Requirements Management Interface

For Use with Simulink®

Modeling

Simulation

Implementation

User's Guide

Version 1

The MathWorks

How to Contact The MathWorks:

`www.mathworks.com`	Web
`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
`service@mathworks.com`	Order status, license renewals, passcodes
`info@mathworks.com`	Sales, pricing, and general information
508-647-7000	Phone
508-647-7001	Fax

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

For contact information about worldwide offices, see the MathWorks Web site.

Requirements Management Interface User's Guide

Printing History:			
	January 2000	First printing	New for Version 1 (Release 11)
	November 2000	Online only	Version 1.01 (Release 12)
	July 2001	Online only	Version 1.02 (Release 12.1)
	July 2002	Online only	Version 1.04 (Release 13)

Contents

1 Getting Started

2 Using the Requirements Management Interface

3 Using the Requirements Management Interface for DOORS

4 Function Reference

Getting Started

This chapter introduces the Requirements Management Interface. The following sections are included:

"What Is the Requirements Management Interface?" on page 1-3	Provides an overview of the Requirements Management Interface, both the standard version and the DOORS version.
"Configuration Information" on page 1-5	Provides information about system requirements and installation.
"Related Products" on page 1-7	Describes other MathWorks products that are especially relevant to the kinds of tasks you can perform with the Requirements Management Interface.
"Using This Guide" on page 1-8	Describes the expected background for using the Requirements Management Interface, the organization of this manual, and the examples used in this manual.
"Typographical Conventions" on page 1-12	Summarizes the typographical conventions used in this manual.

What Is the Requirements Management Interface?

This section provides an overview of the Requirements Management Interface. The following topics are included:

“Description” on page 1-4	Introduces both the standard version and the DOORS version of the Requirements Management Interface.
“Features” on page 1-4	Describes the capabilities of the Requirements Management Interface.

Description

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.

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).

Configuration Information

This section provides information about setting up the Requirements Management Interface. Topics covered include:

"System Requirements" on page 1-5	Describes the specific system requirements for the Requirements Management Interface.
"Installation" on page 1-6	Provides installation information.

System Requirements

The Requirements Management Interface works on

- Microsoft Windows 98, Windows NT, Windows 2000, and Windows XP 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 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 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 later
  - Microsoft Excel 97 or later
  - Netscape Navigator Version 4.0 or later, or Microsoft Internet Explorer Version 4.0 or later for HTML documents
- For the DOORS version of the Requirements Management Interface, you need Telelogic, Inc.'s DOORS requirements management system, Version 4.0.3 or later.
- To use version 5 of DOORS, you need version 1.02 of the Requirements Management Interface.

- To use version 5.2 of DOORS, you need version 1.04 of the Requirements Management Interface.

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.

Related Products

The MathWorks provides several products that are especially relevant to the kinds of tasks you can perform with the Requirements Management Interface.

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.

Product	Description
MATLAB	The Language of Technical Computing
Simulink	Design and simulate continuous- and discrete-time systems
Stateflow	Design and simulate event-driven systems

Using This Guide

This section provides information on how to use the Requirements Management Interface documentation. The following topics are included:

"Expected Background" on page 1-9	Describes the background needed to use the Requirements Management Interface.
"Organization of the Document" on page 1-9	Provides overview of this documentation.
"Examples Used" on page 1-10	Introduces the Simulink, Stateflow, and MATLAB M-file examples that are used in this documentation.

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

This document contains examples to help you to become familiar with either the standard version or the DOORS version of the Requirements Management Interface.

“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 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`.

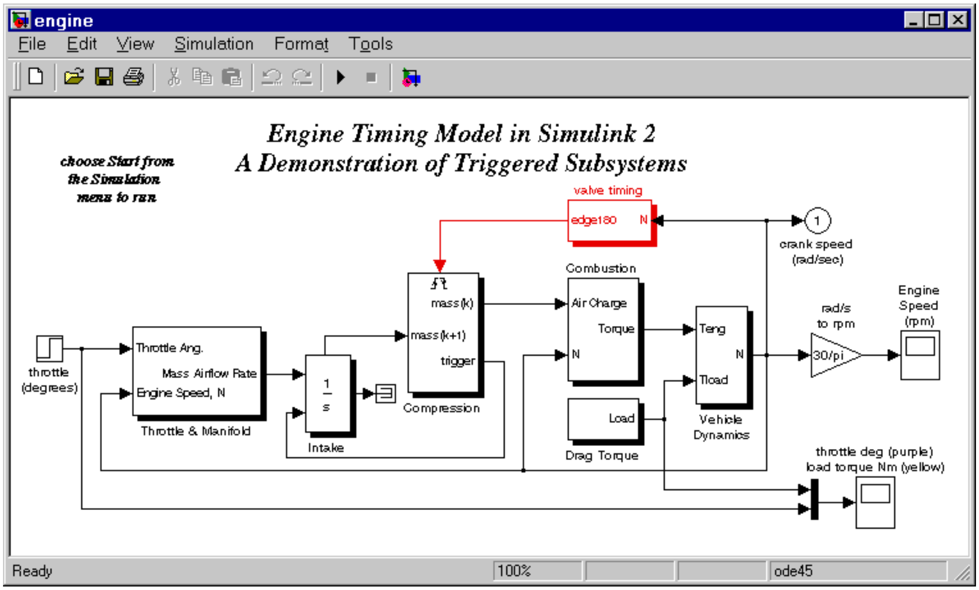


Figure 1-1: Simulink engine Model

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`.

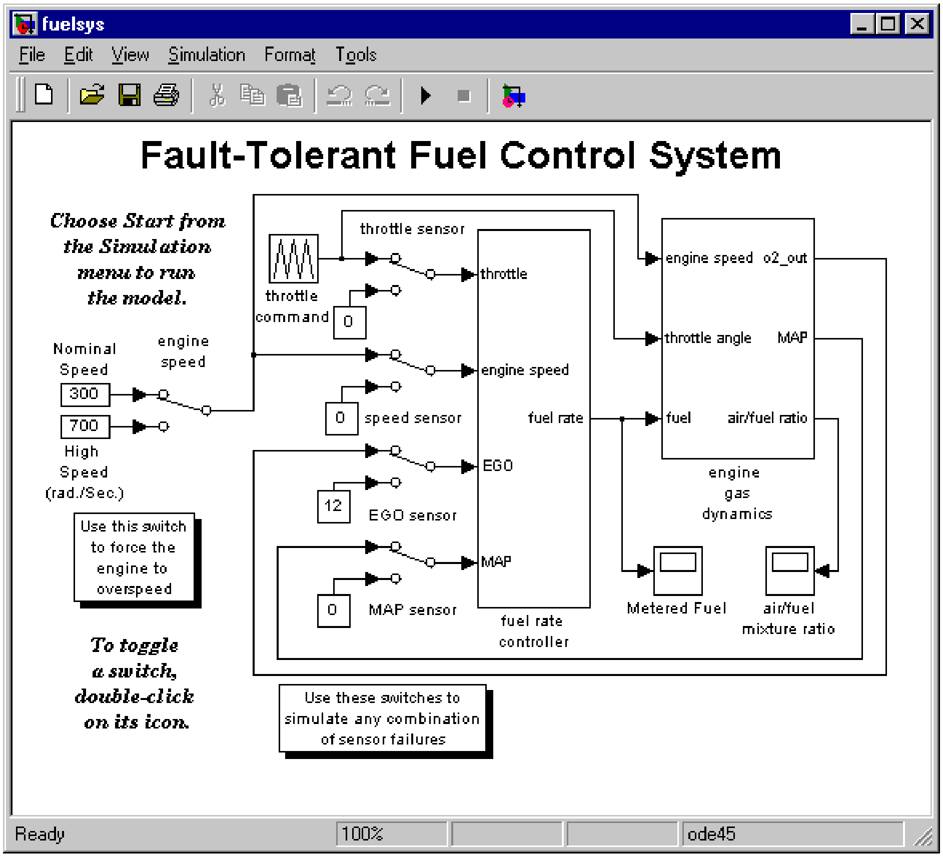


Figure 1-2: fuelsys Model with Stateflow Fuel Rate Controller Chart

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.

Typographical Conventions

This manual uses some or all of these conventions.

Item	Convention	Example
Example code	`Monospace` font	To assign the value 5 to `A`, enter `A = 5`
Function names, syntax, filenames, directory/folder names, and user input	`Monospace` font	The `cos` function finds the cosine of each array element. Syntax line example is `MLGetVar ML_var_name`
Buttons and keys	**Boldface** with book title caps	Press the **Enter** key.
Literal strings (in syntax descriptions in reference chapters)	**`Monospace bold`** for literals	`f = freqspace(n,'`**`whole`**`')`
Mathematical expressions	*Italics* for variables Standard text font for functions, operators, and constants	This vector represents the polynomial $p = x^2 + 2x + 3$.

