, or string expression that is evaluated as `True` or `False`. If the condition is `NULL`, OTScript treats the condition as `False`.

Most report elements include a Condition control on the Selection Properties pane of the Report pane when you select the element in the graphical report structure.

When you write a condition, you must identify the purpose and entities implied in the condition. You must use OTScript methods to handle entities.

The OTScript syntax is checked as you type. The code you enter in the Condition control remains red as long as the code contains syntax errors.

Accessing the OTScript Methods

Requirements Gateway associates a dedicated OTScript method with each element that you can insert into a report. You can combine into conditions each method that depends on the context in which it is inserted.

The Data tab of the Report Elements pane of the Report pane shows all the methods that you can insert into an element in the report. If you hover the cursor over each method, the tooltip displays the following information:

- **Name**—The displayed name of the method.
- **Internal**—The internal name of the method that OTScript uses during evaluation.
- **Return**—The return type of the method.

For example, if you hover over a **requirements and macro-requirements** element, the tooltip displays the following:

- **Name**—requirements and macro-requirements
- **Internal**—abstractRequirements
- **Return**—Requirement/macro-requirement

For the **requirements and macro-requirement** data element, the OTScript method is `abstractRequirements`, which returns requirements and macro-requirements.

Refer to the *Data* section of Chapter 4, *Customizing Reports*, for more information about using the Data tab of the Report Elements pane.

Frequently Used Methods

Table 5-1 lists OTScript methods commonly used for writing conditions.

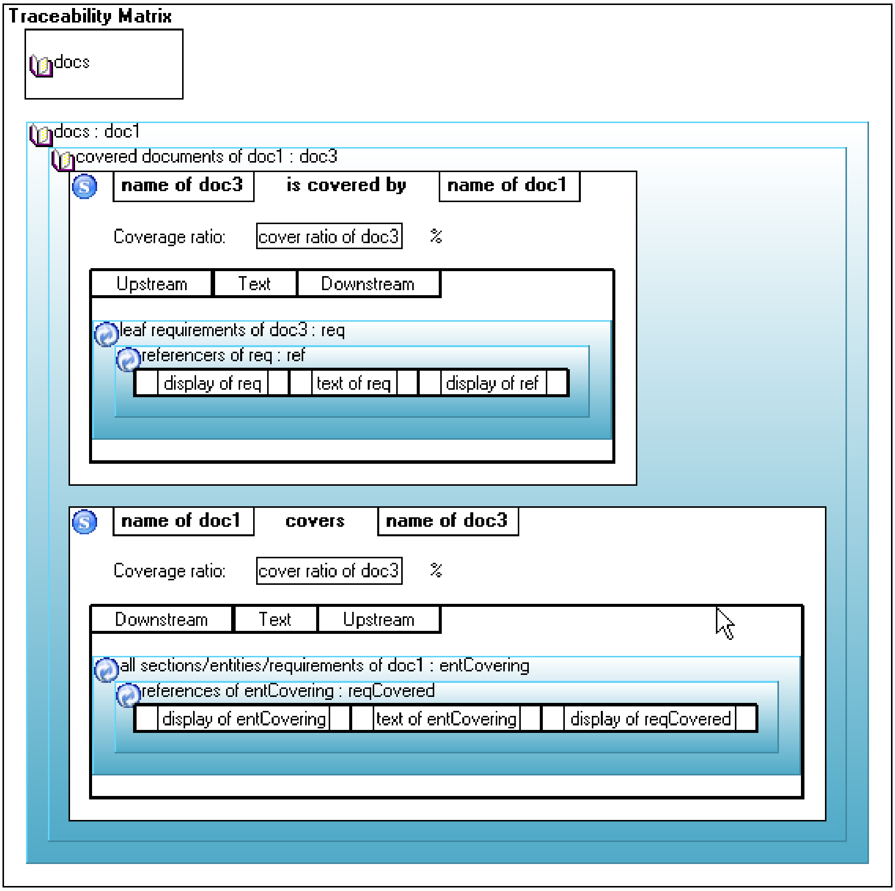
Table 5-1. Frequently Used Methods

Method	Description
`document`	Returns the document that contains the current element.
`isDerived`	Tests if the requirement is derived.
`isCovered`	Tests if the requirement is covered.
`isUncovered`	Tests if the requirement is uncovered.
`isFiltered`	Tests if the requirement is visible.
`root`	Tests if the root is a folder that contains the document. If the document is not contained in a folder, the root is the document itself.
`$SOME(attributes[typeName="Priority"])`	Tests if the requirement has an attribute named Priority.
`document ISA ModificationDocument`	Tests if the document is a modification document.
`STRMATCH(doc1.typeName,"*SRS*")`	Tests if the type of the document contains the string `SRS`.
`$CNT(coverLinks)>=2`	Tests if the requirement is covered at least twice.

Example of Using OTScript in a Report

Some of the built-in reports use OTScript conditions. This example analyzes the built-in traceability matrix report to show how you might use conditions. The traceability matrix report obtains the coverage of upstream documents by downstream documents.

