Data Store Rescope Tool

September 2019

McMaster Centre for Software Certification (McSCert)

1 Introduction

The Data Store Rescope Tool is used to properly scope Data Store Memory blocks in Simulink models. The tool aims to assist software development actives, as well as software refactoring and maintenance.

Using Data Store Memory blocks higher in the model hierarchy than needed is bad software engineering practice, and is similar to using global variables in programming. This can result in inadvertent/unwanted access to the memory, and also clutters the interface of subsystems by showing low-level details. The Data Store Rescope Tool is used on Data Store Memory blocks in order to push them down to the lowest layer possible, while still in scope of their corresponding Data Store Read and Data Store Write blocks in the model.

This tool can also repair the scope of a Data Store Memory block when it is positioned too low in the hierarchy, such that its Data Store Read or Data Store Write blocks are out of its scope. In this case, the Data Store Memory block is repositioned higher in the hierarchy.

More Information

For more information on the tool and how it can be used in a model-based development with Simulink, please refer to the following papers:

Vera Pantelic, Steven Postma, Mark Lawford, Monika Jaskolka, Bennett Mackenzie, Alexandre Korobkine, Marc Bender, Jeff Ong, Gordon Marks, Alan Wassyng, "Software engineering practices and Simulink: bridging the gap," *International Journal on Software Tools for Technology Transfer (STTT)*, 2017, 1-23.

Vera Pantelic, Steven Postma, Mark Lawford, Alexandre Korobkine, Bennett Mackenzie, Jeff Ong, and Marc Bender, "A Toolset for Simulink: Improving Software Engineering Practices in Development with Simulink," *Proceedings of 3rd International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2015)*, SCITEPRESS, 2015, 50–61.

2 How to Use the Tool

This section describes what must be done to setup the Data Store Rescope Tool, as well as how to use the tool.

2.1 Prerequisites and Installation

1. Use MATLAB/Simulink 2011b or newer.

2. To install the tool,

   (a) from a `.zip` file — unzip the contents into your desired location. Ensure the unzipped folder and subfolders are present in your MATLAB search path, or add them if they are not present. Run sl_refresh_customizations to refresh the Context Menu.

   (b) from a `.mltbx` file — simply open MATLAB and double-click on the file. Your MATLAB search path should be automatically configured.

   (c) from the files only — add the folders and subfolders to your MATLAB search path. Run sl_refresh_customizations to refresh the Context Menu.

   - *Note:* If running the command "`which dataStoreRescope`" indicates that the script is not found, then the tool needs to be added to the MATLAB search path. For information on adding files to the MATLAB search path, please see the MathWorks documentation.

3. Ensure your model is open and unlocked.

2.2 Getting Started

The Data Store Rescope Tool can be used via the Simulink Context Menu, which can be viewed by right-clicking in a model. The following options can be available in the Data Store Rescope Tool menu, depending on what is selected in the model. These options are shown in Figure 1.

- *Rescope All* — Available at all times.

- *Rescope Selected* — Available when one or more Data Store Memory, Data Store Read, or Data Store Write blocks are selected.

- *Rescope Non-Selected* — Available when one or more Data Store Memory, Data Store Read, or Data Store Write blocks are selected.
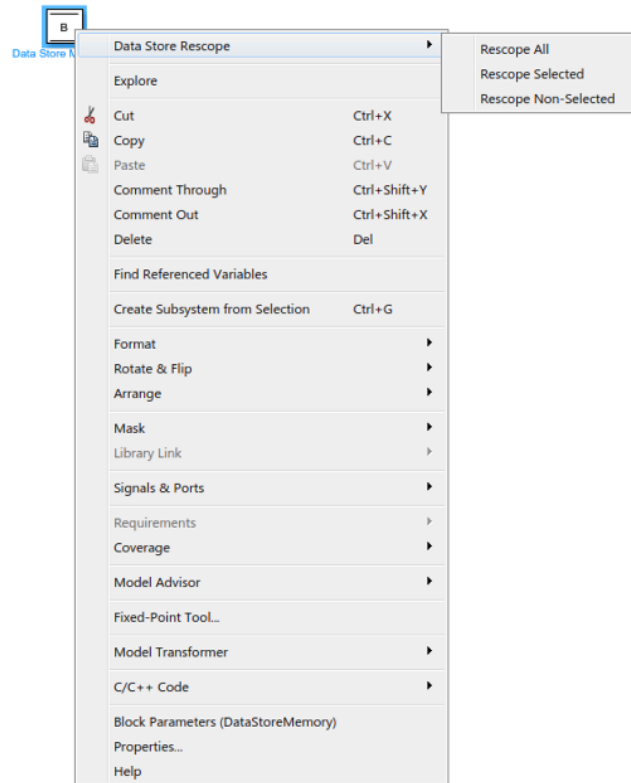
Figure 1: Simulink Context Menu with tool options visible.