MATLAB output	`Monospace font`	MATLAB responds with
		`A =`
		`5`

Menu and dialog box titles	**Boldface** with book title caps	Choose the **File Options** menu.
New terms and for emphasis	*Italics*	An *array* is an ordered collection of information.
Omitted input arguments	(...) ellipsis denotes all of the input/output arguments from preceding syntaxes.	`[c,ia,ib] = union(...)`
String variables (from a finite list)	*`Monospace italics`*	`sysc = d2c(sysd,'`*`method`*`')`

2

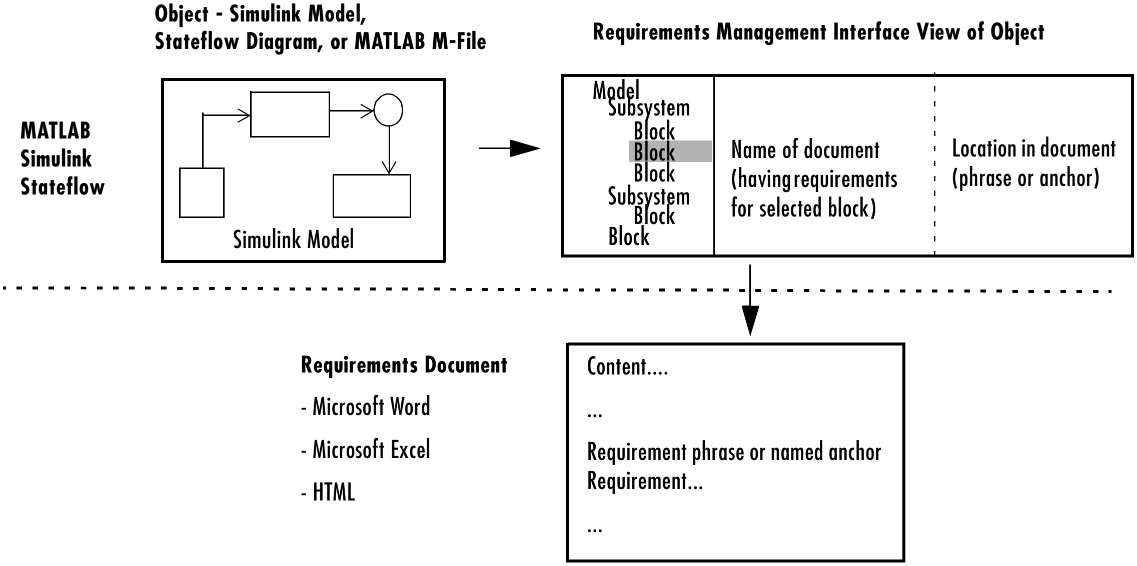
Using the Requirements Management Interface

Using the Requirements Management Interface provides an overview of how the Requirements Management Interface associates objects with requirements documents and the instructions to use the Requirements Management Interface. The following sections are included:

"Requirements Management Interface Standard Version" on page 2-3	Shows how the Requirements Management Interface associates objects with requirements documents.
"Adding Requirements to an Object" on page 2-4	Describes how to start the Requirements Management Interface, open objects, and associate requirements documents with objects.
"Viewing Requirements in the Navigator" on page 2-13	Describes how to view and modify requirements, and how to navigate between Simulink, Stateflow, and M-file objects and their requirements documents.

Requirements Management Interface Standard Version

This diagram shows how the Requirements Management Interface associates objects with requirements documents. This diagram is useful if you are new to requirements management and need an overview of the process.


Object - Simulink Model,
Stateflow Diagram, or MATLAB M-File
Requirements Management Interface View of Object
MATLAB
Simulink
Stateflow
Simulink Model
Model
Subsystem
Block
Block
Block
Subsystem
Block
Block
Name of document (having requirements for selected block)
Location in document (phrase or anchor)
Requirements Document
- Microsoft Word
- Microsoft Excel
- HTML
Content....
...
Requirement phrase or named anchor
Requirement...
...

Adding Requirements to an Object

This section shows how to open a Simulink, Stateflow, or M-file object and associate requirements with it. This section includes the following topics:

"Starting the Requirements Management Interface" on page 2-5	Start the Requirements Management Interface.
"Opening an Object in the Navigator" on page 2-6	Open Simulink, Stateflow, and M-file objects.
"Adding Requirements" on page 2-9	Associate single or multiple requirements with objects.

Starting the Requirements Management Interface

Use this procedure to start the Requirements Management Interface. This is done prior to opening objects and adding requirements to the objects.

Starting from MATLAB Command Window

1 Start MATLAB.

2 Change current directory to `$matlabroot\toolbox\reqmgt`.

3 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.

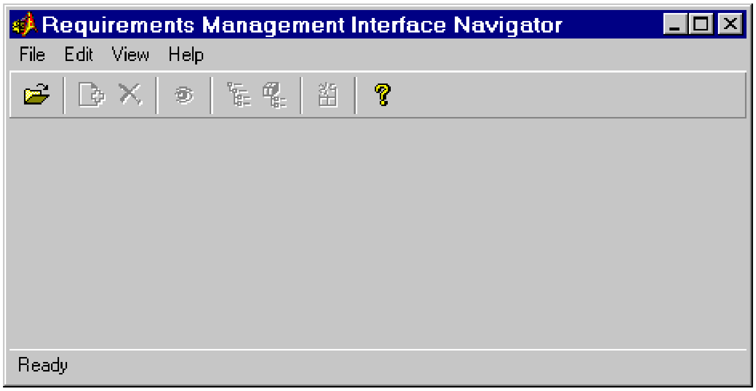
```
        reqsys = 'OTHERS';
```

   c Save `reqmgropts.m` and close it.

4 At the MATLAB prompt, type

```
rminav
```

The **Navigator** window opens.



Note From here forward “Requirements Management Interface Navigator” will be referred to as “Navigator.”

Starting from a Simulink or a Stateflow Model Window

Select **Requirements management interface** from the **Tools** menu.

Opening an Object in the Navigator

Use the Navigator to open the object that needs requirements added or edited. This procedure should be followed after the Navigator is started.

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 associate requirements with a Simulink or Stateflow object that is already open and is the currently active model,

Select **Open Current Model** from the **File** menu in the Navigator

Note The top level system of the Simulink or Stateflow model must be active. For example, if the Simulink f14 model is open with the controller subsystem active, then **Open Current Model** will not work in the Navigator.

To associate requirements with a Simulink or Stateflow object that is not currently open, or if you want to associate requirements with a MATLAB M-file:

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

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

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

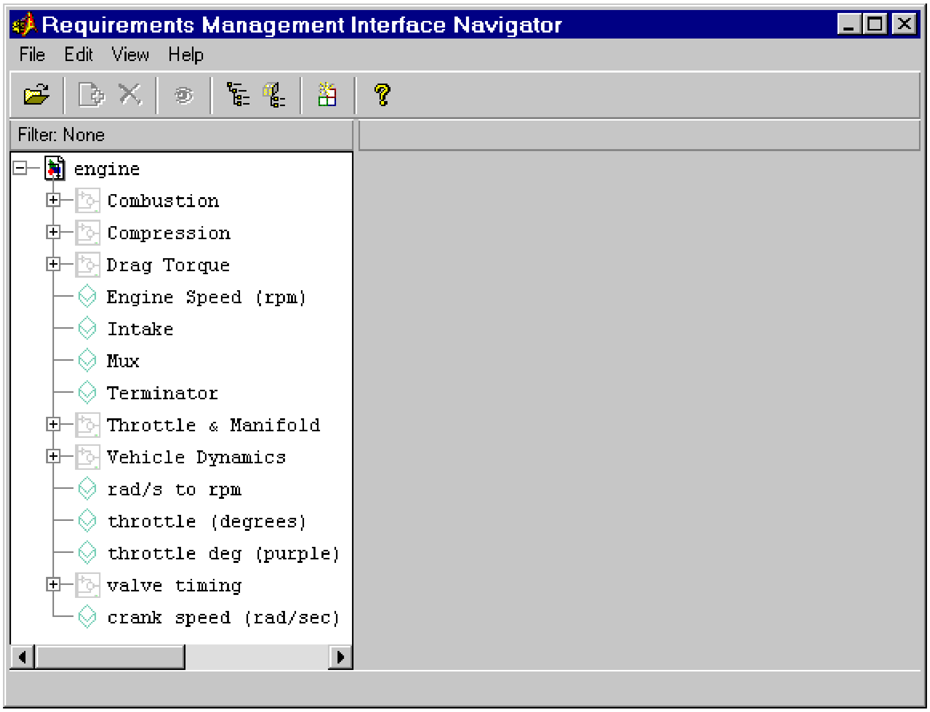
   The object opens in Simulink or Stateflow.

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. The following examples show the Navigator's views of Simulink, Stateflow, and M-file objects.

Simulink Chart in the Navigator

This example shows the Simulink engine model in the Navigator.

