

FAQ 019

# Missing Output in the Variable Browser

## Keywords

missing block output variables; missing signal labels; ControlDesk Variable Browser; referenced models; variable description file; TRC file; <modelName>.trc; inline parameter; virtual blocks; signal storage reuse; optimization

## Question

I have built and downloaded my model without any problems and want to trace some variables with ControlDesk. However, in the ControlDesk Variable Browser, some block output variables, parameters or signal labels seem to be missing. What is the reason for this?

## Relevant Products

Real-Time Interface (RTI) and ConfigurationDesk; differences between Real-Time Interface (RTI) and ConfigurationDesk are explicitly pointed out.

## Solution

The Variable Browser in ControlDesk displays the contents of the variable description file (TRC file, <modelName>.trc) generated by the build process. If signals or parameters are missing, in most cases the TRC file does not contain the information.

The following known issues with TRC file generation lead to missing information:

- Known issues affecting the model globally
- Known issues affecting certain block types or blocks
- Issues based on a specific kind of modeling
  - Model referencing
  - Structured parameters
  - Simulink buses
- Known issues that have occurred since the introduction of TRC file enhancement with MATLAB R2015b and dSPACE Release 2015-B (TRC 3.0)
- Known issues relevant for outdated dSPACE Releases


The following sections provide solutions to the above-mentioned issues.

## Known Issues Affecting the Model Globally

The following issues are known to affect the model globally:

- In Simulink®, open the **Configuration Parameters** dialog<sup>1</sup> and go to **Optimization – Signals and Parameters**. Select the **Signal storage reuse** checkbox.

If you select the 'Signal storage reuse' checkbox, Simulink Coder®<sup>2</sup> does not use a separate variable for each block output but assigns a single buffer to multiple block outputs. This means that, in general, the signals are not available in the TRC file unless you make specific settings for them.

	A warning is displayed during the build process if the 'Signal storage reuse' option is enabled.
---	--

To make sure that an entry for a block output signal is generated into the TRC file even if the 'Signal storage reuse' option is enabled, you can use one of the following methods:

- Enable the 'Test point' option for the signal.
- Specify a label and a global storage class for the signal.
- Specify a label for the signal and create an appropriate Simulink.Signal object with the `ExportedGlobal`, `ImportedExtern`, `ImportedExternPointer`, or `SimulinkGlobal` storage class in the MATLAB workspace (for details, refer to Rules for Simulink.Signal and Simulink.Parameter objects). In addition, you must link the signal to the Simulink.Signal object. For this, you do one of the following:
  - Select the 'Signal must resolve to Simulink.Signal object' checkbox in the signal's Signal Properties dialog.
  - Set the 'Signal resolution' option to 'Explicit and implicit' on the Data Validity page of the Diagnostics dialog.
- In Simulink, open the **Configuration Parameters** dialog<sup>1</sup> and go to **Optimization – Signals and Parameters**. Select the **Default parameter behavior** checkbox (up to MATLAB R2015a: Inline parameters).

If you set the 'Default parameter behavior' option to 'Inlined' (up to MATLAB R2015a: Inline parameters = ON), the numerical values of block parameters are used in the generated code. In this case, Simulink does not generate any global variables for the block parameters, unless you specify them as tunable parameters or create Simulink.Parameter objects with a specific storage class (`ExportedGlobal`, `ImportedExtern`, `ImportedExternPointer`, or `SimulinkGlobal` storage) for them.

- In Simulink, open the **Configuration Parameters** dialog and go to **Code Generation – RTI variable description file options** (for RTI)<sup>3</sup>/**DSRT variable description file options** (for ConfigurationDesk). Select the **Include only Simulink.Parameter and Simulink.Signal objects with global storage class** checkbox.

If this option is enabled, only parameters and signals referencing a Simulink.Parameter or a Simulink.Signal object in the MATLAB workspace are included in the TRC file. If this option is enabled, the TRC file provides only the Labels group and the Tunable Parameters group. It does not contain the entire model hierarchy.

The dependencies between the above-mentioned Configuration Parameters settings (Code options settings) and the model hierarchy in the variable description file's Model Root group are summarized [here](#).

- In Simulink, open the **Configuration Parameters** dialog<sup>1</sup> and go to Optimization – Signals and Parameters. Select the **Enable local block outputs** checkbox.

If this option is enabled, output variables might be created as local function variables and not as global variables.

