Requirements Management Interface

For Use with Simulink®

Modeling

Simulation

Implementation

The MATH WORKS Inc.

User's Guide

Version 1

How to Contact The MathWorks:

508-647-7000 Phone

508-647-7001 Fax

The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098 Mail

`http://www.mathworks.com`	Web
`ftp.mathworks.com`	Anonymous FTP server
`comp.soft-sys.matlab`	Newsgroup

`support@mathworks.com`	Technical support
`suggest@mathworks.com`	Product enhancement suggestions
`bugs@mathworks.com`	Bug reports
`doc@mathworks.com`	Documentation error reports
`subscribe@mathworks.com`	Subscribing user registration
`service@mathworks.com`	Order status, license renewals, passcodes
`info@mathworks.com`	Sales, pricing, and general information

Requirements Management Interface User's Guide

Printing History: January 2000 First printing New for Version 1 (Release 11)
September 2000 Online only Updated for Version 1 (Release 12)

Contents

Preface

1 Using the Requirements Management Interface

2 Using the Requirements Management Interface for QSS's DOORS

Preface

What the Requirements Management Interface Is

The Requirements Management Interface allows you to associate requirements with Simulink® models, Stateflow® diagrams, and MATLAB® M-files. There is a standard version of the Requirements Management Interface and a version that works with the DOORS requirements management system, a product offering from Quality Systems & Software, Inc. (QSS).

For the standard version, the requirements are in HTML, Microsoft Word, and Microsoft Excel documents. For the DOORS version, the requirements are managed by DOORS.

When you use the DOORS version of the Requirements Management Interface for a given model, diagram, or M-file, you can only associate it with DOORS requirements (and not Word, Excel, or HTML requirements documents). Similarly, when you use the standard version of the Requirements Management Interface for a given model, diagram, or M-file, you can only associate Word, Excel, and HTML requirements documents with it (and not DOORS requirements).

Features

Use the Requirements Management Interface to:

- Associate requirements with these objects:
  - Simulink subsystems and blocks
  - Stateflow charts, states, and transitions
  - MATLAB M-files
- See which objects have requirements associated with them.
- Go from the Requirements Management Interface Navigator to a requirement.
- Go from a requirement in DOORS to a MATLAB object (for DOORS version only).
- Run a MATLAB script that executes a Simulink simulation or M-file from DOORS (for DOORS version only).

Related Products

The Requirements Management Interface requires:

- MATLAB Version 5.3.1 or later
- Simulink Version 3.0.1 or later
- Stateflow Version 2.0.1 or later, if you want to use the Requirements Management Interface for Stateflow diagrams

For more information about any of these products, see either:

- The online documentation for that product if it is installed or if you are reading the documentation from the CD
- The MathWorks Web site, at `http://www.mathworks.com`; see the “products” section

Using This Guide

Expected Background

To use the Requirements Management Interface:

- You must understand MATLAB, Simulink, or Stateflow.
- If you are using the standard version of the Requirements Management Interface, no special knowledge of Microsoft Word, Microsoft Excel, or HTML documents is required.
- If you are using the DOORS version, you must have experience working with projects and formal modules in DOORS.

Organization of the Document

Section	Description
"Using the Requirements Management Interface"	Instructions for using the standard version of the Requirements Management Interface. Use this to associate Simulink models, Stateflow diagrams, and MATLAB M-files with requirements in HTML, Microsoft Word, and Microsoft Excel documents.
"Using the Requirements Management Interface for QSS's DOORS"	Instructions for using the DOORS version of the Requirements Management Interface. Use this if you use the DOORS requirements management system and want to associate Simulink models, Stateflow diagrams, and MATLAB M-files with requirements in DOORS.

Examples Used

Simulink Example

This guide uses the Simulink `engine` model as the example object for Simulink. It is located in `$matlabroot\toolbox\simulink\simdemos`.

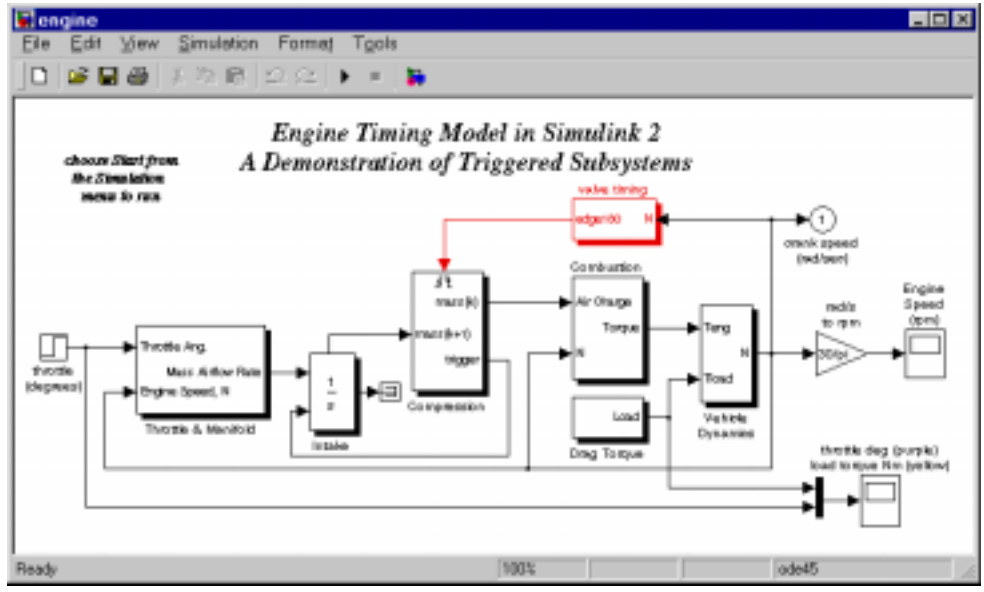


Figure 1: Simulink engine Model

Stateflow Example

This guide uses the `fuelsys` model as the example object for Stateflow. The model contains a Stateflow chart in the **fuel rate controller**. The `fuelsys` model is located in `$matlabroot\toolbox\stateflow\sfdemos`.

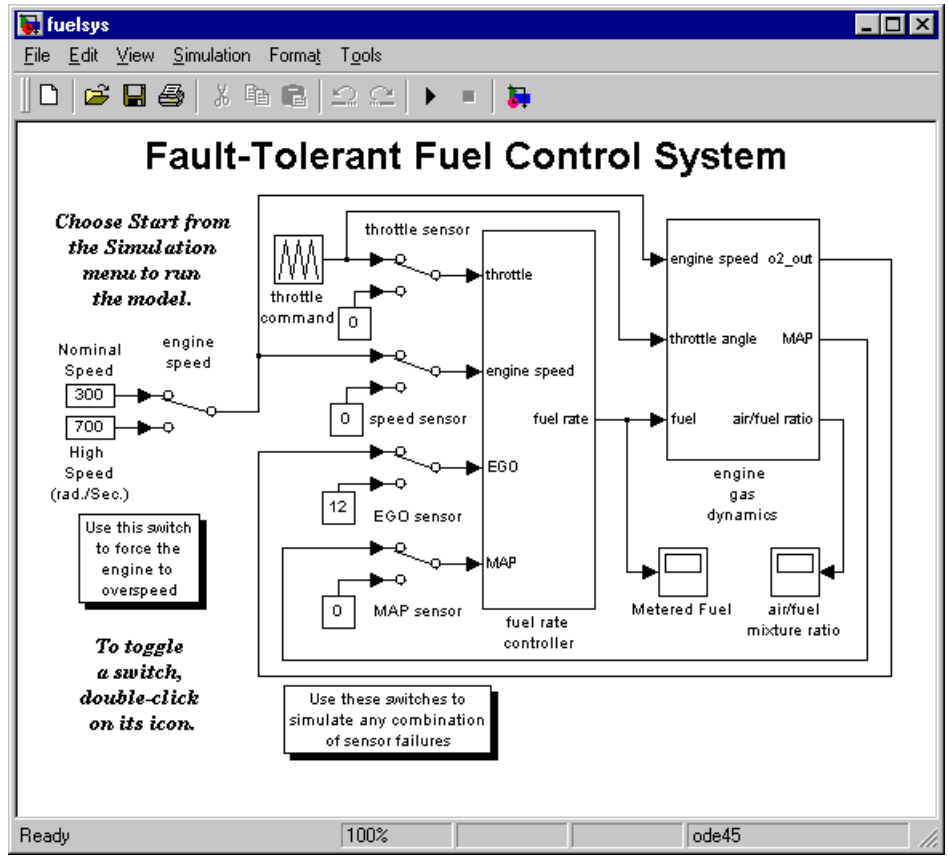


Figure 2: fuelsys Model with Stateflow Fuel Rate Controller Chart

MATLAB M-File Example

This guide uses an M-file `engine.m` as the example object for MATLAB. Any M-file works the same way in the Requirements Management Interface.

Configuration Information

System Requirements

The Requirements Management Interface works on:

- Microsoft Windows 95, Windows 98, and Windows NT systems
- UNIX systems where MATLAB supports Java (for HTML-based requirements documents only).

To use the Requirements Management Interface, you need the following software installed:

- MATLAB Version 5.3.1 or higher
- Simulink Version 3.0.1 or higher
- Stateflow Version 2.0.1 or higher, if you want to use the Requirements Management Interface for Stateflow diagrams
- For the standard version of the Requirements Management Interface, you need the applications that you use to view your requirements documents. The following applications are supported:
  - Microsoft Word 97 or higher
  - Microsoft Excel 97 or higher
  - Netscape Navigator Version 4.0 or higher, or Microsoft Internet Explorer Version 4.0 or higher for HTML documents
- For the DOORS version of the Requirements Management Interface, you need QSS's DOORS requirements management system, Version 4.0.3 or higher.

Installation

You use the standard MATLAB installation process to install the Requirements Management Interface. However, if you plan to use the DOORS version of the Requirements Management Interface, you must install additional files.

Additional Installation for DOORS Users

1 If DOORS is running, close DOORS.

2. Copy `addins.idx` and `addins.hlp` from `$matlabroot\toolbox\reqmgt` to the `<doors>\lib\dxl\addins` directory. `<doors>` represents the top-level directory where DOORS is installed. Replace any existing versions of the files if they have not been modified; otherwise, merge their contents.

3. Copy the following files from `$matlabroot\toolbox\reqmgt` to the `<doors>\lib\dxl\addins\dmi` directory. Replace any existing versions of the files.

   `dmi.hlp`

   `dmi.idx`

   `dmi.inc`

   `runsim.dxl`

   `selblk.dxl`

4. Change the include path in the DOORS startup file. To do so:

   a. Open `<doors>\lib\dxl\startup.dxl` in an editor.

   b. In the user defined files section of `startup.dxl`, add the line

      ```
      #include <addins/dmi/dmi.inc>
      ```

   c. Save `startup.dxl` and close it.

Typographical Conventions

Item	Convention to Use	Example
Function names/syntax	`Monospace` font	The `cos` function finds the cosine of each array element. Syntax line example is `MLGetVar ML_var_name`
Keys	**Boldface** with an initial capital letter	Press the **Return** key.
Menu names, menu items, and controls	**Boldface** with an initial capital letter	Choose the **File** menu.
New terms	*Italics*	An *array* is an ordered collection of information.

1

Using the Requirements Management Interface

Overview of the Process

This diagram depicts how the Requirements Management Interface associates objects with requirements documents.