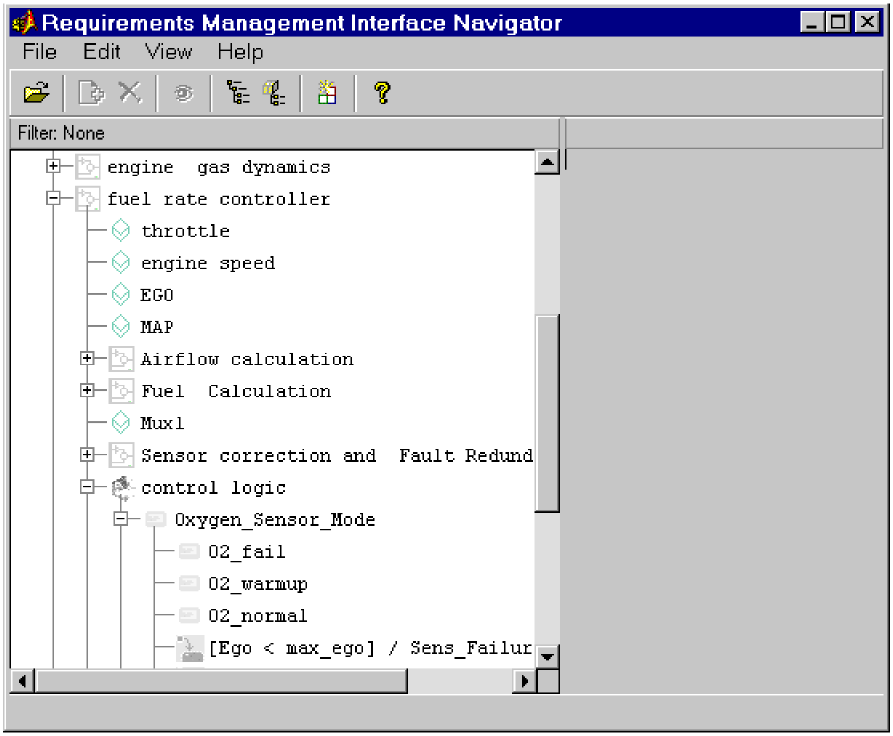


```
engine
    Combustion
    Compression
    Drag Torque
    Engine Speed (rpm)
    Intake
    Mux
    Terminator
    Throttle & Manifold
    Vehicle Dynamics
    rad/s to rpm
    throttle (degrees)
    throttle deg (purple)
    valve timing
    crank speed (rad/sec)
```

Stateflow Chart in the Navigator

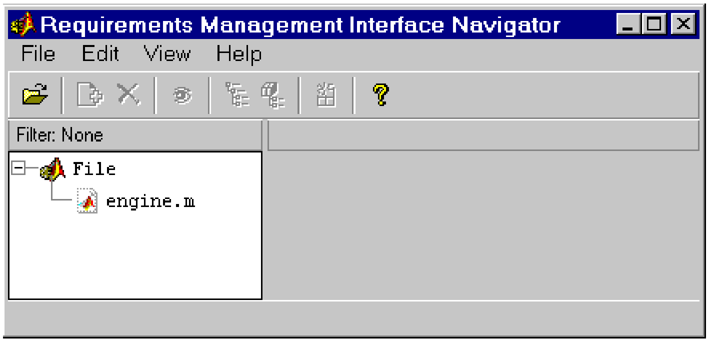
This example shows a partial view of the `fuelsys` model, which contains the **control logic** Stateflow chart in the **fuel rate controller** subsystem.

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



MATLAB M-File in Navigator

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



Adding Requirements

After opening an object in the Navigator, you can associate a node in the hierarchical tree with requirements 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.

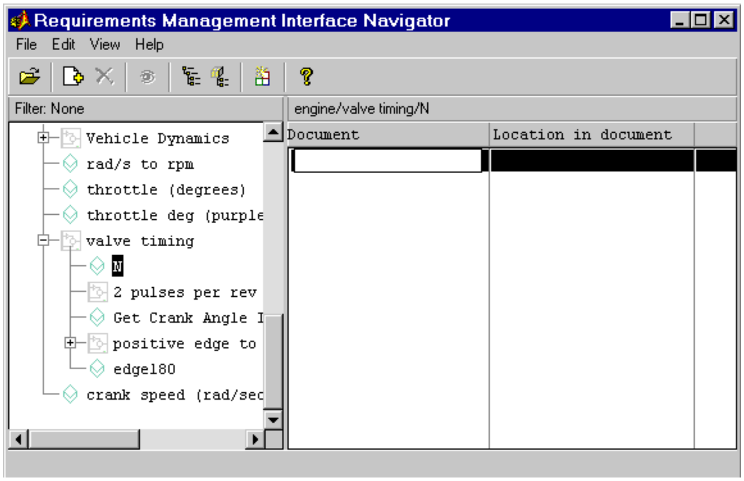
Note Stateflow allows for multiple transitions at the same level to have the same name. However, the Requirements Management Interface does not support associating requirements documents with transition peers that have the same name. For example, Stateflow, by default, assigns '?' as the transition name. If you add two transitions at the same level you will have two transitions named '?'. Using the Navigator to associate requirements documents to these nodes will result in improper tracking of the requirements documents.

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 the ⊞ button 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.



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

Here are examples of full pathnames.

Pathname	For Document Type
`c:\Requirements\Car Requirements.doc`	Word
`d:\car_files\car_reqts.xls`	Excel

Pathname	For Document Type
`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.

To add the location, 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.

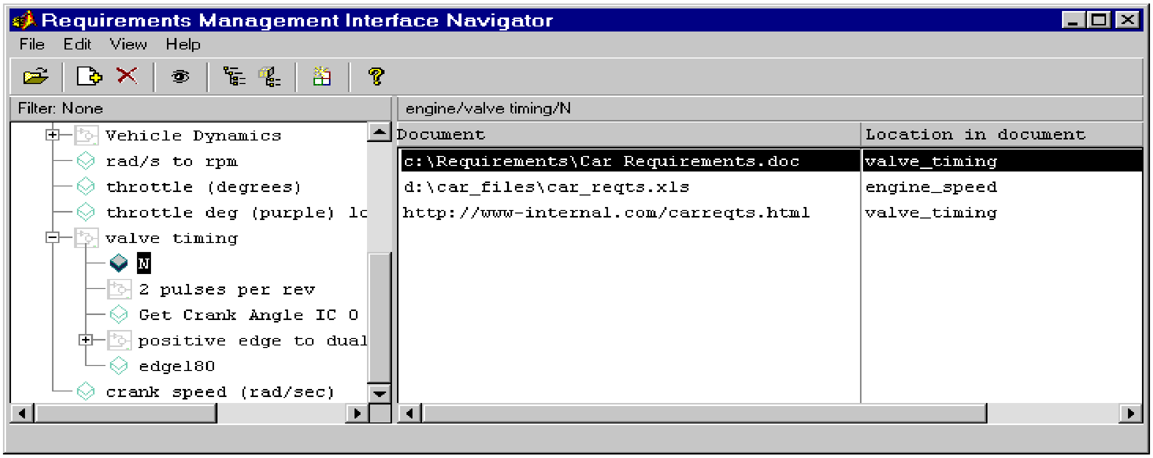
Note If your Stateflow chart contains transitions that are not unique, do not associate requirements with them.

About the Requirements Documents

The requirements document can be in Microsoft Word, Microsoft Excel, or HTML:

- 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">`.

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



Filter: None

- Vehicle Dynamics
- rad/s to rpm
- throttle (degrees)
- throttle deg (purple) lc
- valve timing
  - N
  - 2 pulses per rev
  - Get Crank Angle IC 0
  - positive edge to dual
  - edge180
- crank speed (rad/sec)

engine/valve timing/N

Document	Location in document
c:\Requirements\Car Requirements.doc	valve_timing
d:\car_files\car_reqts.xls	engine_speed
http://www-internal.com/carreqts.html	valve_timing

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

This section shows how to view, modify, and edit requirements and to navigate between the Requirements Management Interface Navigator and objects and requirements documents. This section includes the following topics:

"Viewing the Requirements for a Node" on page 2-14	See which nodes have requirements associated with them.
"Changing or Deleting a Requirement" on page 2-18	Edit requirements that are associated with objects.
"Changing the View in the Navigator" on page 2-19	Optimize the tree view of a Simulink model, Stateflow model, or M-file object.
"Viewing the Requirements Document" on page 2-21	Go to a requirements document and view details.
"Viewing an Object from the Navigator" on page 2-21	Go to an object and evaluate the Simulink model, Stateflow diagram, or MATLAB M-file for additional requirements needs.
"Highlighting Subsystems and Blocks with Requirements" on page 2-22	Modify the view of a Simulink model and identify the subsystems and blocks that have associated requirements.

Viewing the Requirements for a Node

Use the Navigator to see which nodes have requirements associated with them. This is done when you are reviewing the objects (Simulink model, Stateflow diagram, or MATLAB M-file) requirements.

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

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

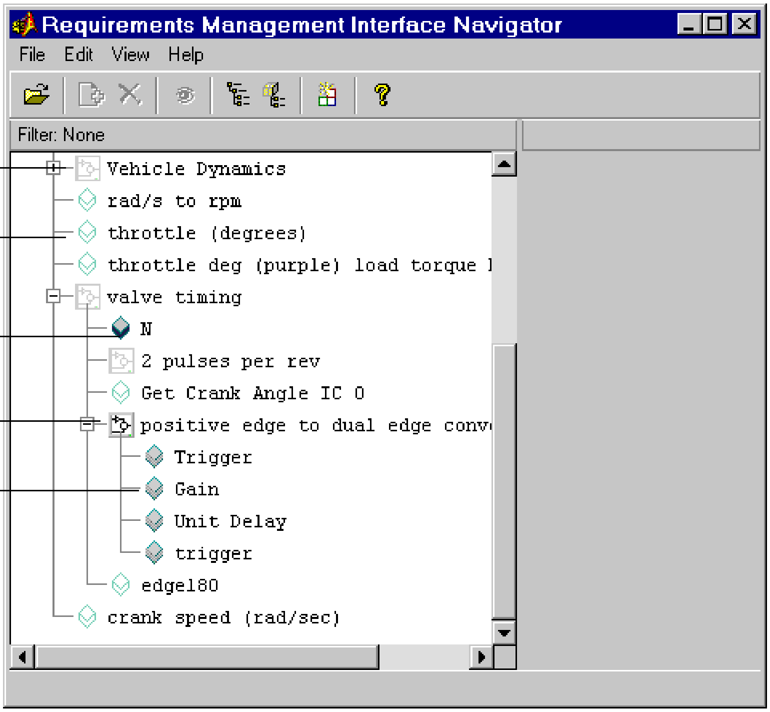
Table 2-1: Node icons in the Navigator and What They Represent

□	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

Table 2-1: Node icons in the Navigator and What They Represent (Continued)

	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

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.