- In Simulink, open the **Configuration Parameters** dialog<sup>1</sup> and go to Optimization – Signals and Parameters. Select the **Inline invariant signals** checkbox.

If this option is enabled, constants and signals that depend only on constants are not available as variables in the generated code

- In Simulink, open the **Configuration Parameters** dialog and go to **Code Generation – RTI variable description file options** (for RTI)<sup>3</sup>/**DSRT variable description file options** (for ConfigurationDesk). Select the **Include signal labels** and **Include virtual blocks** checkboxes.

If this option is disabled, the signal labels of the model are not generated into the TRC file and virtual blocks are not available in the TRC file.

## Known Issues Affecting Certain Block Types or Blocks

The following issues are known to affect certain block types or blocks:

- Check whether the I/O block whose signal you want to trace (RTI block or Model Port block) is **not connected to any block** or **only to virtual blocks** (Ground, Terminator, Input, Output, Goto, From, Mux Demux, etc.).

If an I/O block (RTI block or Model Port block) is not connected or connected only to virtual blocks, Simulink Coder<sup>2</sup> marks the I/O block as unconnected. For performance reasons, real-time code is optimized. No code is generated for the I/O channel and either the entry for the block port is not generated or the port is marked as **No data (unused channel)** in the variable browser.

In this case, include at least one non-virtual block in the signal line, for example, a unity Gain block (Gain value set to 1) or, for Boolean signals, a Simulink Logical Operator whose operator parameter is set to 'OR' and whose second input port is set to a constant Boolean 0.

The **Hardware Interrupt, Software Interrupt, Data Capture, Background, IPI** and **Runnable Function** blocks (formerly **Trigger Event Port** block) are completely virtual and offer no signal to be traced. Therefore, such blocks do not appear in the TRC file.

- In some cases, Simulink Coder<sup>2</sup> does not generate separate variables for the block output signals of **Constant** blocks.
  - Instead of the block output signals, you can also observe the Constant block parameters within ControlDesk. In the Variable Browser, locate the Constant block you want to observe and drag its parameter to the desired instrument.
  - To make sure that Simulink Coder generates a separate variable for an output signal of a Constant block, you can declare the signal to be a test point. To declare a test point, use the Simulink **Signal Properties** dialog. For further information on test points, refer to *Simulink Coder User's Guide* by MathWorks®.

This also applies to the outputs of virtual blocks that are connected to Constant blocks.

- Check whether the following TRC file feature is set in the affected subsystem or in one of the subsystems located above the affected subsystem. Normally, this feature is used to either hide the contents of a subsystem in the TRC file or reduce the size of the TRC file:
  - RTI: An activated **TRC Exclusion** block (Block Display: TRC Exclusion <activated>) is part of the model  
OR  
In the **Configuration Parameters** dialog, under **Code Generation – RTI variable description file**<sup>3</sup>, the **Apply subsystem omission tags** option is enabled and a subsystem's DsVdOmit tag is set to 1 or -1.
  - ConfigurationDesk: In the **Configuration Parameters** dialog, under **Code Generation – DSRT variable description file**, the **Apply subsystem omission tags** option is enabled and a subsystem's DsVdOmit tag is set to 1 or -1.

- ConfigurationDesk: Entries for blocks displaying sinks, such as the Simulink Display block or the Data Output block of the dSPACE Model Port Library, are not included in the TRC file.

As of MATLAB R2015b, input signals of signal sink blocks are generated into the TRC file.

- ConfigurationDesk: If you use Simulink Inport and Outport blocks as interface blocks for your model instead of the Data Inport and Data Output blocks of the Model Port Block library, the following limitation applies:
  - Outport blocks that retrieve a **vectorized signal** from a virtual block are not supported.

- **Signal generators and viewers** defined with the Signal & Scope Manager are not generated into the TRC file.
- If the block's output data type is specified by **Inherit: Inherit via internal rule**, the data type propagation for some Simulink blocks can lead to an unexpected output data type (fixed-point data type, e.g., ufix8\_En7).  
If a 64-bit fixed-point data type (e.g., ufix64\_En30) is chosen automatically, the signal is missing from the TRC file.  
To avoid this behavior, you have to specify **Inherit: Inherit via back propagation** or **Inherit: Same as input** as the output data type in the block.