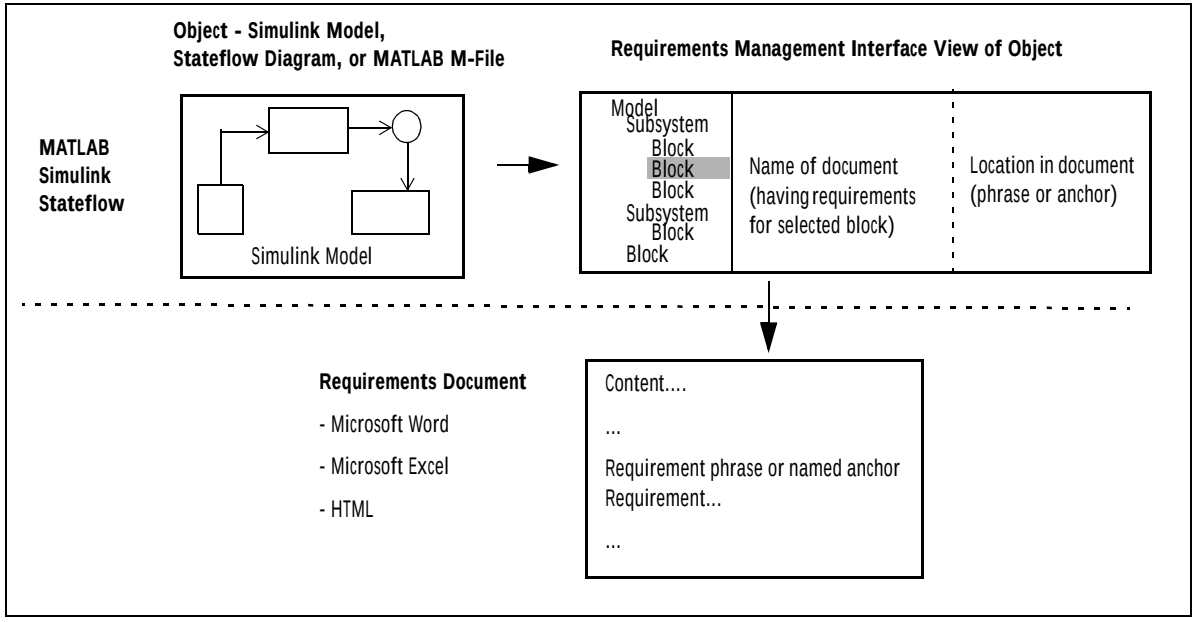


Figure 1-1: How the Requirements Management Interface Works with Requirements Documents

Summary of Steps to Use the Requirements Management Interface

Step	Description
1	Start the Requirements Management Interface – see "Startup" on page 1-4.
2	In the Requirements Management Interface Navigator, open an object (Simulink model, Stateflow diagram, or MATLAB M-file) – see "Opening an Object in the Navigator" on page 1-5.
3	In the Navigator, add requirements for elements in the object – see "Adding Requirements" on page 1-9.
4	For an object that has associated requirements, you can use the Navigator to: • See which nodes have requirements associated with them – see "Indication of Requirements" on page 1-13. • View only nodes that have requirements – see "Changing the View in the Navigator" on page 1-17. • Go to a requirement in its document – see "Going to the Requirements Document" on page 1-19. • Go to an element in an object – see "Going from the Navigator to an Object" on page 1-20. • Indicate in an object the elements that have requirements associated with them – see "Highlighting Subsystems and Blocks That Have Requirements" on page 1-21.

Startup

1. Start MATLAB.
2. Edit the file `$matlabroot\toolbox\reqmgt\reqmgropts.m` to specify the `OTHERS` option for the Requirements Management Interface.
   a. Type `edit reqmgropts` at the MATLAB prompt.

      The `reqmgropts.m` file opens in your editor.

   b. Edit the `reqsys` line so it appears as follows.

      ```
      reqsys = 'OTHERS';
      ```

   c. Save `reqmgropts.m` and close it.
3. At the MATLAB prompt, type

   ```
   rminav
   ```

   or, in Simulink or Stateflow, select **Requirements management interface** from the **Tools** menu.

   The Navigator window opens.

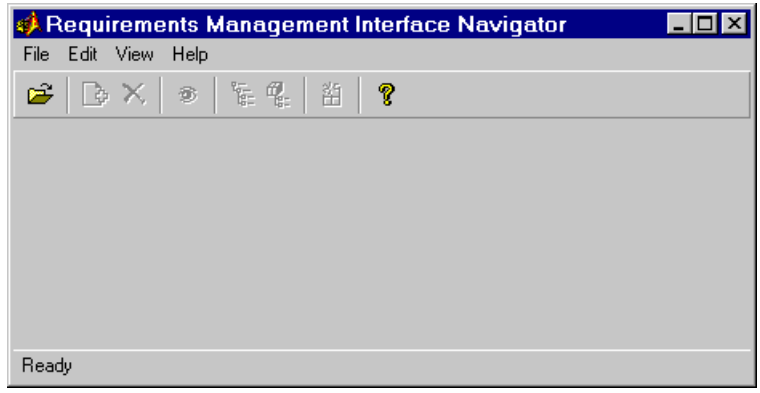


Figure 1-2: Navigator – No Object Open

Opening an Object in the Navigator

You can use the current model or the file for another object. The object's file must be on the MATLAB path or in the current directory. To open an object:

- If the Simulink or Stateflow object you want to associate requirements with is already open and is the currently active model, select **Open Current Model** from the **File** menu in the Navigator.
- If the Simulink or Stateflow object you want to associate requirements with is not currently open, or if you want to associate requirements with a MATLAB M-file, use **Open**:

  a Select **Open** from the **File** menu in the Navigator.

  The **Open Model/File** dialog box appears.

  b Select the object that you want to associate with requirements and click **Open**.

  The object opens in Simulink or Stateflow.

Note While using the Navigator, do not close the model in Simulink.

For a Simulink or Stateflow object, the subsystems, blocks, charts, states, and transitions appear as nodes in a hierarchical tree structure in the left pane of the Navigator. An M-file appears as a single node in the left pane of the Navigator.

This example shows the Simulink `toolbox\simulink\simdemos\engine` model in the Navigator.

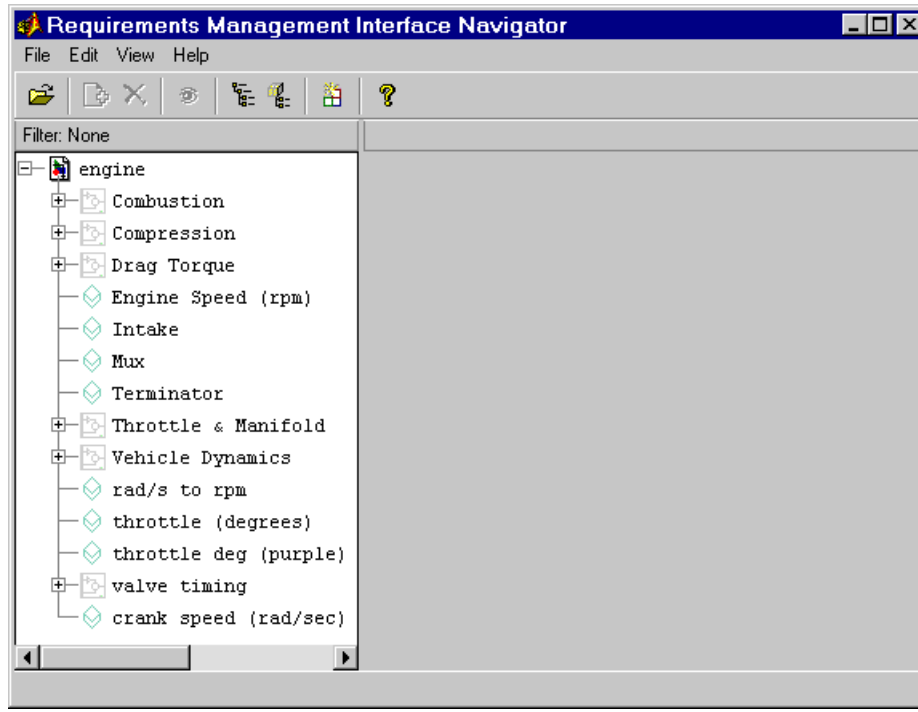


Figure 1-3: Navigator with engine Model Open

Stateflow Charts and M-Files in the Navigator

Stateflow Chart in the Navigator

This example shows a partial view of the `fuelsys` model, which contains the **control logic** Stateflow chart in the **fuel rate controller** subsystem.

You must have Stateflow installed in order to expand the Stateflow chart node in the Navigator.

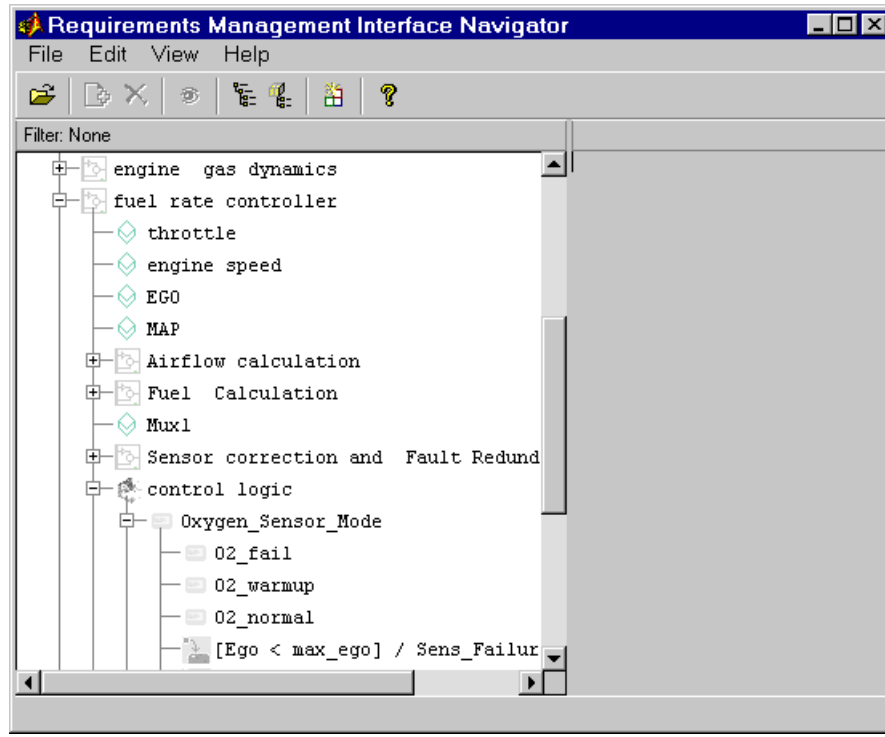


Figure 1-4: Navigator with fuelsys Model Open, Showing Stateflow Chart

MATLAB M-File in Navigator

This example shows a MATLAB M-file, `engine.m`, in the Navigator.

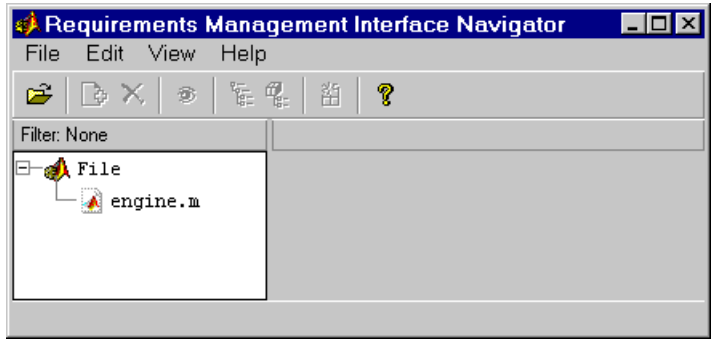


Figure 1-5: Navigator with M-File engine.m Open

Adding Requirements

After opening an object in the Navigator, you can associate a node in the hierarchical tree with requirements in documents. Then you can go from the Navigator to a requirement. You can associate multiple requirements documents with a single node in the tree.

Note You must have write permission for an object in order to add requirements. Adding a requirement automatically saves the model or M-file.

About the Requirements Documents

The requirements can be in Microsoft Word, Microsoft Excel, or HTML documents.

- You can specify just the requirements document name. Then when you go to it from the Navigator, you will go to the top of that document.
- You can specify a specific location within the requirements document.
  - For Word and Excel, the specific location is simply the first instance of a given word in the requirements document.
  - For HTML, the specific location is a named anchor, for example, `<A href="valve_timing.html">`.

How To Add a Requirement

1 Select the node you want to associate with a requirement. You can add a requirement to any node in the hierarchical tree.

  If a node contains other nodes, click ⊞ next to the node to expand it. You can then select one of the nodes in the expanded hierarchical tree.

2 From the **Edit** menu, select **Add**, or click the button on the toolbar.

The cursor moves to the right pane, allowing you to type in the requirement information. This example shows adding a requirement for the **N** node of the **valve timing** subsystem in the engine model.