Unshaded subsystem-no requirement

Unshaded block -no requirement

Shaded block - has requirement

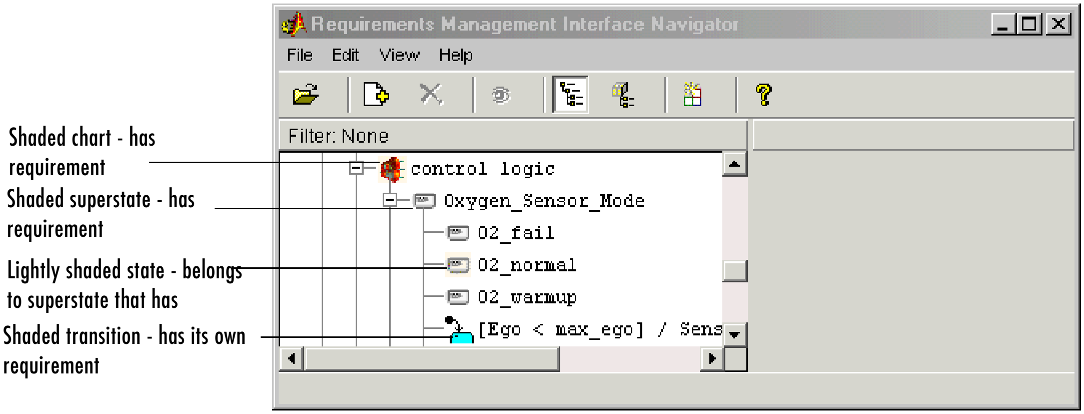
Shaded subsystem - has requirement

Lightly shaded block - belongs to subsystem that has requirement

Stateflow Chart with Requirements

Note Stateflow allows for multiple transitions at the same level to have the same name. However, the Requirements Management Interface does not support associating requirements documents with transition peers that have the same name.

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



Shaded chart - has requirement

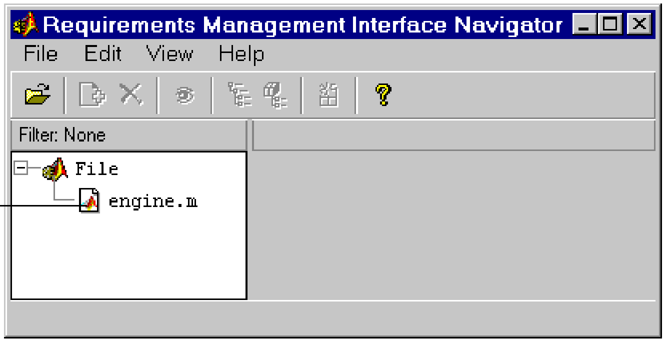
Shaded superstate - has requirement

Lightly shaded state - belongs to superstate that has

Shaded transition - has its own requirement

MATLAB M-File with Requirements

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



Shaded M-file icon - has requirement

Changing or Deleting a Requirement

Objects with associated requirements can be edited in the Navigator to keep the requirement information current.

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

The Navigator view can be changed to optimize the display of the object's requirements.

After the requirements have been added, you can change the view in the Navigator 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

1 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.

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

Show or Hide Lower Levels in the Tree

1 Click the ⊞ button to expand a node in the tree, which shows lower levels.

2 Click the ⊟ button to collapse a node in the tree, which hides the lower levels.

Show Only the Highest Level in the Tree

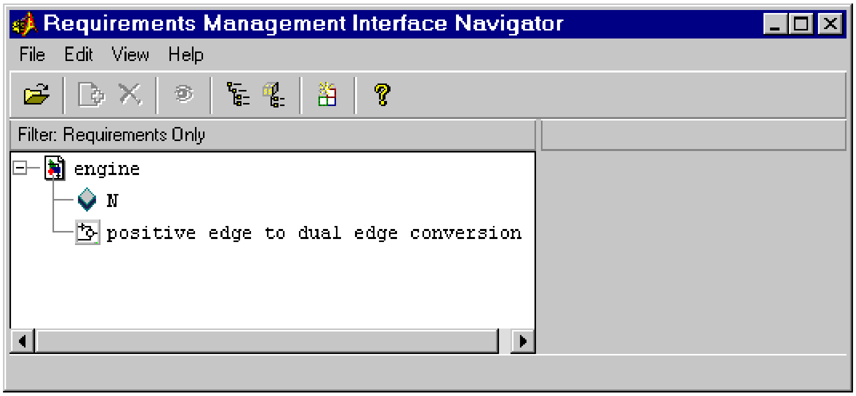
1 From the **View** menu, select **Refresh**.

This collapses the tree so that only the highest level nodes are shown.

Show Only Nodes That Have Requirements

1 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**.


Requirements Management Interface Navigator
File Edit View Help
Filter: Requirements Only
engine
N
positive edge to dual edge conversion

2 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.

Viewing the Requirements Document

You can use the Navigator to go to a requirements document to view the requirements details.

This can be done after requirements have been associated with the object:

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.

Viewing an Object from the Navigator

You can use the Navigator to go to an Object to evaluate the Simulink model, Stateflow diagram, or MATLAB M-file for additional requirements needs.

This can be done after requirements have been associated with the object.

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

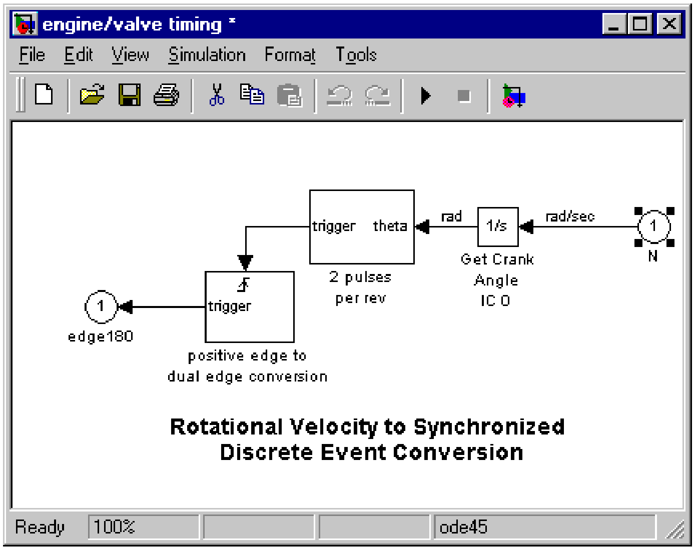
1. In the Navigator, select a node.
2. From the **View** menu, select **View Block**.

   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**.


engine/valve timing *
File Edit View Simulation Format Tools
trigger theta
2 pulses per rev
rad
1/s
Get Crank Angle IC 0
rad/sec
1
N
trigger
positive edge to dual edge conversion
1
edge180
Rotational Velocity to Synchronized Discrete Event Conversion
Ready
100%
ode45

Highlighting Subsystems and Blocks with Requirements

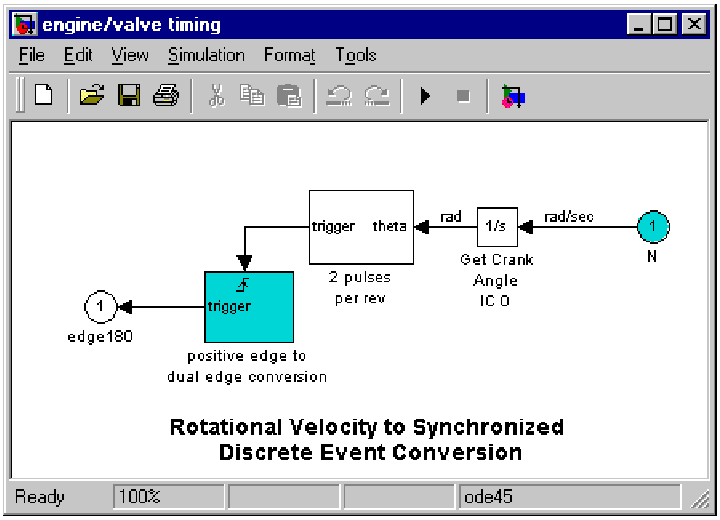
You can modify the view of a Simulink model to identify the subsystems and blocks that have associated requirements.

After the requirements have been associated with a Simulink model with the Requirements Management Interface, you can highlight subsystems and blocks that have requirements:

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.



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.