On the Reports pane, expand the **Library Reports** node in the Report List and select **Traceability Matrix** to show the graphical structure of the traceability matrix report, as shown in Figure 5-1.



Traceability Matrix

docs

docs : doc1

covered documents of doc1 : doc3

name of doc3 | is covered by | name of doc1

Coverage ratio: cover ratio of doc3 %

Upstream | Text | Downstream

leaf requirements of doc3 : req

references of req : ref

display of req | text of req | display of ref

name of doc1 | covers | name of doc3

Coverage ratio: cover ratio of doc3 %

Downstream | Text | Upstream

all sections/entities/requirements of doc1 : entCovering

references of entCovering : reqCovered

display of entCovering | text of entCovering | display of reqCovered

Figure 5-1. Traceability Matrix Report Structure

Analyzing Sub-Parts of the Report

For the purpose of this example, the traceability matrix report contains the following important sub-parts:

- **docs**—The set of documents and folders you select for which you want to generate the traceability matrix report.
- **doc1**—An iterator to loop on docs elements.
- **doc3**—An iterator to loop on the set of documents which are covered by doc1.

For a given document `doc1`, the `covered documents` function returns all the documents that `doc1` covers. Only the covered documents that belong to the selected documents, `docs`, are relevant. You must specify a condition to reduce the scope of analyzed documents.

Writing a Condition

You must create a condition that compares the documents that result from the `covered documents` function with the selected documents. The condition must verify if a covered document, `doc3`, belongs to `docs`. Use the `$<` function to test if `doc3` belongs to the `docs` list. The resulting OTScript expression is `(doc3 $< docs).`

However, `docs` can contain both documents and folders. The condition must also verify if `doc3` must be inserted into a folder that belongs to the folders of `docs`. Use the `root` function to test if the folder that contains `doc3` must also belong to `docs`. The resulting OTScript expression is `(doc3.root $< docs).`

You can use the Boolean `OR` operator to concatenate the two expressions. The resulting OTScript expression is `(doc3 $< docs) OR (doc3.root $< docs).` You can verify this expression by selecting the **covered documents of doc1** element in the report and checking the expression in the Condition control on the Selection Properties pane.

6

Managing Customized Projects

When you create custom project information, such as report templates or models, you might want to reuse the custom information for other projects. Requirements Gateway provides several options for storing customized information.

When you create custom information for the current project, Requirements Gateway adds the following files and subdirectories to the root project directory:

- `<project name>.types`—Contains the custom types
- `<project name>.filters`—Contains the custom filters.
- `<doc_templates>`—Directory that contains the custom report format templates.
- `<doc_models>`—Directory that contains the custom report structures that you create on the Reports pane of the Configuration dialog box.

You can reuse this custom information in the following ways:

- You can copy all the files to an additional project directory to make the files available for only one other project.
- You can make the files available automatically for the default configuration of all projects by placing them in the `<Requirements Gateway Public>\Config` directory as follows:
  - Place types files in the `<Requirements Gateway Public>\Config\types\public` directory. If you create a subdirectory and add a `.types` file to the subdirectory, the directory appears as a folder on the Types List on the Types pane of the Configuration dialog box.
  - Place report template files in the `<Requirements Gateway Public>\Config\doc_templates` directory.
  - Place report model XML files in the `<Requirements Gateway Public>\Config\doc_models` directory.
  - Place bitmap icons and associated mask icons for custom types and type elements in the `<Requirements Gateway Public>\Config\images` directory.

Refer to the *NI Requirements Gateway Help* for more information about Requirements Gateway file types. Refer to the *NI Requirements Gateway Release Notes* for more information about persisting customizations when you upgrade to a new version of Requirements Gateway.

A

Technical Support and Professional Services

Visit the following sections of the award-winning National Instruments Web site at ni.com for technical support and professional services:

- **Support**—Technical support at ni.com/support includes the following resources:
  - **Self-Help Technical Resources**—For answers and solutions, visit ni.com/support for software drivers and updates, a searchable KnowledgeBase, product manuals, step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on. Registered users also receive access to the NI Discussion Forums at ni.com/forums. NI Applications Engineers make sure every question submitted online receives an answer.
  - **Standard Service Program Membership**—This program entitles members to direct access to NI Applications Engineers via phone and email for one-to-one technical support as well as exclusive access to on demand training modules via the Services Resource Center. NI offers complementary membership for a full year after purchase, after which you may renew to continue your benefits.

    For information about other technical support options in your area, visit ni.com/services, or contact your local office at ni.com/contact.
- **Training and Certification**—Visit ni.com/training for self-paced training, eLearning virtual classrooms, interactive CDs, and Certification program information. You also can register for instructor-led, hands-on courses at locations around the world.
- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, National Instruments Alliance Partner members can help. To learn more, call your local NI office or visit ni.com/alliance.

If you searched `ni.com` and could not find the answers you need, contact your local office or NI corporate headquarters. Phone numbers for our worldwide offices are listed at the front of this manual. You also can visit the Worldwide Offices section of `ni.com/niglobal` to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

Index