2.3 Functionality

This section describes the tool functionality when being used from the Simulink Context Menu (Figure 1).

Rescope All

Right-clicking anywhere in the model and selecting `Rescope All` will move up/down all Data Store Memory blocks starting from the root level of the model. They are each moved to their lowest level of the hierarchy such that all references to them via Data Store Read/Write blocks are still within scope.

Rescope Selected

Selecting one or more Data Store Memory blocks, and then selecting `Rescope Selected` will move up/down the selected Data Store Memory blocks only. They are each moved to their lowest level of the hierarchy such that all references to

them via Data Store Read/Write blocks are still within scope. All other non-selected Data Store Memory blocks remain in their original location.

Rescope Non-Selected

Selecting one or more Data Store Memory blocks, and then selecting `Rescope Non-Selected` will move up/down all Data Store Memory blocks, except those which are selected. Those not selected are moved to their lowest level of the hierarchy such that all references to them via Data Store Read/Write blocks are within scope.

2.4   Configuration Parameters

The configuration file `config.txt` is included in `DataStoreRescope\src`. The following configuration parameters are utilized by the tool, and can be modified by the user in order to tailor tool functionality:

- `linkedBlocksEnabled` — Enables or disables the ability to rescope Data Store Memory blocks inside linked subsystems.

Please see the configuration file for more details regarding parameter usage and accepted values. These parameters can be modified with MATLAB open, and do not require that MATLAB be restarted for the changes to take effect.

2.5   Logs

When invoking any of the tool options, a log text file is created to summarize the changes in the model. This text file will be created in the same directory as the model on which the operation was performed, and will be named `modelName_RescopeLog.txt`. If a previous log file already exists in this location, subsequent log content will be appended to the file. The log file includes the following information:

- Data and time the rescope operation was performed

- Total number of Data Store Memory blocks in the model

- Total number of Data Store Memory blocks rescoped

- Percentage of Data Store Memory blocks rescoped

- A list of all Data Store Memory blocks rescoped, including:

  - Block name
  - Initial location in the model
  - New location in the model

2.6 Errors and Warnings

Any errors or warnings during tool use will be visible in the MATLAB Command Window. Errors will be shown when the model is locked or function parameters are incorrect.

Warnings will be given if the operation has completed successfully, however the user should be made aware of a situation which may or may not need further attention. For example, at different levels of the hierarchy, Data Store Memory blocks can have the same name. Given that the default names are "Data Store Memory", "Data Store Memory1", "Data Store Memory2", etc., and if the user does not change them, it may be the case that multiple Data Store Memory blocks, located in different subsystems, have the same name. As a result, when rescoping is performed, two or more Data Store Memory blocks of the same name may be placed in the same subsystem, resulting in an error. Therefore, the conflicting blocks will be renamed, following the usual Simulink convention described earlier. The user will be notified of this via a warning.

3 Example

Use the command `DataStoreRescopeDemo` in the Simulink Command Window to open the example model, shown[1] in Figure 2. There are three Data Store Memory blocks in this example: B and C at the root level, and A at the second level of the hierarchy. These three Data Store Memory blocks are used throughout the models in various locations:

- Although they are in the root level, B and C are used lower in the hierarchy at levels 3 and 4, respectively. Therefore, Data Store Memory B and C are located higher than they need to be in the model, and can be moved downwards.

- Data Store Write A is at the root level, however, Data Store Memory A is located below it in the hierarchy, at level 2. This will result in an error upon simulation. To repair this, Data Store Memory A should be placed at the root level, so that it is within the scope of Data Store Write A.

To to rescope all possible Data Store Memory blocks correctly, right-click anywhere in the model and select the `Rescope All` option. The resulting model is shown[1] in Figure 3. Data Store Memory blocks B and C were moved down in the hierarchy, while A was moved up to the root.

It is also possible to right-click on each of the Data Store Memory blocks (or their associated Data Store Read/Write blocks) individually, and select the `Rescope Selected` option to achieve the same result.