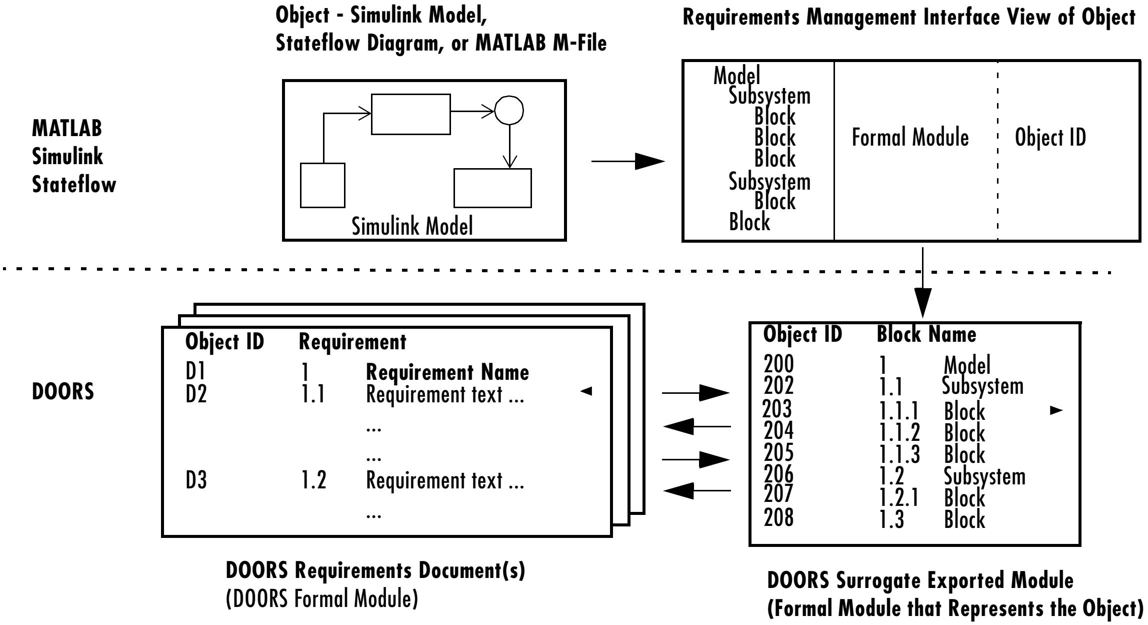
Using the Requirements Management Interface for DOORS

Using the Requirements Management Interface for DOORS provides an overview of how objects are associated with DOORS requirements and the instructions to use the Requirements Management Interface with DOORS. The following sections are included:

"Requirements Management Interface DOORS Version" on page 3-3	Shows how the Requirements Management Interface associates objects with DOORS requirements.
"Linking Objects to DOORS Requirements" on page 3-5	Describes how to start the Requirements Management Interface, open objects in the navigator, synchronize the Navigator, and link objects with DOORS requirements.
"Viewing Requirements in the Navigator" on page 3-18	Describes how to move between the Navigator, the DOORS surrogate exported module, and MATLAB objects and how to view which nodes in an object have requirements associated with them.
"Running a Script from DOORS" on page 3-29	Describes how to run a MATLAB script or Simulink simulation from DOORS to test the behavior of an M-file or model.

Requirements Management Interface DOORS Version

This diagram depicts how the Requirements Management Interface associates objects with DOORS requirements. This diagram is useful if you are new to requirements management and need an overview of the process.



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.

Linking Objects to DOORS Requirements

This section shows how to start the Requirements Management Interface, open objects in the navigator, synchronize the Navigator, and link objects with DOORS requirements. This section includes the following topics:

"Starting the Requirements Management Interface" on page 3-6	Open the Requirements Management Interface for DOORS.
"Opening an Object in the Navigator" on page 3-7	Open Simulink, Stateflow, and M-file objects in the Navigator.
"Synchronizing the Navigator with DOORS" on page 3-11	Describes why synchronization is necessary, when to synchronize, and how to synchronize.
"Linking an Object to a DOORS Requirement" on page 3-16	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.

Starting the Requirements Management Interface

Use this procedure to start the Requirements Management Interface for DOORS. This is done prior to opening objects, synchronizing with DOORS and linking objects to DOORS requirements.

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 the MATLAB Requirements Management Interface.

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

   For example, make a shortcut to start MATLAB:

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

Note 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';
      ```

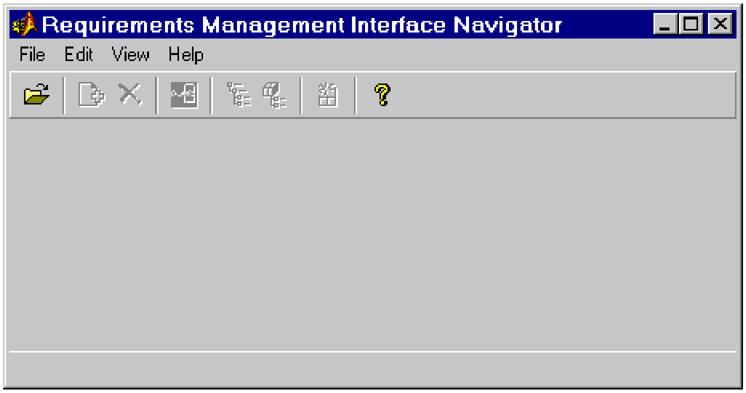
   c Save `reqmgropts.m` and close it.

4 At the MATLAB prompt, type

```
rminav
```

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

The **Navigator** window opens.



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.

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.

Note The top level system of the Simulink or Stateflow model must be active. For example, if the Simulink f14 model is open with the controller subsystem active, then **Open Current Model** will not work 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:

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

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

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

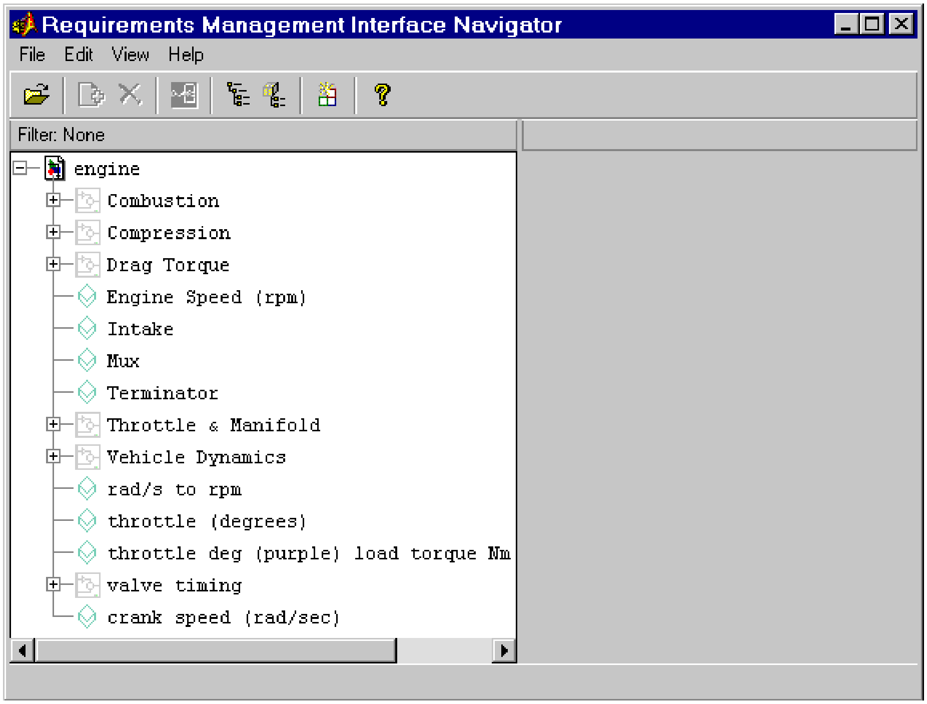
The object opens in Simulink or Stateflow.

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.

Simulink Model in the Navigator

This example shows the Simulink engine model in the Navigator.

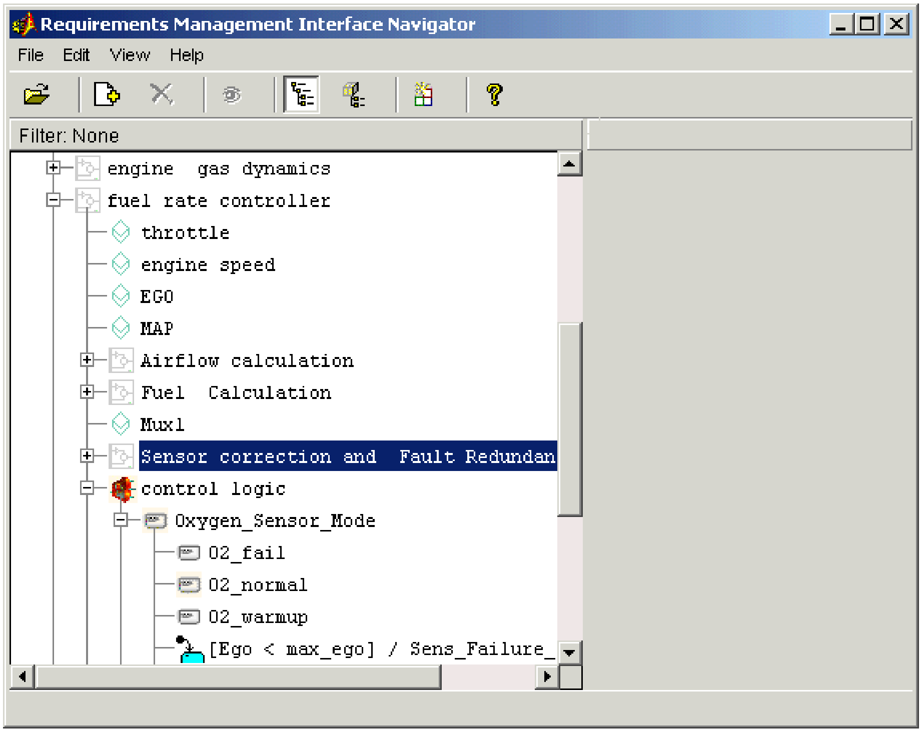


```
engine
  Combustion
  Compression
  Drag Torque
  Engine Speed (rpm)
  Intake
  Mux
  Terminator
  Throttle & Manifold
  Vehicle Dynamics
  rad/s to rpm
  throttle (degrees)
  throttle deg (purple) load torque Nm
  valve timing
  crank speed (rad/sec)
```

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.