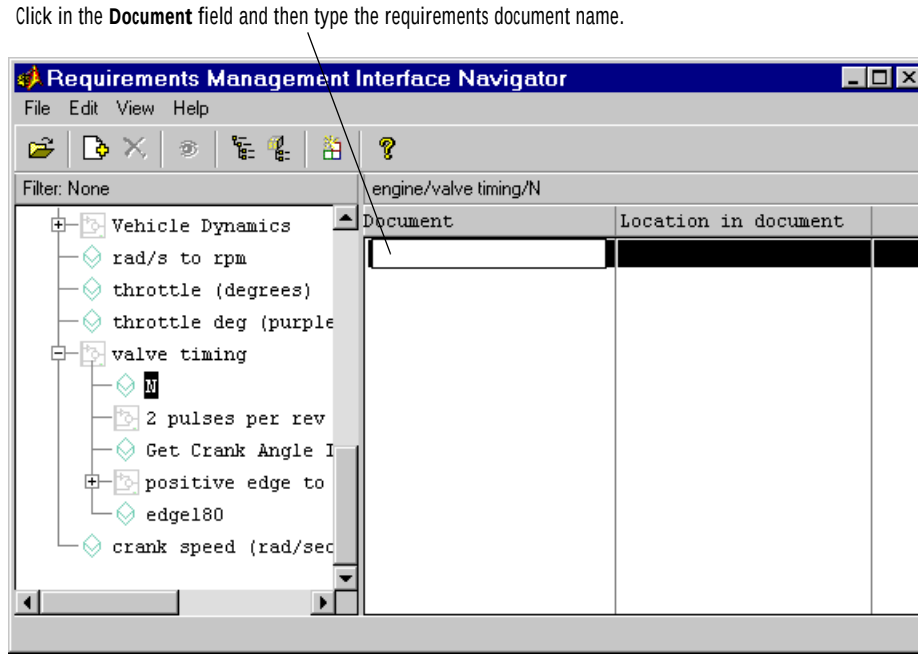


Figure 1-6: Adding a Requirement

3 Click in the **Document** field in the right pane, type the full path for the requirements document, and press **Enter**.

Here are examples of full pathnames.

Pathname	For Document Type
`c:\Requirements\Car Requirements.doc`	Word
`d:\car_files\car_reqts.xls`	Excel
`http://www-internal.com/carreqts.html`	HTML
`file:///bserver/car/requirements.html`	HTML

4 If the requirement is at a specific location within the document, you can add the location. Otherwise, skip this step.

Double-click in the **Location in document** field, type the location, and then press **Enter**. An example location for any document type is `valve_timing`.

5 Repeat this process to add additional requirements for the same node or for other nodes.

Note If your Stateflow chart contains transitions that are not unique, do not associate requirements with them.

This example shows three requirements for the **N** node of the **valve timing** subsystem in the engine model.

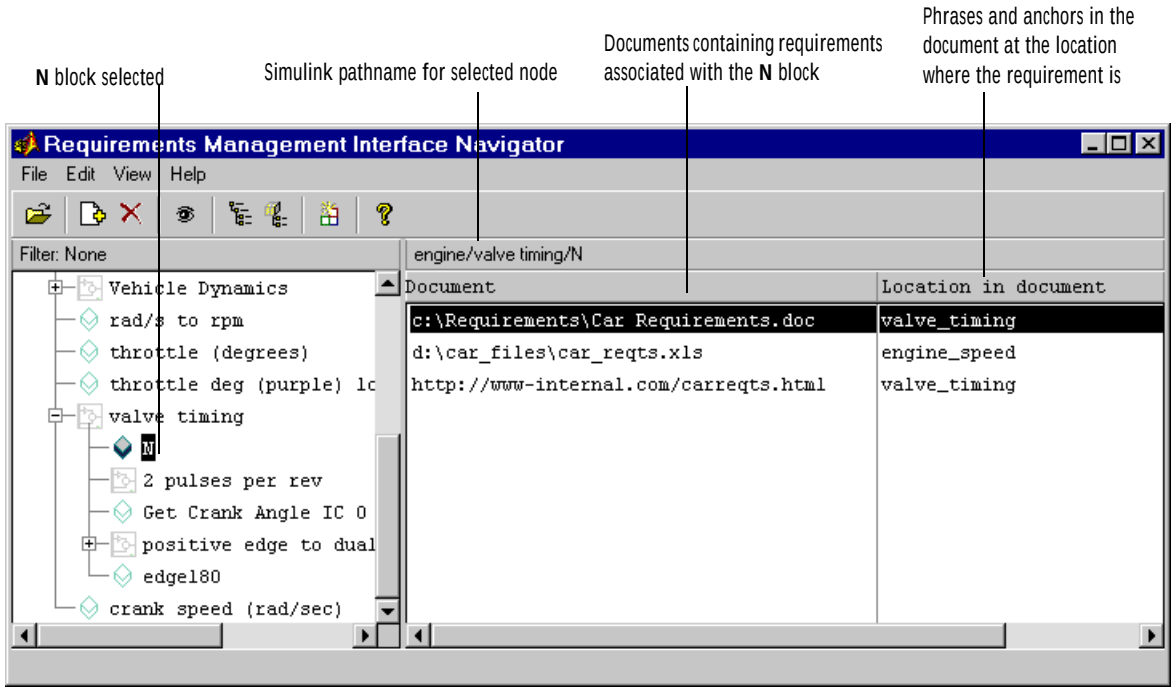


Figure 1-7: Node with Associated Requirements

Saved Requirements

Adding a requirement automatically saves the model or M-file.

- For a Simulink or Stateflow node, the Requirements Management Interface stores information about the requirements document with the object's data structure.
- For an M-file node, the Requirements Management Interface writes a commented line to the M-file. Do not modify this line. For example,

```
% Autogenerated requirements (do not modify) = { 'OTHERS'
'c:\Requirements\Car Requirements.doc' '' 'true' }
```

Viewing Requirements in the Navigator

Use the Navigator to see which objects have requirements associated with them.

Indication of Requirements

After adding a requirement to a node, the shading of the node's icon changes. In the Navigator, you can see which nodes have requirements associated with them by the shading of the node's icons.

The following example shows that nodes for the **N** block in the **valve timing** subsystem and the **positive edge to dual edge conversion** subsystem are shaded, indicating they have requirements associated with them. The nodes under the **positive edge to dual edge conversion** subsystem are lightly shaded, indicating they belong to a subsystem that has requirements associated with it.

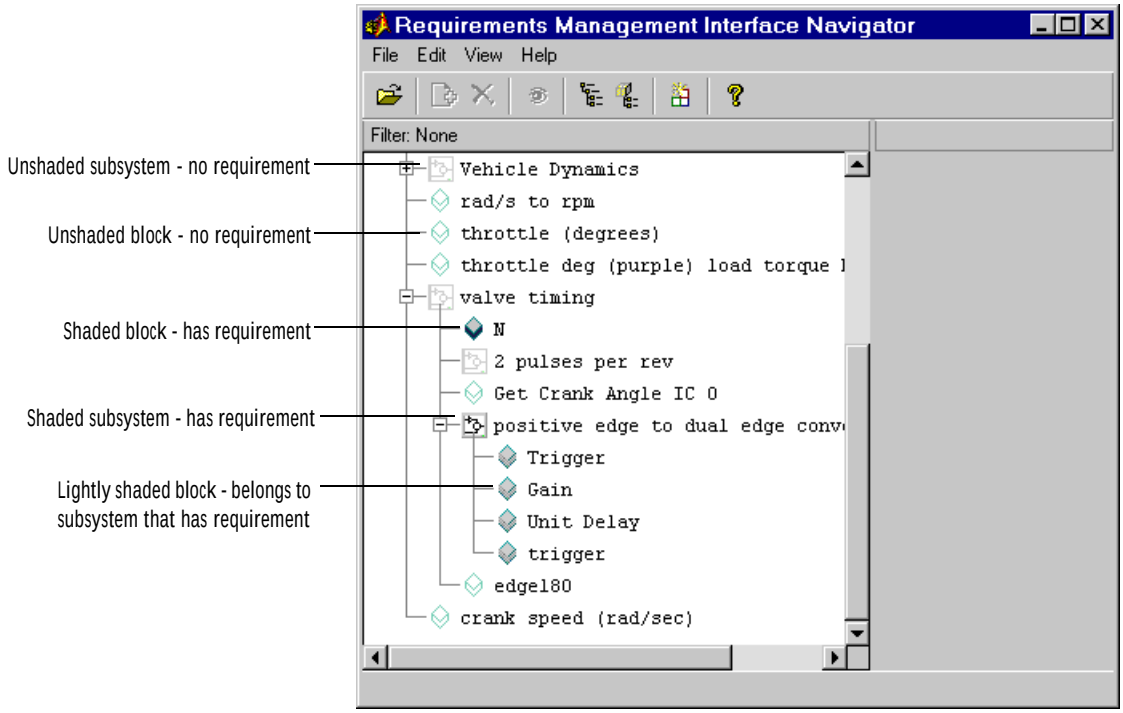


Figure 1-8: Navigator Showing Simulink Nodes That Have Requirements

The shading of a node's icon indicates whether or not it has associated requirements, as shown in the following table.

Table 1-1: Node Icons in Navigator and What They Represent

Node	Shading	What It Represents
	Not shaded	Simulink subsystem with no requirements
	Shaded	Simulink subsystem that has requirements
	Lightly shaded	Simulink subsystem belonging to a subsystem that has requirements; subsystem itself has no requirements
	Not shaded	Simulink block with no requirements
	Shaded	Simulink block that has requirements
	Lightly shaded	Simulink block belonging to a subsystem that has requirements; block itself has no requirements
	Not shaded	Stateflow chart with no requirements
	Shaded	Stateflow chart that has requirements
	Lightly shaded	Stateflow chart belonging to a subsystem that has requirements; chart itself has no requirements
	Not shaded	Stateflow state with no requirements
	Shaded (with yellow)	Stateflow state that has requirements
	Lightly shaded	Stateflow state belonging to a superstate, chart, or subsystem that has requirements; state itself has no requirements
	Not shaded	Stateflow transition with no requirements
	Shaded (with blue)	Stateflow transition that has requirements
	Lightly shaded	Stateflow transition belonging to a superstate, chart, or subsystem that has requirements; transition itself has no requirements
	Not shaded	MATLAB M-file with no requirements
	Shaded	MATLAB M-file that has requirements

Stateflow Chart with Requirements

This example shows the **control logic** Stateflow chart (in the **fuel rate controller** subsystem of the `fuelsys` model) with requirements.

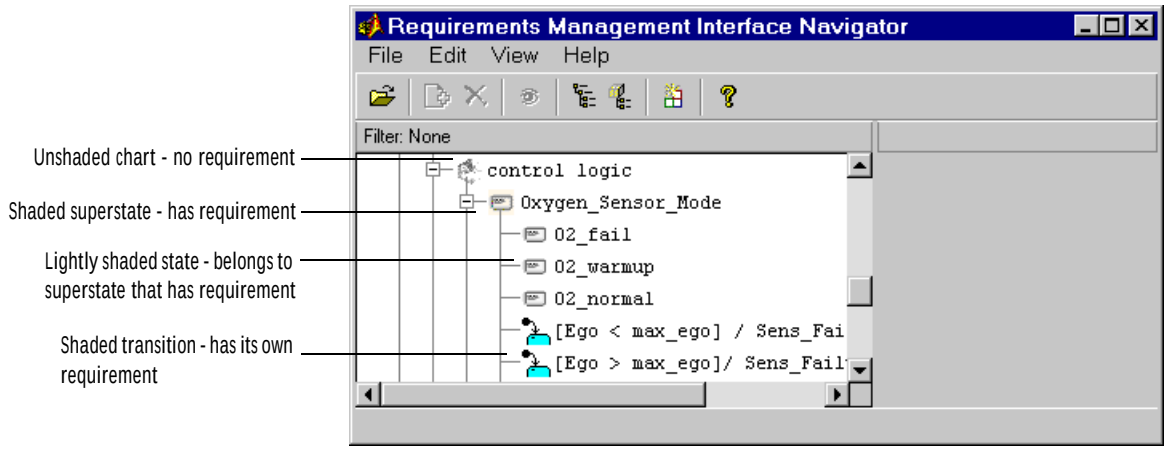


Figure 1-9: Navigator for Stateflow Chart with Requirements

MATLAB M-File with Requirements

This example shows the M-file `engine.m` with requirements.