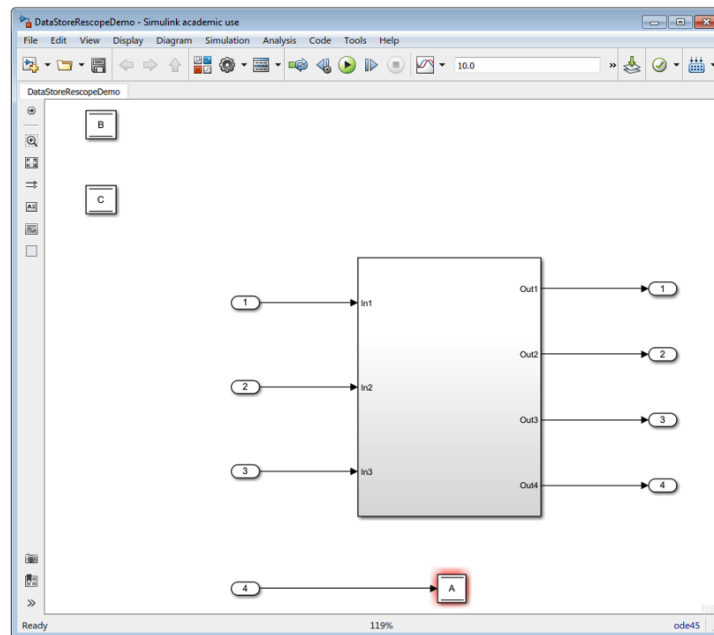[1]As there are 5 hierarchical levels, we do not show them all here.

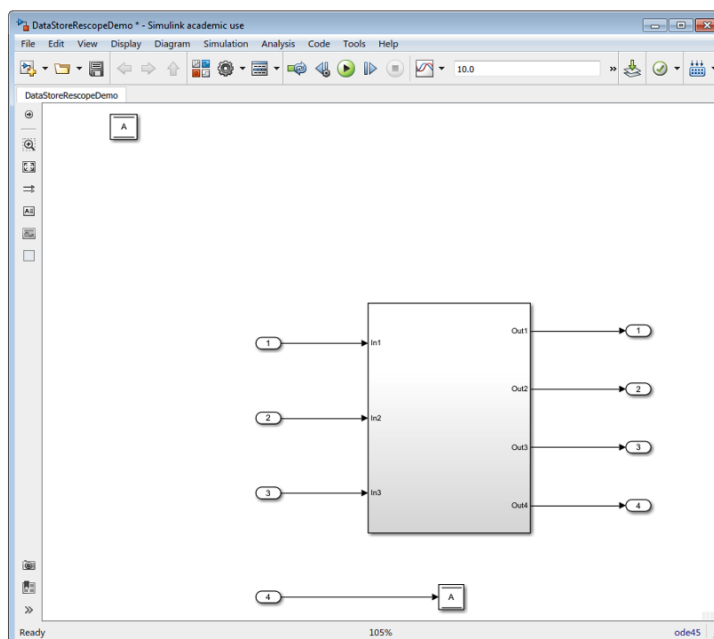Figure 2: Data Store Rescope Tool demo model.

Figure 3: Resulting model after `Rescope All` operation.

4 Matlab Commands

The tool can also be used via the MATLAB command line, with the following functions.

Function	`dataStoreRescope`
Syntax	`dataStoreRescope`(*model, dontMove*)
Description	Rescopes all Data Store Memory blocks except for those listed in *dontMove*.
Inputs	*model*: The Simulink model name (or top-level system name) of the model on which to perform the rescoping operation. *dontMove*: A cell array of Data Store Memory pathnames which are not to be moved.
Outputs	N/A

Example: The following command moves all Data Store Memory blocks in the open/loaded model `DataStoreRescopeDemo` to their proper level in the hierarchy such that Data Store Read/Write references remain within scope. The resulting model is shown as Figure 3.

```
dataStoreRescope('DataStoreRescopeDemo', {})
```

Function	`rescopeSelected`
Syntax	`rescopeSelected`(*model, dataStores*)
Description	Rescopes only specified Data Store Memory blocks.
Inputs	*model*: The Simulink model name (or top-level system name) of the model on which to perform the rescoping operation. *dataStores*: A cell array of Data Store Memory pathnames which are to be moved.
Outputs	N/A

Example: The following command moves Data Store Memory blocks A, B, and C in the open/loaded model `DataStoreRescopeDemo` to their proper level in the hierarchy such that Data Store Read/Write references remain within scope. The resulting model is shown as Figure 3.

```
rescopeSelected('DataStoreRescopeDemo', {'DataStoreRescopeDemo/Data
Store Memory1', 'DataStoreRescopeDemo/Data Store Memory2',
'DataStoreRescopeDemo/Subsystem/Data Store Memory'})
```

Note: *Simulink block names, such as for Subsystems and Data Store Memory/Read/Write, can contain newline characters. When using the command line functions, please ensure you include them in the pathnames of the dontMove and dataStores arguments.*