**General**

* IMPORTANT: Make sure requirement settings are consistent across all users before establishing links between models and requirement documents.
* To control the requirement settings, select *Analysis > Requirements Traceability > Settings* to open the Requirement Settings dialog.
* In the *Set Default storage mode for traceability data* panel under the *Storage* tab, select *Store externally (in a separate \*.slmx file)*.

NOTE: When starting from a model that already has requirements links stored internally, you must first clear all requirement links and then copy the contents into a new slx file. Otherwise the above setting will not be applied.

* In the *Duplicating outgoing links when copying Simulink and Stateflow objects* panel under the *Storage* tab, select *Duplicate links only when model requirements are highlighted*.
* In the *Linking to the active selection within an external document* panel under the *Selection Linking* tab, set *Enabled applications* by enabling *Word*, *Excel,* and *DOORS*.
* In the *Linking to the active selection within an external document* panel under the *Selection Linking* tab, set *Document file reference* by selecting *filename only (on MATLAB path)*.
* In the *Linking to the active selection within an external document* panel under the *Selection Linking* tab, set *Apply this user tag to new links* by leaving it empty.
* In the *When creating selection-based linking* panel under the *Selection Linking* tab, clear *Modify destination for bidirectional linking*.
* Under the *Selection Linking* tab, make sure *Enable external connectivity at MATLAB startup* is enabled. Otherwise, enable the option, and then restart MATLAB.
* In the *Keyword filters* panel under the *Filters* tab, clear *Filter links by keywords when highlighting and reporting requirements*.
* In the *Link type filters* panel under the *Filters* tab, clear *Disable synchronization item links*.
* In the *External links in generated report* panel under the *Report* tab, enable *Use internal HTTP server to support navigation from system browsers*.
* Use the following guidelines to determine where should you establish requirement links in high-level and mid-level models:
  + Consider partitioning your models so that you can map each Simulink subsystem, Stateflow subchart (or superstate), or MATLAB Function block to a reasonable set of requirements.
  + Establish requirement links at the lowest level component of model elements. Exemptions are model elements that do not impact the model's behavior or the generated code.
  + To reduce the number of requirement links you must maintain for a model, establish requirement links at the level of a component that contains a group of model elements. In Simulink, this level can be the top-level diagram of a model, a subsystem, or a MATLAB function. In Stateflow, this level can be a chart, a state, a box, a Simulink function, or a graphical function. Note that requirement links for virtual subsystems are not preserved in the generated code due to block elimination optimization. However, the information is captured in a traceability report that is available upon code generation.
  + Use the following guidelines to determine where should you establish requirement links in low-level reusable models or user-defined library blocks:
  + If requirements exist for a reusable utility itself (not the instance where the utility is used), establish requirement links at the root-level of the reusable model or library block subsystem. Note that these requirement links are established inside the reusable model or library block subsystem, not on the Model or Subsystem block. You should also insert a System Requirements block at the root-level of the reusable model or library block subsystem to display the requirements.
  + If requirements exist for an instance of a reusable model or user-defined library block, establish requirement links based on the above guidelines for high-level and mid-level models.

**Importing Requirements from Third-Party Tools**

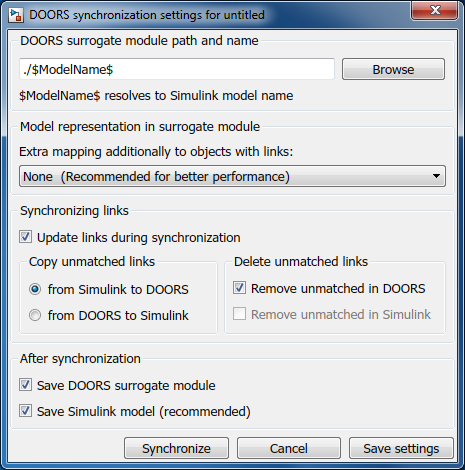
* If your software requirements are authored in a third-party tool, consider importing the requirement data into Requirements Toolbox Editor.
* Key benefits for importing your software requirements into Requirements Toolbox Editor are:
* Traceability between requirements and models/tests is inherently bidirectional.
* Monitoring of the implementation/verification of each functional requirement is fully integrated.
* Scoping of model coverage based on requirement links eliminates incidental coverage.
* If your software requirements will continue to be maintained in the third-party tool, remember to enable *Allow updates from external source* when performing the import for the first time. This option allows you to update the imported requirements with changes to the source document when necessary.

**Linking to Word**

* Do not establish bidirectional links to enable traceability between Simulink and Word in both directions. Bidirectional links insert navigation objects in the original requirement documents, turning the documents dirty.
* Before establishing any requirement links from a model to Word documents, you should first insert bookmarks in the Word documents where you want links to trace. With the bookmarks in place, the Word documents will not become dirty when you establish requirement links from Simulink.

**Linking to DOORS**

* Do not establish bidirectional links to enable traceability between Simulink and DOORS in both directions. Bidirectional links insert navigation objects in the original requirement modules, turning the modules dirty. Generate surrogate modules in DOORS to provide a mechanism for bidirectional traceability instead.
* After establishing requirement links from a model to DOORS modules, select *Analysis > Requirements Traceability > Synchronize with DOORS* to create a surrogate module in DOORS.
* Use the following settings when performing synchronization with DOORS:



* Any changes to a requirement should be handled using the following steps:
  1. Make changes to the original requirement module in DOORS.
  2. Make changes to the model in Simulink.
  3. Replace obsolete requirement links with new links to the modified requirement module.
  4. Resynchronize the model with DOORS.
  5. Clean up the updated surrogate module by removing flagged obsolete objects.