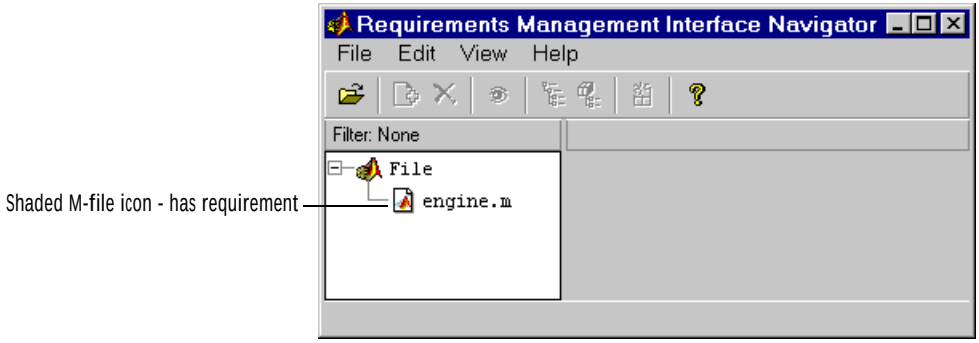


Figure 1-10: Navigator for M-File with Requirements

Viewing the Requirements for a Node

Select a node that has requirements associated with it. The requirements for that node appear in the right pane of the Navigator.

Changing or Deleting a Requirement

After adding a requirement in the Navigator, you can change the requirements document or location, or you can delete the requirement.

Changing a Requirement

1. In the left pane, select the node whose requirement you want to change.
2. In the right pane, select the requirement you want to change.
3. From the **Edit** menu, select **Replace** to edit the **Document** field. Note that double-clicking in this field opens the requirement document. To edit the **Location in document** field, double-click in the field.
4. Type the changes in the **Document** and/or **Location in document** fields and then press **Enter**.

Deleting a Requirement

1. In the left pane, select the node whose requirement you want to delete.
2. In the right pane, select the requirement you want to delete.
3. From the **Edit** menu, select **Delete**. Or click the  button.

   The requirement is removed and no longer appears.

Changing the View in the Navigator

In the Navigator, you can change the view to:

- "Make Panes Wider or Narrower"
- "Show or Hide Lower Levels in the Tree"
- "Show Only the Highest Level in the Tree" hiding the lower levels
- "Show Only Nodes That Have Requirements" associated with them

Make Panes Wider or Narrower

Drag the separator bar between the left and right panes in the Navigator to make one of the panes wider and the other pane narrower. You can also adjust the width of the **Document** and **Location in document** columns using the separator between the column titles in the right pane.

You can make the overall window larger or smaller by dragging any edge of the window.

Show or Hide Lower Levels in the Tree

- Click ⊞ to expand a node in the tree, which shows lower levels.
- Click ⊟ to collapse a node in the tree, which hides the lower levels.

Show Only the Highest Level in the Tree

From the **View** menu, select **Refresh**. This collapses the tree so that only the highest level nodes are shown.

Show Only Nodes That Have Requirements

From the **View** menu, select **Show Requirement Nodes** or click the  button.

Only nodes that have requirements associated with them are shown. Other nodes are hidden. The **Filter** display indicates **Requirements Only**.

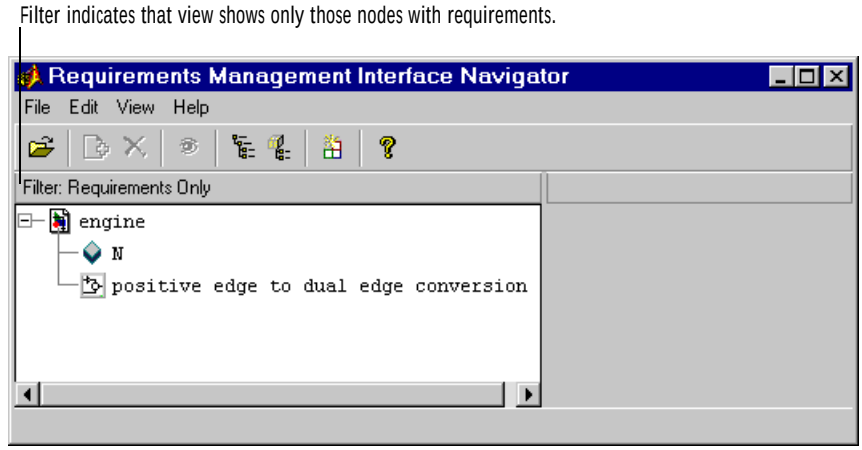


Figure 1-11: Navigator with Requirements Only Filter

When the **Filter** is **Requirements Only**, to see all nodes, select **Show All Nodes** from the **View** menu, or click the button. The **Filter** display then indicates **None**, meaning that there is no filter applied to the requirements listing and all requirements are included.

Going to the Requirements Document

1. In the left pane of the Navigator, select the node that has an associated requirement.

2. In the right pane of the Navigator, select the requirement you want to go to.

3. From the **View** menu, select **Navigate to Requirement**, or click the navigate button .

   The requirements document (specified in the **Document** column) opens. If you specified the **Location in document**, the document opens scrolled to that location.

Double-clicking in the **Document** field is another way to go from the Navigator to the requirements document.

Going from the Navigator to an Object

From the Navigator, you can go to the selected node for an object:

1 In the Navigator, select a node.

2 From the **View** menu, select **View Block**.

The object opens.

- For a Simulink node, the subsystem containing the selected node opens with that block or subsystem selected in the model.
- For a Stateflow node, the diagram containing the selected node opens.
- For an M-file node, the M-file opens in your editor.

Double-clicking on a node is another way to go from the Navigator to the object.

This example shows the results when you select the **N** block in the **valve timing** subsystem in the Navigator and then select **View Block**.

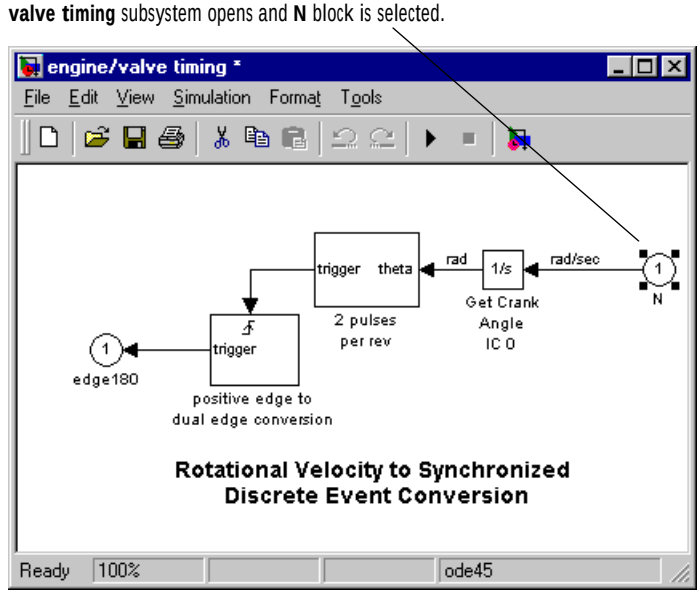


Figure 1-12: Simulink Block Accessed from Navigator

Highlighting Subsystems and Blocks That Have Requirements

You can highlight subsystems and blocks that have requirements in a Simulink model:

1 In the Navigator, select **Highlight Blocks that have Requirements** from the **View** menu, or click the button. The menu item becomes checked.

In the Simulink model, the subsystems and blocks that have requirements associated with them appear highlighted in blue.

This example shows the nodes in the **valve timing** subsystem that have associated requirements.

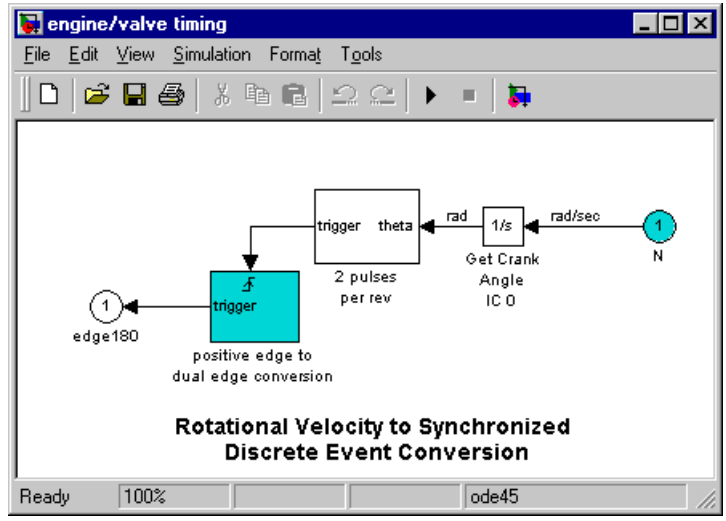


Figure 1-13: Simulink Blocks Highlighted to Indicate Associated Requirements

2 The highlight feature operates as a toggle switch. To turn off the highlighting, select **Highlight Blocks that have Requirements** from the **View** menu; the menu item is no longer checked. Or click the toolbar button.

The blocks and subsystems in the model are no longer highlighted.

Note When you save a Simulink model, it is saved with the highlighting. You *cannot* later turn off the highlighting without opening the model again in the Navigator. Therefore, unless you want the highlighting to become a "permanent" part of the model file, turn off highlighting using the Navigator *before* you save a model file in Simulink.

2

Using the Requirements Management Interface for QSS's DOORS

Overview of the Process

This diagram depicts how the Requirements Management Interface associates objects with DOORS requirements.

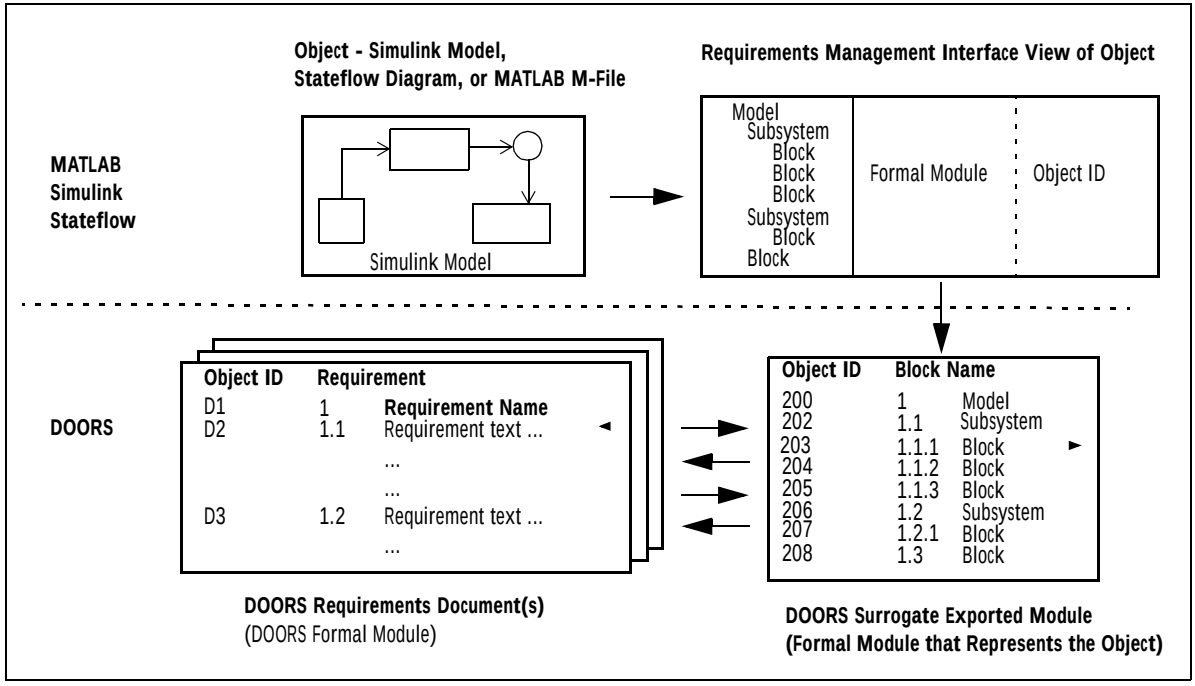


Figure 2-1: How the Requirements Management Interface and DOORS Work Together

---

Note The Requirements Management Interface and DOORS both use the term *object*, but each uses the term differently. In the Requirements Management Interface, and in this document, object refers to a Simulink model, Stateflow diagram, or MATLAB M-file. In DOORS, object refers to

each numbered element in the exported formal module, sometimes called a surrogate module, for a Simulink model, Stateflow diagram, or MATLAB M-file. DOORS assigns each of these objects a unique object identifier. In this document, these objects will be referred to as DOORS objects.