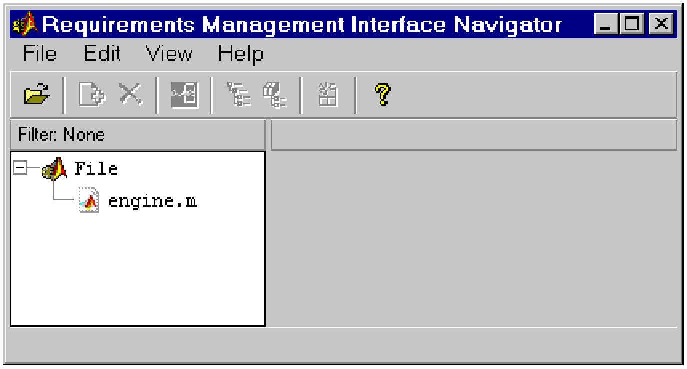
You must have Stateflow installed in order to expand the Stateflow chart node in the Navigator.



```
engine  gas dynamics
fuel rate controller
  throttle
  engine speed
  EGO
  MAP
  Airflow calculation
  Fuel  Calculation
  Mux1
  Sensor correction and  Fault Redundan
  control logic
    Oxygen_Sensor_Mode
      O2_fail
      O2_normal
      O2_warmup
      [Ego < max_ego] / Sens_Failure_
```

MATLAB M-File in Navigator

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



Synchronizing the Navigator with DOORS

You synchronize the object that is open in the Navigator to associate it with a project in DOORS. Synchronization maps the latest version of the DOORS surrogate exported module to the latest version of the Simulink model, Stateflow model, or M-file object.

- 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' }
      ```

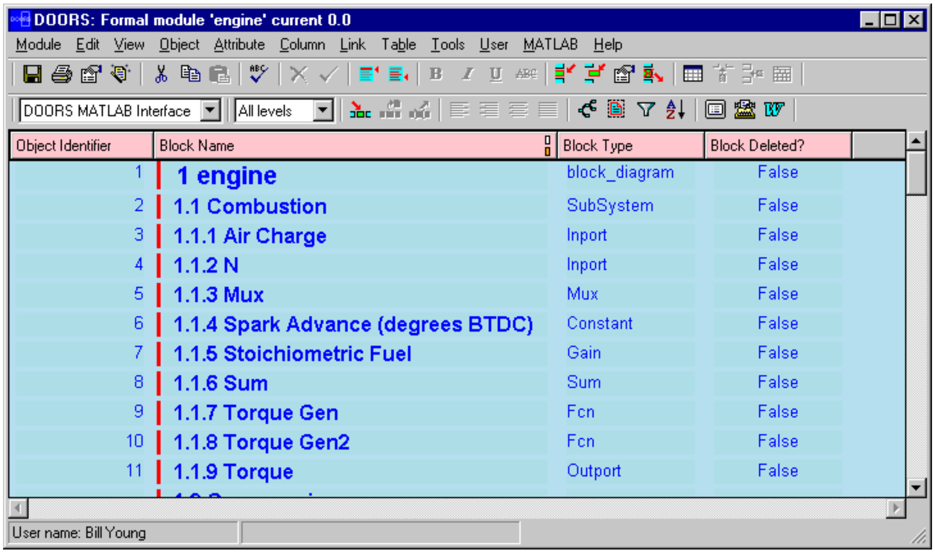
Note You must have write permission for the object in order to synchronize. If you do not have write permission for the object, you can still view the DOORS requirements associated with the object.

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.


DOORS: Formal module 'engine' current 0.0
Module Edit View Object Attribute Column Link Table Tools User MATLAB Help
DOORS MATLAB Interface
All levels
User name: Bill Young

Object Identifier	Block Name	Block Type	Block Deleted?
1	1 engine	block_diagram	False
2	1.1 Combustion	SubSystem	False
3	1.1.1 Air Charge	Inport	False
4	1.1.2 N	Inport	False
5	1.1.3 Mux	Mux	False
6	1.1.4 Spark Advance (degrees BTDC)	Constant	False
7	1.1.5 Stoichiometric Fuel	Gain	False
8	1.1.6 Sum	Sum	False
9	1.1.7 Torque Gen	Fcn	False
10	1.1.8 Torque Gen2	Fcn	False
11	1.1.9 Torque	Outport	False

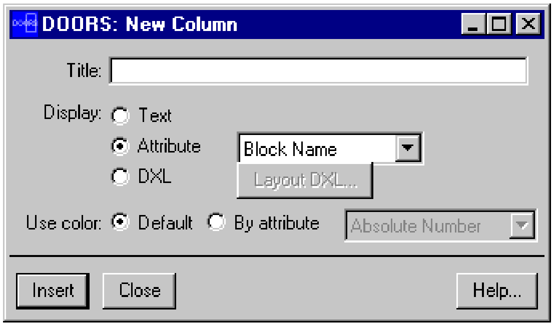
Note that:

- 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.



DOORS: New Column

Title:

Display: ○ Text ● Attribute ○ DXL

Block Name

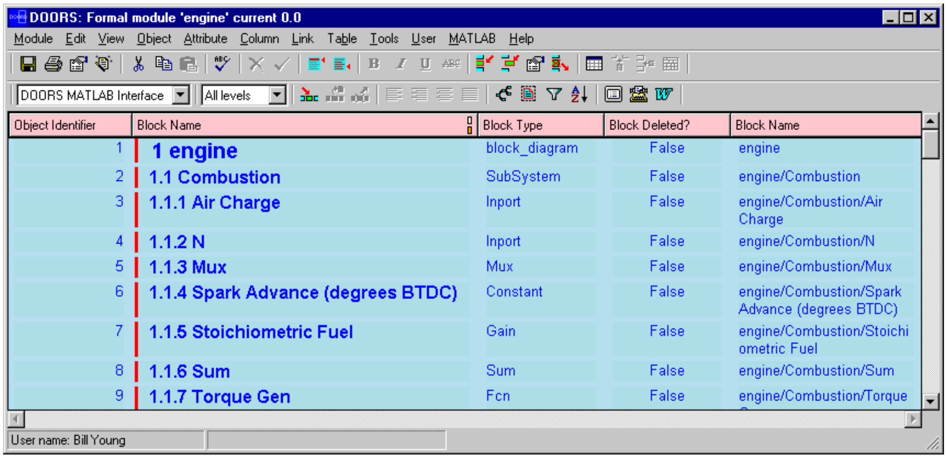
Use color: ● Default ○ By attribute

Absolute Number

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.



DOORS: Formal module 'engine' current 0.0

Object Identifier	Block Name	Block Type	Block Deleted?	Block Name
1	1 engine	block_diagram	False	engine
2	1.1 Combustion	SubSystem	False	engine/Combustion
3	1.1.1 Air Charge	Inport	False	engine/Combustion/Air Charge
4	1.1.2 N	Inport	False	engine/Combustion/N
5	1.1.3 Mux	Mux	False	engine/Combustion/Mux
6	1.1.4 Spark Advance (degrees BTDC)	Constant	False	engine/Combustion/Spark Advance (degrees BTDC)
7	1.1.5 Stoichiometric Fuel	Gain	False	engine/Combustion/Stoichiometric Fuel
8	1.1.6 Sum	Sum	False	engine/Combustion/Sum
9	1.1.7 Torque Gen	Fcn	False	engine/Combustion/Torque

User name: Bill Young

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.

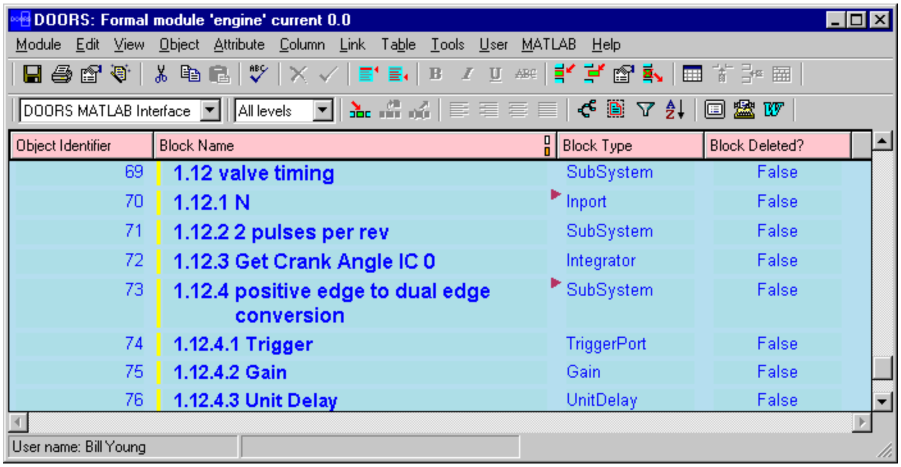
4. Right-click on the selected object, and from the pop-up menu select **Link>Make link from start**. Or from the **Link** menu, select **Make link from start**.

   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. Stateflow allows for multiple transitions at the same level to have the same name. However, the Requirements Management Interface does not support associating requirements documents with transition peers that have the same name

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.



DOORS MATLAB Interface

All levels

Object Identifier	Block Name	Block Type	Block Deleted?
69	1.12 valve timing	SubSystem	False
70	1.12.1 N	Inport	False
71	1.12.2 2 pulses per rev	SubSystem	False
72	1.12.3 Get Crank Angle IC 0	Integrator	False
73	1.12.4 positive edge to dual edge conversion	SubSystem	False
74	1.12.4.1 Trigger	TriggerPort	False
75	1.12.4.2 Gain	Gain	False
76	1.12.4.3 Unit Delay	UnitDelay	False

User name: Bill Young

Viewing Requirements in the Navigator

This section shows how to move between the Navigator, the DOORS surrogate exported module, and MATLAB objects and how to view which nodes in an object have requirements associated with them.