Summary of Steps to Use the Requirements Management Interface

Step	Description
1	Start the Requirements Management Interface – see “Startup” on page 2-5.
2	In the Requirements Management Interface Navigator, open an object (Simulink model, Stateflow diagram, or MATLAB M-file) – see “Opening an Object in the Navigator” on page 2-7.
3	In the Navigator, synchronize the object with DOORS. Synchronizing causes DOORS to create an exported formal module for that project. This surrogate exported module provides the mapping from DOORS to the Navigator. See “Synchronizing the Navigator with DOORS” on page 2-11.
4	Create links between the surrogate exported module and other formal modules in DOORS. See “Linking an Object to a DOORS Requirement” on page 2-16.

Step	Description (Continued)
5	For an object that has associated DOORS requirements, you can use the Navigator to: • See which nodes have requirements associated with them – see "Indication of Requirements" on page 2-18. • View only nodes that have requirements – see "Changing the View in the Navigator" on page 2-23. • Go to DOORS – see "Going from the Navigator to DOORS" on page 2-24. • Go to an element in a object – see "Going from the Navigator to a MATLAB Object" on page 2-25. • Indicate in an object which elements have requirements associated with them – see "Highlighting Subsystems and Blocks That Have Requirements" on page 2-26.
6	In DOORS, you can go from DOORS to an object – see "Going from DOORS to an Object" on page 2-28.
7	From DOORS, you can run a script in MATLAB – see "Running a Script from DOORS" on page 2-29.

Startup

1. Start DOORS and open the project containing the requirements you want to associate with the object.

   You must have a project open in DOORS (be logged in) in order to use MATLAB's Requirements Management Interface.

2. Start MATLAB with the `/automation` startup option.

   For example, make a shortcut to start MATLAB:

   ```
   \...\matlab.exe /automation
   ```

   Note that MATLAB starts up minimized and the directory that MATLAB starts in is `<matlabroot>\bin`.

3. Edit the file `$matlabroot\toolbox\reqmgt\reqmgropts.m` to specify the `DOORS` option for the Requirements Management Interface.

   a. Type `edit reqmgropts` at the MATLAB prompt.

      The `reqmgropts.m` file opens in your editor.

   b. Edit the `reqsys` line so it appears as follows.

      ```
      reqsys = 'DOORS';
      ```

   c. Save `reqmgropts.m` and close it.

4 At the MATLAB prompt, type

```
rminav
```

or, in Simulink or Stateflow, select **Requirements management interface** from the **Tools** menu.

The Navigator window opens.

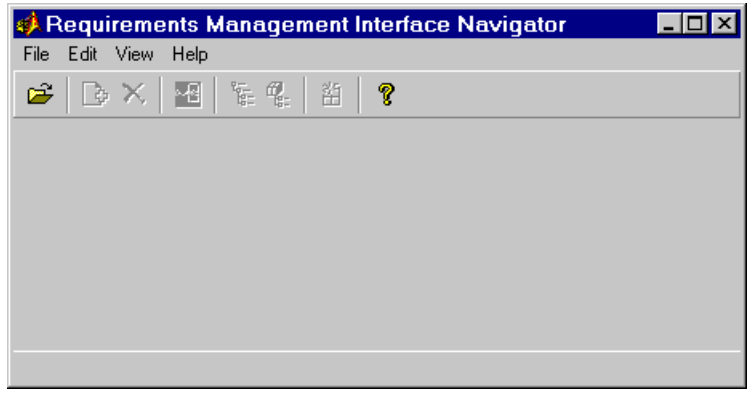


Figure 2-2: Navigator – No Object Open

Opening an Object in the Navigator

You can use the current model or the file for another object. The object's file must be on the MATLAB path or in the current directory. To open an object:

- If the Simulink or Stateflow object you want to associate requirements with is already open and is the currently active model, select **Open Current Model** from the **File** menu in the Navigator.
- If the Simulink or Stateflow object you want to associate requirements with is not currently open, or if you want to associate requirements with a MATLAB M-file, use **Open File**:

  a Select **Open File** From the **File** menu in the Navigator.

  The **Open Model/File** dialog box appears.

  b Select the object that you want to associate with requirements and click **Open**.

  The object opens in Simulink or Stateflow.

Note While using the Navigator, do not close the model in Simulink. In addition, if you close DOORS while using the Navigator, MATLAB closes. If the object has changed, MATLAB prompts you to save your changes.

For a Simulink or Stateflow object, the subsystems, blocks, charts, states, and transitions appear as nodes in a hierarchical tree structure in the left pane of the Navigator. An M-file appears as a single node in the left pane of the Navigator.

This example shows the Simulink `toolbox\simulink\simdemos\engine` model in the Navigator.

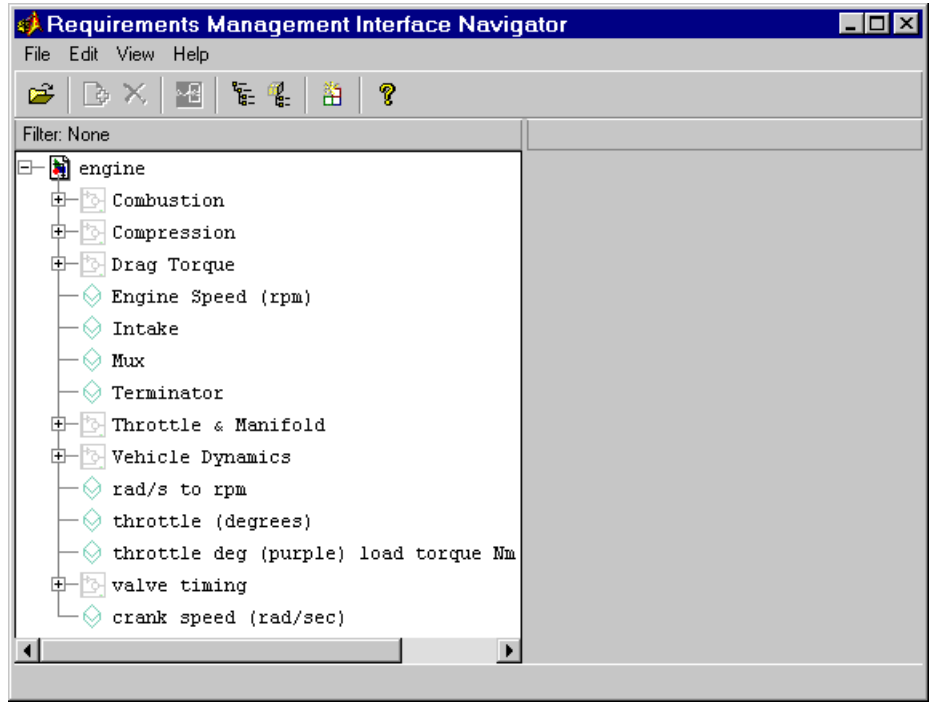


Figure 2-3: Navigator with engine Model Open

Stateflow Charts and MATLAB M-Files in the Navigator

Stateflow Chart in Navigator

This example shows a partial view of the `fuelsys` model, which contains the **control logic** Stateflow chart in the **fuel rate controller** subsystem.

You must have Stateflow installed in order to expand the Stateflow chart node in the Navigator.

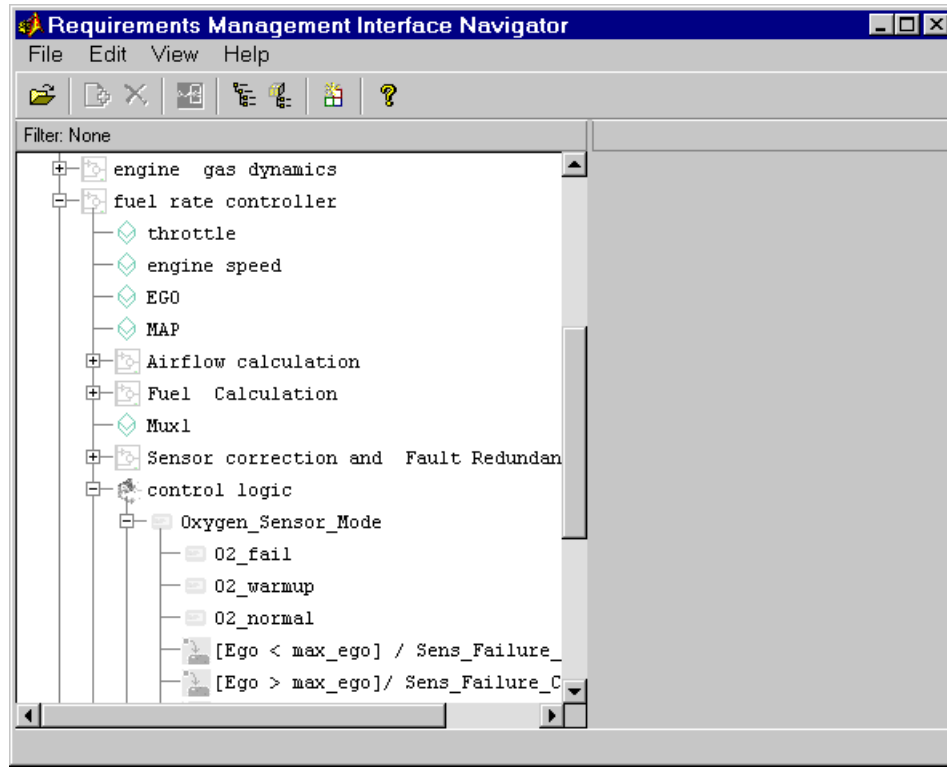


Figure 2-4: Navigator with fuelsys Model Open, Showing Stateflow Chart

MATLAB M-File in Navigator

This example shows a MATLAB M-file, `engine.m`, in the Navigator.

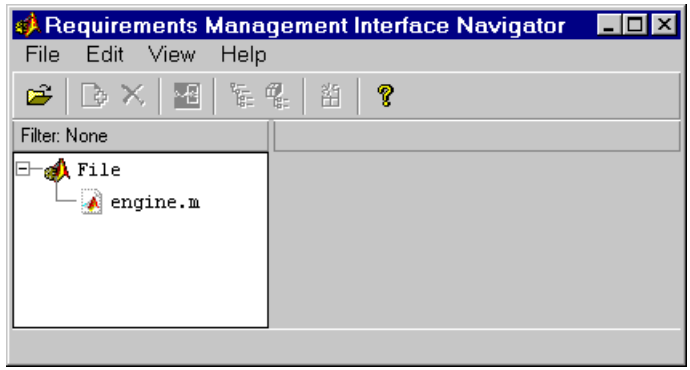


Figure 2-5: Navigator with M-File engine.m Open

Synchronizing the Navigator with DOORS

What Synchronizing Does and When to Synchronize

You synchronize the object that is open in the Navigator to associate it with a project in DOORS.

- Synchronize the first time you open a given object in the Navigator. This causes DOORS to create a formal module called a surrogate exported module that represents the hierarchical structure of elements in the object. You can then use DOORS to associate requirements with the elements in that surrogate exported module.
- When you open an object in the Navigator that already has a DOORS surrogate exported module associated with it, the Navigator checks whether the object and module are synchronized. If the date of the model or M-file is more recent than the synchronization date for the DOORS module, you must synchronize again.
- Synchronize again whenever you:
  - Change links in the DOORS project
  - Make changes to the object in Simulink, Stateflow, or MATLAB

  Synchronizing causes changes to be reflected in the Navigator for that object, and in the DOORS surrogate exported module for that object.
- Synchronizing automatically saves the object.
  - For a Simulink or Stateflow node, the Requirements Management Interface stores information about the requirements with the object.
  - For an M-file node, the Requirements Management Interface writes a commented line to the M-file. Do not modify this line. For example,

    ```
    % Autogenerated requirements (do not modify) = { 'DOORS'
    'engine' '1' 'true' }
    ```

Note You must have write permission for the object in order to synchronize. If you do not have write permission for the object, you can still view the DOORS requirements associated with the object.

How to Synchronize

1 From the **File** menu in the Navigator, select **Synchronize**. Note that an hourglass does not appear during synchronization; however, you must wait for a confirmation dialog box to appear.

2 Once the confirmation dialog box appears stating the number of items synchronized, click **OK** to continue.

Synchronizing opens the DOORS surrogate exported module for that object, creating it if one does not already exist. The name of the surrogate exported module is the same as the object name. This example shows the surrogate exported module created for the Simulink `engine` model.

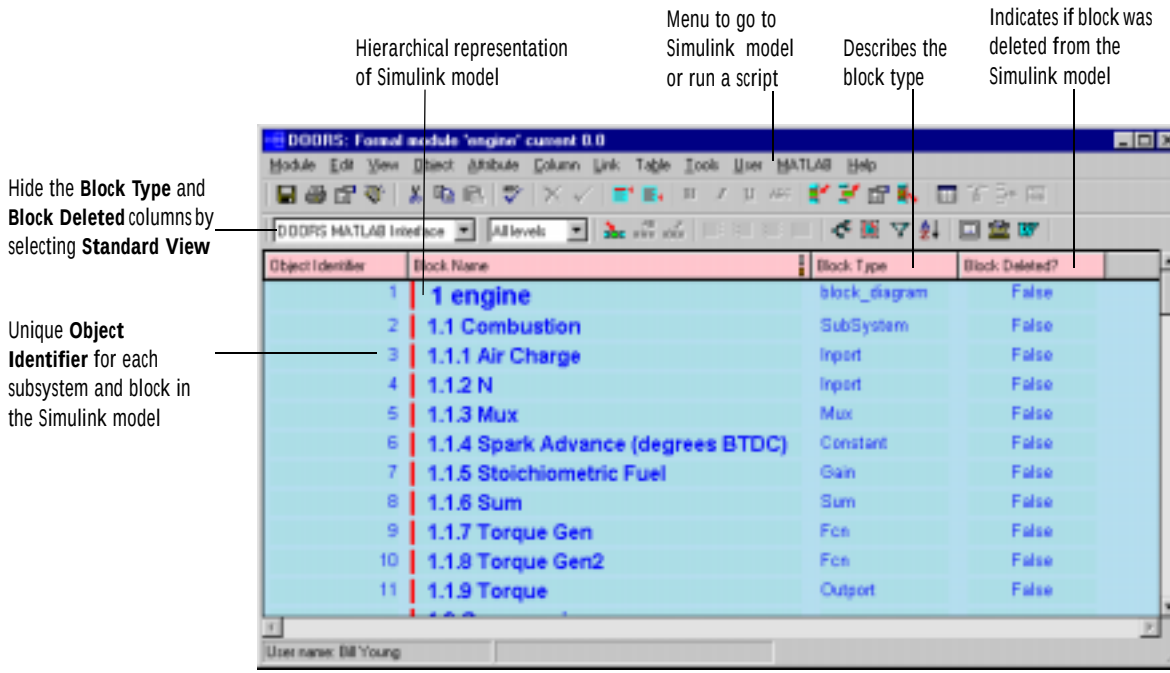


Figure 2-6: DOORS Surrogate Exported Module for Simulink engine Model

Note that:

- Each subsystem and block in the Simulink `engine` model has a unique **Object Identifier**. For example, the identifier for **Air Charge** is **3**.
- Each DOORS object has a hierarchical number associated with it, (shown in the **Block Name** column), which represents its relationship to other objects in the `engine` model. For example, **1.1.1 Air Charge**, is a block in the **1.1 Combustion** subsystem.
- For each DOORS object, there is a **Block Type** description.
- If a block has been deleted from a Simulink model or a state has been deleted from a Stateflow diagram, `True` appears in the **Block Deleted** column. Deleted items do not appear in the Navigator.
- You can hide the **Block Type** and **Block Deleted** columns by changing the view in the toolbar from **DOORS MATLAB Interface** to **Standard View**.

3 You can add a column to the DOORS surrogate exported module that displays the path for each object.

a Select **New** from the **Column** menu, or click the  button on the DOORS toolbar.

The **New Column** dialog box appears.

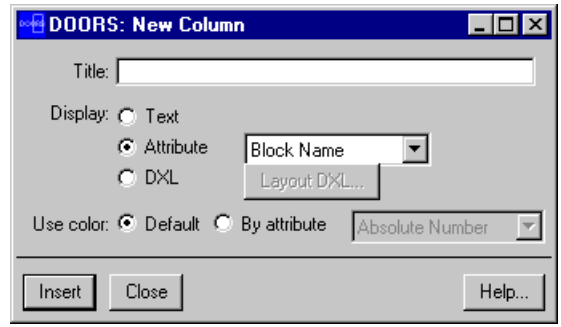


Figure 2-7: New Column Dialog Box in DOORS

b Select **Block Name** for the **Display Attribute**.

c Click **Insert**.

The **Block Name** column is added to the DOORS surrogate exported module and the complete path within the model for each object is shown in that column.

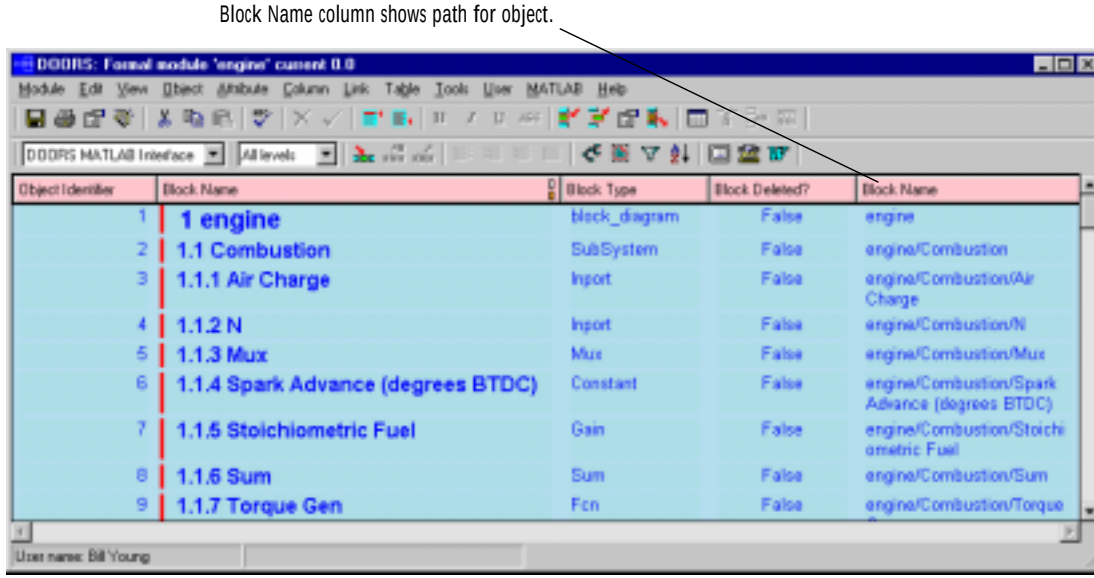


Figure 2-8: DOORS Surrogate Exported Module Showing Pathname for Objects

4 Before you close the DOORS project, save the surrogate exported module in DOORS.

Checking Synchronization

You can check at any time whether a model or M-file is synchronized with DOORS.

1 From the **File** menu in the Navigator, select **Check Synchronization**. Note that an hourglass does not appear during this check; however, you must wait for a confirmation dialog box to appear.

2 Once a confirmation dialog box appears stating whether or not the model or M-file is synchronized with DOORS, click **OK** to continue.

3 If the model or M-file is not synchronized with DOORS and you have write access to the object, from the **File** menu select **Synchronize**.

You can view an object and its associated DOORS requirements from the Navigator even if you don't have write access to the object. First, use **Check Synchronization** to make sure you're viewing a synchronized object.

Linking an Object to a DOORS Requirement

You can link a DOORS object in a surrogate exported module to another DOORS object in a formal module in order to indicate requirements for the object. Do this the same way you make links in any DOORS formal module:

1. To add a link from an object in the surrogate exported module to an object in another formal module, select the object.

   That object becomes highlighted.

2. Right-click on the selected object, and from the pop-up menu select **Link>Start link**. Or from the **Link** menu, select **Start link**.

   An outgoing link icon appears with that object in the surrogate exported module.

3. To complete the link, select the object in the formal module you want to link to. This can be another object in the current surrogate exported module, an object in another surrogate exported module, or an object in any other formal module such as a requirements document.

   That object becomes highlighted.

4. Right-click on the selected object, and from the pop-up menu select **Link>Make link from start**. Or from the **Link** menu, select **Make link from start**.

   An incoming link icon appears with that object in the exported module.

You can also add links that go in the other direction. A link can start from an object in a formal module and link to an object in the surrogate exported module.

Note If your Stateflow chart contains transitions that are not unique, do not associate requirements with them.

The example for the `engine` model shows two outgoing links: one for the **N** block in the **valve timing** subsystem, and one for the **positive edge to dual edge conversion** subsystem.

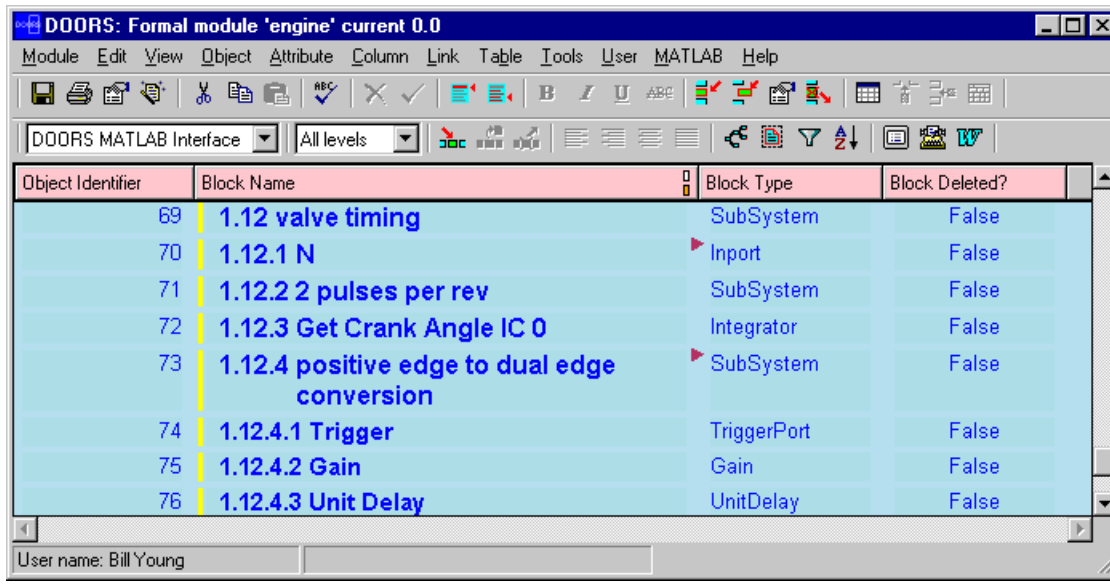


Figure 2-9: DOORS Surrogate Exported Module with Requirements for engine

Viewing Requirements in the Navigator

Use the Navigator to see which objects have requirements associated with them.

Indication of Requirements

In the Navigator, you can see which nodes have requirements associated with them by the shading of the node's icons.

1 Select **Synchronize** from the **File** menu in the Navigator so that the links you just made in DOORS are reflected in the Navigator.

This example shows that nodes for the **N** block in the **valve timing** subsystem and the **positive edge to dual edge conversion** subsystem are shaded, indicating they have DOORS requirements associated with them.

The nodes in the **positive edge to dual edge conversion** subsystem are lightly shaded, indicating they belong to a subsystem that has requirements associated with it.