Viewing the Requirements for a Node

Use the Navigator to see which nodes have requirements associated with them. This is useful when you are reviewing the Simulink, Stateflow, or M-file object's requirements.

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

Table 3-1: Node Icons in the Navigator and What They Represent

□	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

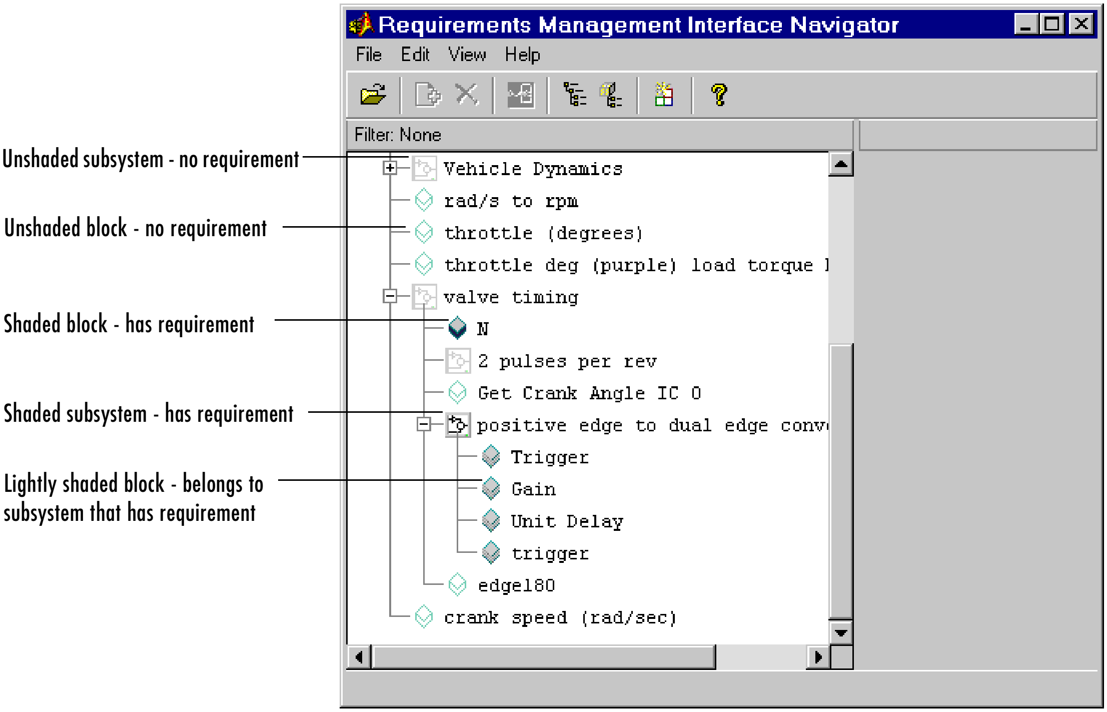
Table 3-1: Node Icons in the Navigator and What They Represent (Continued)

	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

To view the requirements for a node:

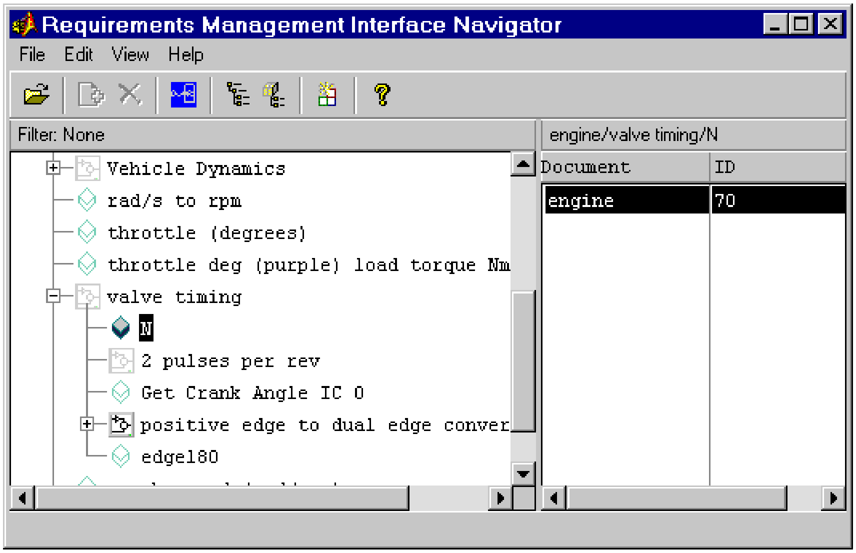
1 Select **Synchronize** from the **File** menu in the Navigator so that the links you just made in DOORS are reflected in the Navigator.

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.


Requirements Management Interface Navigator
File Edit View Help
Filter: None
Vehicle Dynamics
rad/s to rpm
throttle (degrees)
throttle deg (purple) load torque l
valve timing
N
2 pulses per rev
Get Crank Angle IC 0
positive edge to dual edge conv
Trigger
Gain
Unit Delay
trigger
edge180
crank speed (rad/sec)
Unshaded subsystem - no requirement
Unshaded block - no requirement
Shaded block - has requirement
Shaded subsystem - has requirement
Lightly shaded block - belongs to subsystem that has requirement

2 Select a node that has requirements associated with it. If the node contains other nodes, first click the ⊞ button 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.



Filter: None

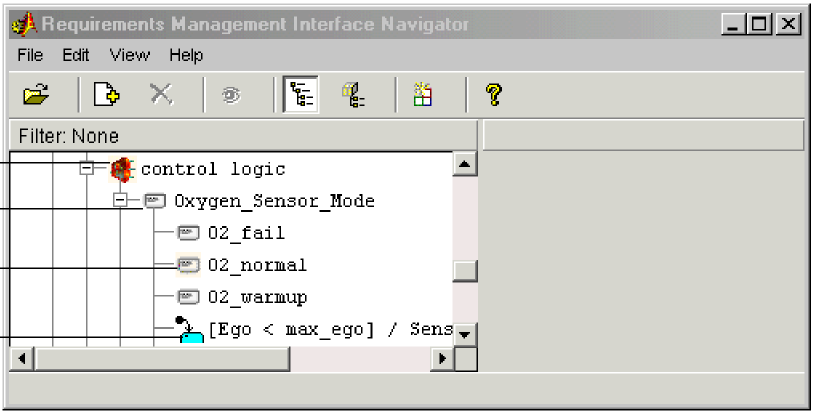
- Vehicle Dynamics
- rad/s to rpm
- throttle (degrees)
- throttle deg (purple) load torque Nm
- valve timing
  - N
  - 2 pulses per rev
  - Get Crank Angle IC 0
  - positive edge to dual edge conver
  - edge180

engine/valve timing/N

Document	ID
engine	70

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.



Filter: None

- control logic
  - Oxygen_Sensor_Mode
    - O2_fail
    - O2_normal
    - O2_warmup
    - [Ego < max_ego] / Sens

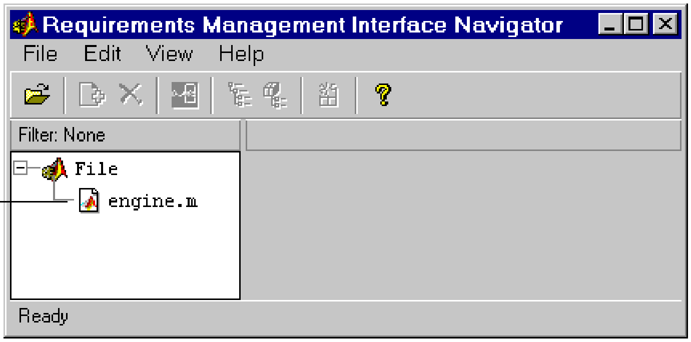
Shaded chart - has requirement

Shaded superstate - has requirement

Lightly shaded state - belongs to superstate that has

Shaded transition - has its own requirement

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



Shaded M-file icon - has requirement

Changing the View in the Navigator

Use the Navigator to optimize the tree view of the Simulink, Stateflow, or M-file object. 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 the ⊞ button to expand a node in the tree, which shows lower levels.
- Click the ⊟ button to collapse a node in the tree, which hides the lower levels.

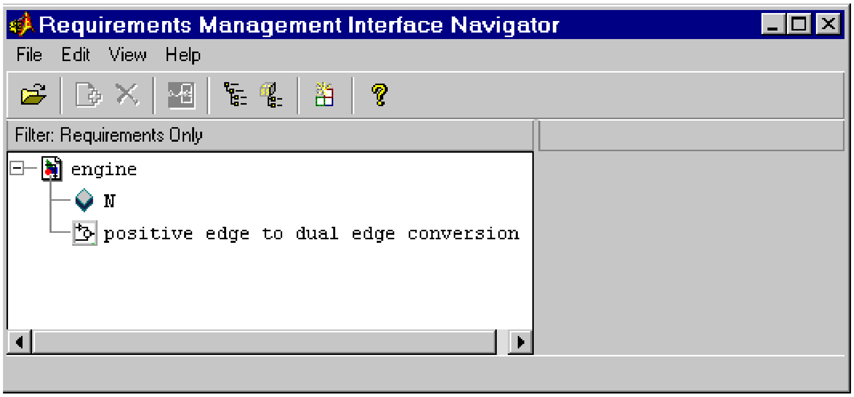
Show Only the Highest Level in the Tree

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 right pane. The **Filter** display indicates **Requirements Only**.