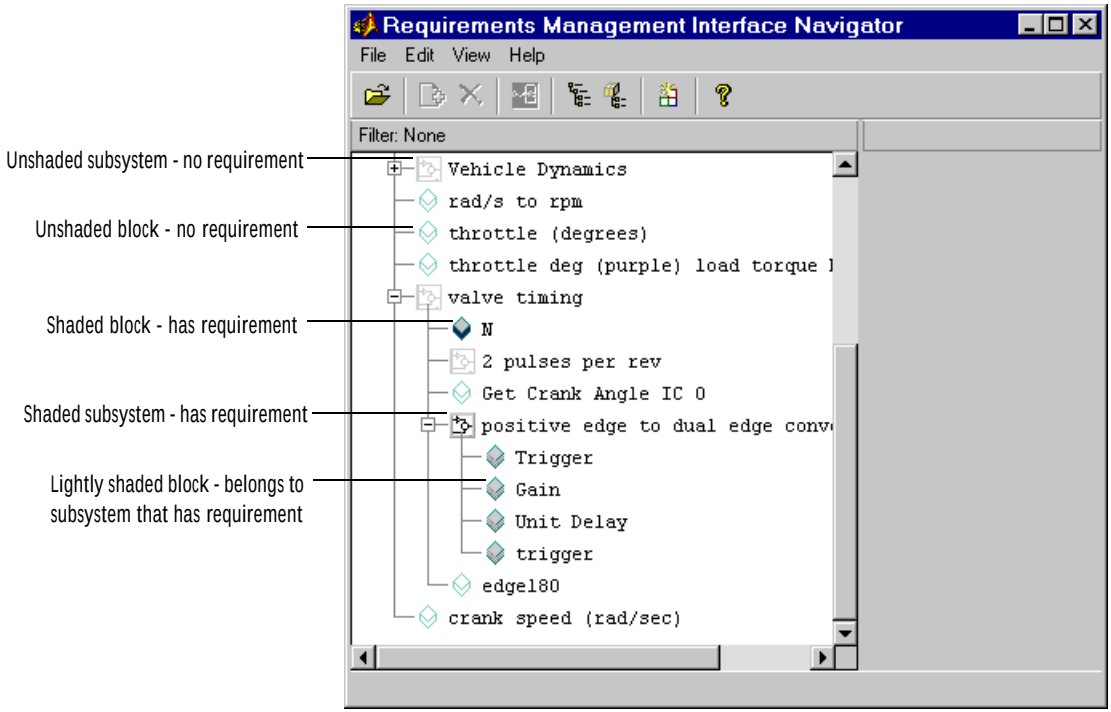


Figure 2-10: Navigator Showing Simulink Nodes That Have DOORS Requirements

The shading of a node's icon indicates whether or not it has associated requirements, as shown in the following table.

Table 2-1: Node Icons in Navigator and What They Represent

Node	Shading	What It Represents
	Not shaded	Simulink subsystem with no requirements
	Shaded	Simulink subsystem that has requirements
	Lightly shaded	Simulink subsystem belonging to a subsystem that has requirements; subsystem itself has no requirements
	Not shaded	Simulink block with no requirements
	Shaded	Simulink block that has requirements
	Lightly shaded	Simulink block belonging to a subsystem that has requirements; block itself has no requirements
	Not shaded	Stateflow chart with no requirements
	Shaded	Stateflow chart that has requirements
	Lightly shaded	Stateflow chart belonging to a subsystem that has requirements; chart itself has no requirements
	Not shaded	Stateflow state with no requirements
	Shaded (with yellow)	Stateflow state that has requirements
	Lightly shaded	Stateflow state belonging to a superstate, chart, or subsystem that has requirements; state itself does has no requirements
	Not shaded	Stateflow transition with no requirements
	Shaded (with blue)	Stateflow transition that has requirements
	Lightly shaded	Stateflow transition belonging to a superstate, chart, or subsystem that has requirements; transition itself has no requirements
	Not shaded	MATLAB M-file with no requirements
	Shaded	MATLAB M-file that has requirements

2 Select a node that has requirements associated with it. If the node contains other nodes, first click ⊞ next to the node to expand it, and then select the specific node you want.

The surrogate exported module name (**Document**) and object identifier (**ID**) appear in the right pane of the Navigator.

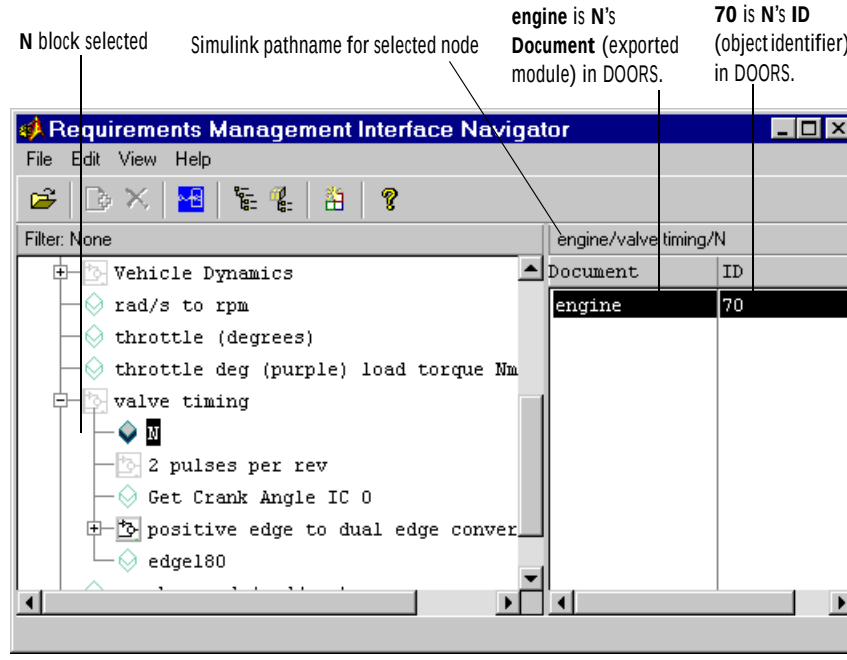


Figure 2-11: Selected Node with Requirement

Stateflow Charts and MATLAB M-Files with Requirements

Stateflow Chart with Requirements. This example shows the **control logic** Stateflow chart (in the **fuel rate controller** subsystem of the `fuelsys` model) with requirements.

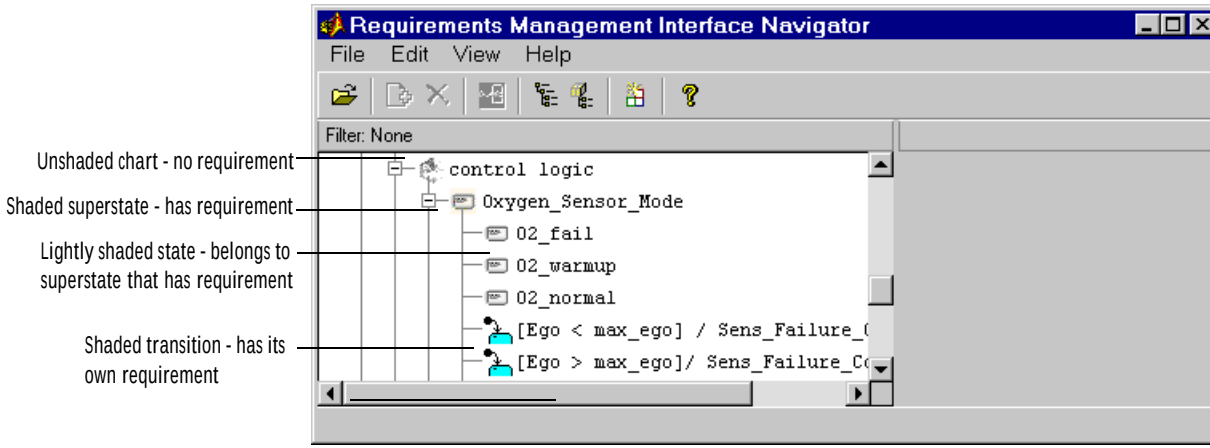


Figure 2-12: Navigator for Stateflow Chart with Requirements

MATLAB M-File with Requirements. This example that shows the M-file `engine.m` with requirements.

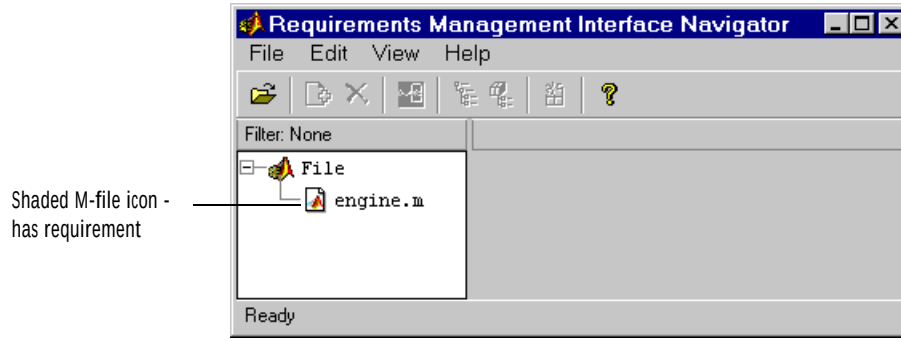


Figure 2-13: Navigator for MATLAB M-File with Requirements

Changing the View in the Navigator

In the Navigator, you can change the view to:

- "Make Panes Wider or Narrower"
- "Show or Hide Lower Levels in the Tree"
- "Show Only the Highest Level in the Tree" hiding the lower levels
- "Show Only Nodes That Have Requirements" associated with them

Make Panes Wider or Narrower

Drag the separator bar between the left and right panes in the Navigator to make one of the panes wider and the other pane narrower.

You can make the overall window larger or smaller by dragging any edge of the window.

Show or Hide Lower Levels in the Tree

- Click ⊞ to expand a node in the tree, which shows lower levels.
- Click ⊟ to collapse a node in the tree, which hides the lower levels.

Show Only the Highest Level in the Tree

From the **View** menu, select **Refresh**. This collapses the tree so that only the highest level nodes are shown.

Show Only Nodes That Have Requirements

From the **View** menu, select **Show Requirement Nodes** or click the  button.

Only nodes that have requirements associated with them are shown. Other nodes are hidden. When a node is highlighted, the full path of the node is shown in the righthand pane. The **Filter** display indicates **Requirements Only**.

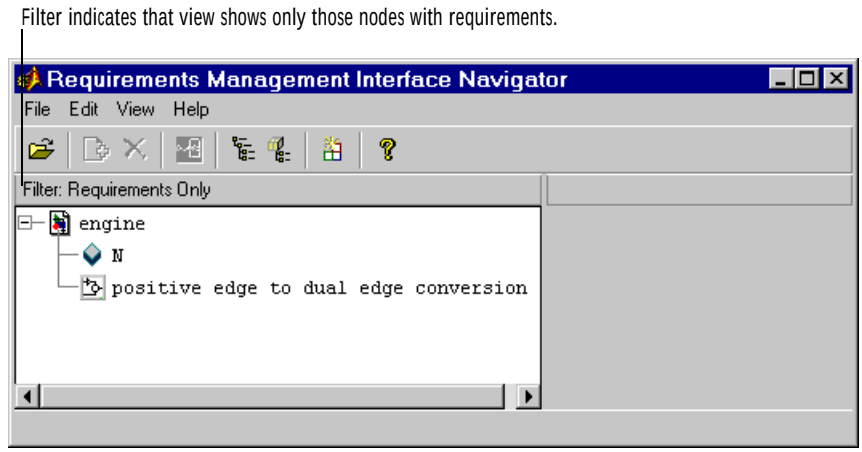


Figure 2-14: Navigator with Requirements Only Filter

When the **Filter** is **Requirements Only**, to see all nodes, select **Show All Nodes** from the **View** menu, or click the button. The **Filter** display then indicates **None**, meaning that there is no filter applied to the requirements listing and all requirements are included.

Going from the Navigator to DOORS

1 In the Navigator, select a node that has an associated requirement.

2 From the **View** menu, select **Navigate to Requirement**, or click the navigate button .

That object becomes highlighted in the DOORS exported module.

Double-clicking in the **Document** or **ID** field for the requirement is another way to go from the Navigator to DOORS.

Going from the Navigator to a MATLAB Object

From the Navigator, you can go to the selected node for an object:

1 In the Navigator, click on a node to select it.

2 From the **View** menu, select **View Block**.

The object opens.

- For a Simulink node, the subsystem containing the selected node opens in Simulink with that block or subsystem selected in the model.
- For a Stateflow node, the diagram containing the selected node opens in Stateflow.
- For an M-file node, the M-file opens in your editor.

Double-clicking on a node is another way to go from the Navigator to the object.

This example shows the results when you select the **N** block in the **valve timing** subsystem in the Navigator and then select **View Block**.

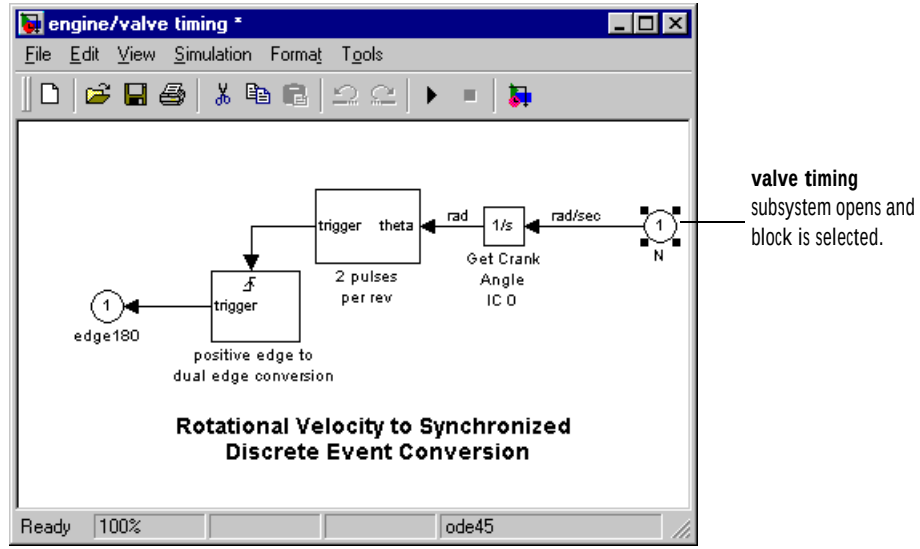


Figure 2-15: Simulink Block Accessed from Navigator

Highlighting Subsystems and Blocks That Have Requirements

You can highlight subsystems and blocks that have requirements in a Simulink model:

1 In the Navigator, select **Highlight Blocks that have Requirements** from the **View** menu, or click the button. The menu item becomes checked.

In the Simulink model, the subsystems and blocks that have requirements associated with them appear highlighted in blue.

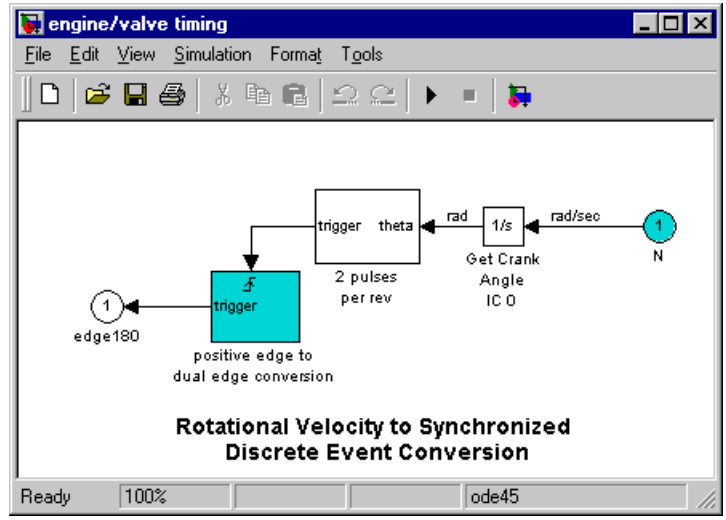


Figure 2-16: Simulink Blocks Highlighted to Indicate Associated Requirements

2 The highlight feature operates as a toggle switch. To turn off the highlighting, select **Highlight Blocks that have Requirements** from the **View** menu; the menu item is no longer checked. Or click the toolbar button.

The blocks and subsystems in the model are no longer highlighted.

Note When you save a Simulink model, it is saved with the highlighting. You *cannot* later turn off the highlighting without opening the model again in the Navigator. Therefore, unless you want the highlighting to become a "permanent" part of the model file, turn off highlighting using the Navigator *before* you save a model file in Simulink.

Going from DOORS to an Object

In DOORS, you can go from a DOORS object in a surrogate exported module to Simulink, Stateflow, or your M-file editor:

1 In the DOORS exported module, click on an object to select it.

That object becomes highlighted.

2 From the **MATLAB** menu, choose **Select item**.

The object opens:

- For a Simulink object, the subsystem containing the selected object opens in Simulink with that block or subsystem selected in the model.
- For a Stateflow object, the diagram containing the selected object opens in Stateflow.
- For an M-file object, the M-file opens in your editor.

If the DOORS **Block Deleted** status for the object is `True`, you cannot navigate to the object.

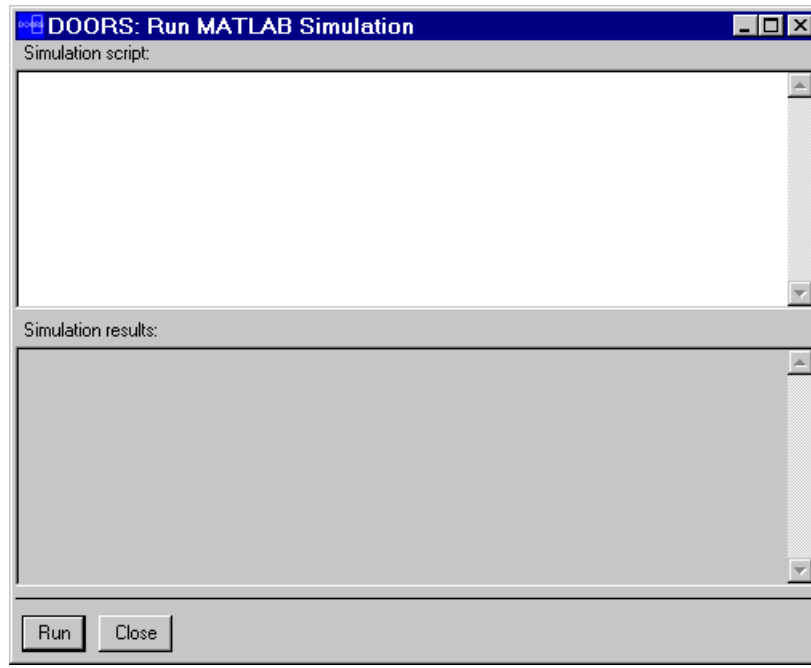
Note Although the **MATLAB** menu and **Select item** feature appear in all DOORS formal modules, you can only use them in a surrogate exported module.

Running a Script from DOORS

From DOORS, you can run a MATLAB script or Simulink simulation to test the behavior of an M-file or model.

1 From the **MATLAB** menu in DOORS, select **Run script**.

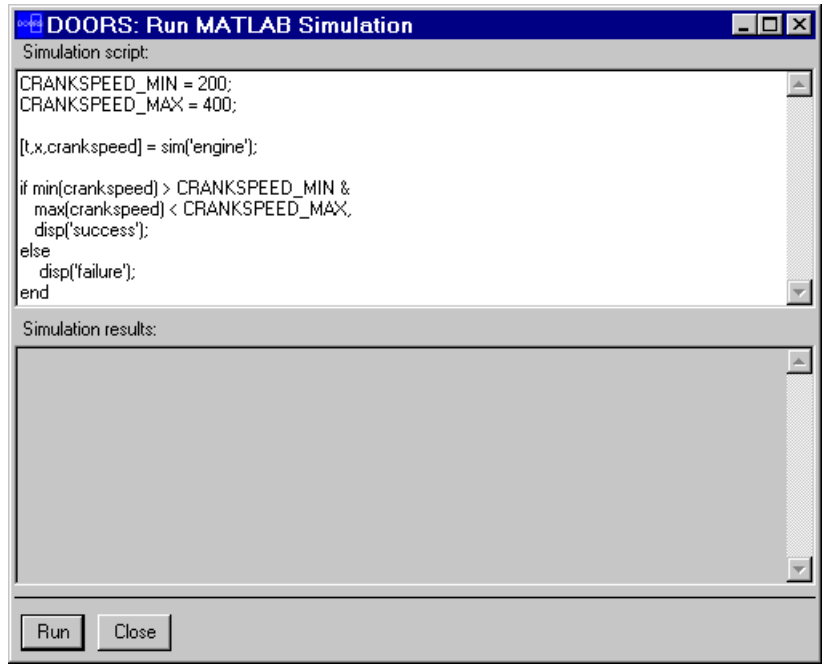
The **Run MATLAB Simulation** dialog box appears.



2 Enter MATLAB code in the **Simulation script** window.

Enter code just as you would within the MATLAB command window. Workspace variables that are current in MATLAB and Simulink are available to you. You can also use `get_param` and `set_param` to access model

parameters. However, do not enter MATLAB comments or Handle Graphics® code. You must return any results as strings.



The sample script in the **Simulation script** window:

a Sets a desired range for the crank speed of the engine by defining values for `CRANKSPEED_MIN` and `CRANKSPEED_MAX`.

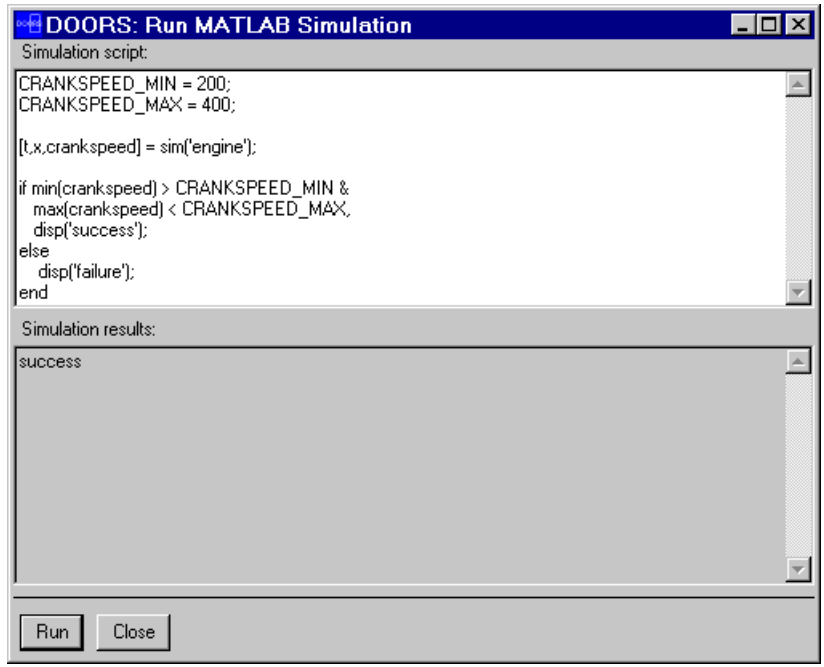
b Runs the simulation.

c Tests the minimum and maximum values in the vector `crankspeed` against the desired minimum and maximum values.

d Displays `success` if all the values in the `crankspeed` vector are within the range or displays `failure` if any value falls outside the range.

3 Click **Run**.

DOORS sends the script to MATLAB. MATLAB runs the M-file or simulation and then returns the results as a string to DOORS, which displays the results in the **Simulation results** window.



MATLAB also returns error messages to the **Simulation results** window.

4 Vary the script to test other simulations.

In this script you could vary the values of `CRANKSPEED_MIN` and `CRANKSPEED_MAX`.

5 Click **Close**.

DOORS attaches the script to the current DOORS surrogate exported module so it is available the next time you want to run a MATLAB M-file or Simulink simulation from DOORS.

3

Reference

There is only one function for the Requirements Management Interface, `rminav`.

rminav

Purpose

Start the Requirements Management Interface

Graphical Interface

As an alternative to the rminav function, in Simulink or Stateflow, select **Requirements management interface** from the **Tools** menu.

Syntax

```
rminav
```

Description

`rminav` starts the Requirements Management Interface Navigator window.

If you specified `reqsys = 'OTHERS'` in the MATLAB file `reqmgropts.m`, the standard version of the Navigator window opens. You can associate requirements documents written in HTML, Microsoft Word, or Microsoft Excel with Simulink models, Stateflow diagrams, and MATLAB M-files.

If you specified `reqsys = 'DOORS'` in `reqmgropts.m`, the DOORS version of the Navigator window opens. You can associate QSS DOORS requirements with Simulink models, Stateflow diagrams, and MATLAB M-files.

To associate DOORS requirements with MATLAB objects, you must start MATLAB with the `/automation` option.

See the *Requirements Management Interface User's Guide* for instructions to use the Navigator window.

Index

N

O

R

S

T