Filter: Requirements Only

```
engine
  N
  positive edge to dual edge conversion
```

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.

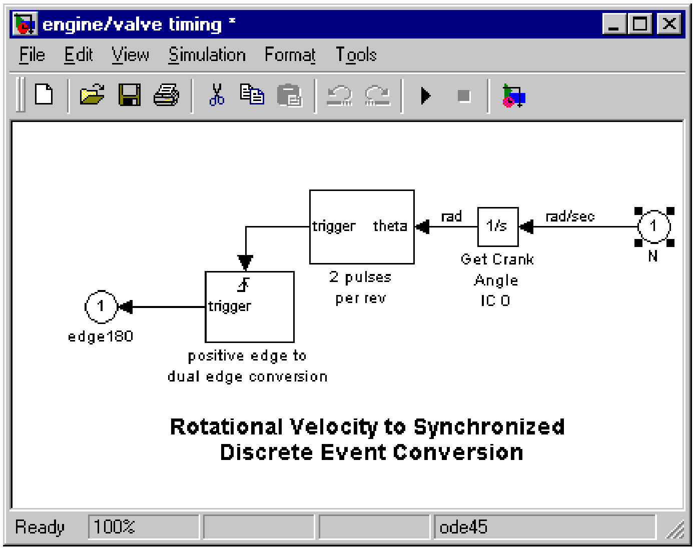
2 From the **View** menu, select **View Block**.

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**.


engine/valve timing *
File Edit View Simulation Format Tools
trigger
theta
rad
1/s
rad/sec
1
N
2 pulses per rev
Get Crank Angle IC 0
1
edge180
trigger
positive edge to dual edge conversion
Rotational Velocity to Synchronized Discrete Event Conversion
Ready
100%
ode45

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.

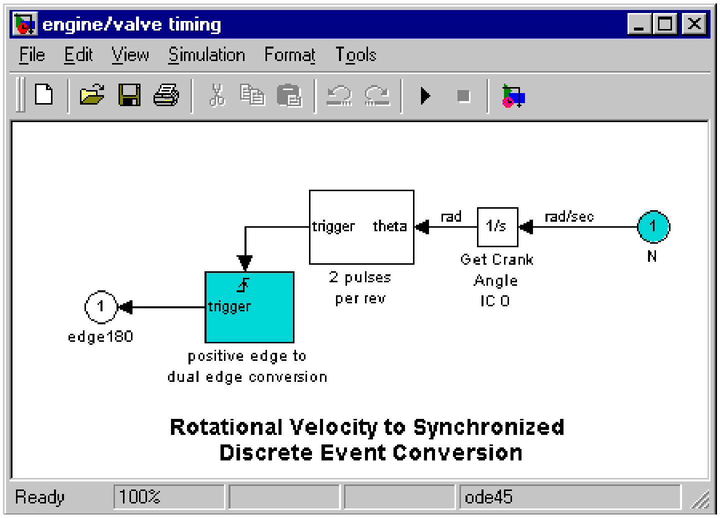
Note Although the **MATLAB** menu and **Select item** feature appear in all DOORS formal modules, you can only use them in a surrogate exported module.

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.



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.

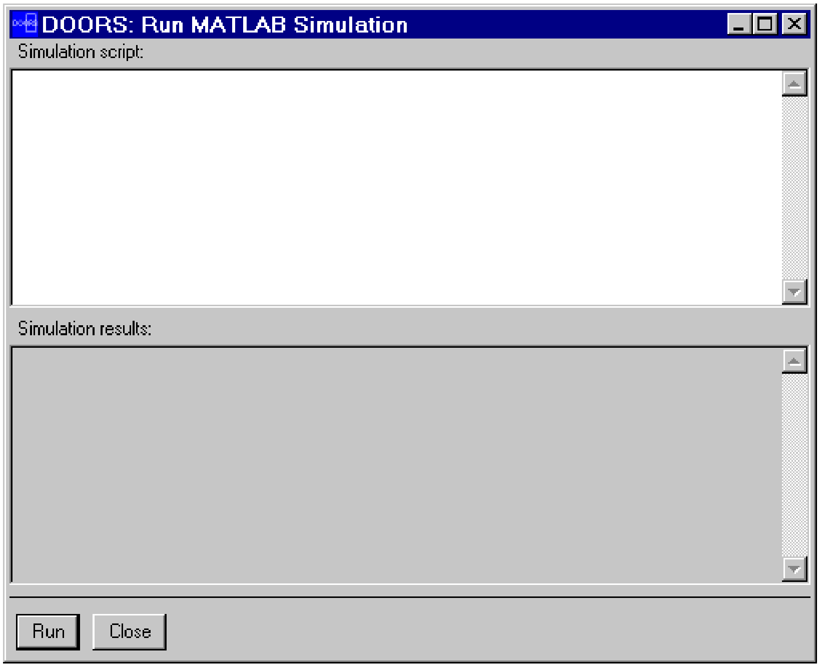
Note When you save a Simulink model, it is saved with the highlighting. You *cannot* later turn off the highlighting without opening the model again in the Navigator. Therefore, unless you want the highlighting to become a “permanent” part of the model file, turn off highlighting using the Navigator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

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:

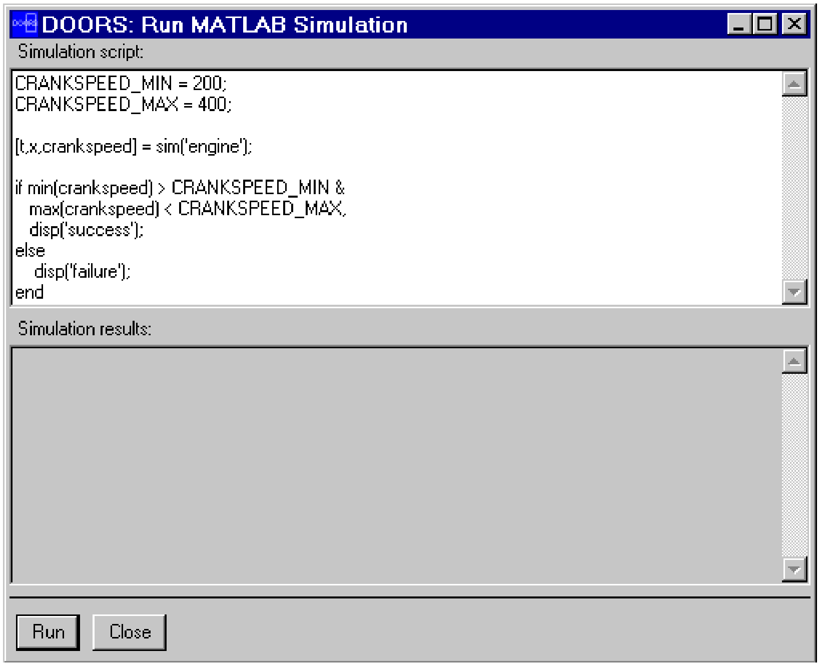
1 From the **MATLAB** menu in DOORS, select **Run script**.

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.



```
CRANKSPEED_MIN = 200;
CRANKSPEED_MAX = 400;

[t,x,crankspeed] = sim('engine');

if min(crankspeed) > CRANKSPEED_MIN &
   max(crankspeed) < CRANKSPEED_MAX,
   disp('success');
else
    disp('failure');
end
```

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.

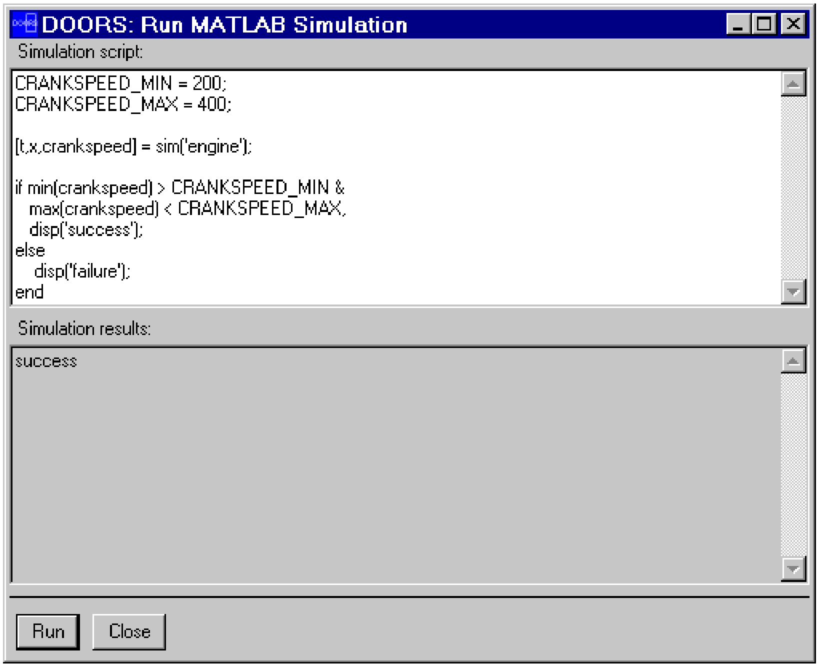
c Tests the minimum and maximum values in the vector `crankspeed` against the desired minimum and maximum values.

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.



Simulation script:

```
CRANKSPEED_MIN = 200;
CRANKSPEED_MAX = 400;

[t,x,crankspeed] = sim('engine');

if min(crankspeed) > CRANKSPEED_MIN &
  max(crankspeed) < CRANKSPEED_MAX,
  disp('success');
else
   disp('failure');
end
```

Simulation results:

```
success
```

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.

4

Function 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 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.

Index