## Issues Based on a Specific Kind of Modeling

- When using **referenced models**, you have to consider several aspects regarding the inclusion of signals and parameters in the TRC file.
  - If the **Total number of instances per top model** option on the **Model Referencing** page of the **Configuration Parameters** dialog is set to 'Multiple', the contents of the relevant referenced models do not appear in the TRC file.
  - Variables of **Inport** blocks located on the root level of a referenced model are never generated into the TRC file. This also applies to the outputs of virtual blocks (Mux, Demux, Goto, From blocks, subsystems) that are directly connected to such an Inport block.
  - Block outputs of blocks that drive an **Outport** block located on the root level of a referenced model might not be generated. To solve this problem, you can set a test point for the relevant signal.
  - The entries for global block parameters of referenced models depend on the MATLAB version you work with. If you work with MATLAB R2015a and earlier, the following applies: You can access tunable block parameters of referenced models if they are resolved by Simulink.Parameter objects defined in the MATLAB workspace. Their storage class must be either SimulinkGlobal, ExportedGlobal, ImportedExtern, or ImportedExternPointer. These parameters appear below the Tunable Parameters group in TRC files. For each referenced model (with 'The total number of instances' set to 'One') there is a separate Tunable Parameters group. This group contains the tunable parameters of the model, including all the tunable parameters of lower-level referenced models. As a result, the Tunable Parameters group of the top-level model also contains all the tunable parameters of all referenced models, provided that the 'Inline parameters' option is enabled for the top-level model.
- The tunable parameters can be structures containing numerical values. If MATLAB workspace variables and **Simulink.Parameter** objects provide a data structure as a value, they can be configured as tunable parameter structures. However, tunable parameter structures are not generated into the TRC file.

As of MATLAB R2015b, structured parameters are available in the TRC file.

- The **Constant** block supports a bus object as data type to use structured parameters as Constant values. Structured parameters are not generated into the TRC file.

As of MATLAB R2015b, structured parameters are available in the TRC file.

- You can use arrays of buses to represent structured data. A Simulink bus itself can also contain bus arrays. This feature is supported, but no entries are generated into the TRC file for the elements of a bus array.

As of MATLAB R2015b, bus arrays are available in the TRC file.

## Known Issues That Have Occurred Since the Introduction of TRC File Enhancement with MATLAB R2015b and dSPACE Release 2015-B (TRC 3.0)

The following issues have occurred since the introduction of TRC file enhancement:

- Entries for virtual Simulink buses and muxed signals (e.g., represented as `Out1{SubArray1}`), are no longer generated.
- A TRC file no longer contains entries for the output signals of a Signal Selector block connected to a non-virtual bus and for other virtual blocks connected to these signals.
- The **Data Stores** group is no longer generated into the TRC file.
- Structured parameters (structs) are *not* generated into the TRC file if one of the following conditions is met:
  - a. A non-numerical value (e.g., string) is part of the structured parameter.
  - b. A reserved TRC syntax keyword is used as a field name, e.g., `param`, `init`, `outputs`.
- Each referenced model of a model hierarchy had its own **Tunable Parameters** group. These groups are no longer generated. All global parameters referenced in the top-level model or in the referenced models, are generated into the **Tunable Parameters** group of the top-level model.
- The entries for global block parameters of referenced models depend on the MATLAB version you work with. If you work with MATLAB R2015b and later, the following applies: If the 'Default parameter behavior' option of the top model is set to 'Tunable' (formerly: Inline parameters = OFF), all the parameters of referenced models are also tunable. Thus, they are available as parameters either in the Tunable Parameters group or as block parameters. If the 'Default parameter behavior' option of the top model is set to 'Inlined' (formerly: Inline parameters = ON), you can access the global block parameters of referenced models if they are resolved by Simulink.Parameter objects defined in the MATLAB workspace. Their storage class must be either SimulinkGlobal, ExportedGlobal, ImportedExtern, or ImportedExternPointer. These parameters appear in the Tunable Parameters group of the top-level model in TRC files. This group contains the tunable parameters of the entire model referencing hierarchy.
- Look-up Table blocks with a dimension greater than 2, such as 4x3x2 matrices, are no longer automatically divided into two-dimensional slices.
- If a block's parameter definition contains an expression, the local block parameter is no longer available.

## Known Issues Relevant for Outdated dSPACE Releases

The following issues are relevant for outdated dSPACE Releases:

- RTI: In former Releases, the code of the Signal Generator block was extended. With this extension the **WaveForm** parameter was available in the TRC file, switching the Signal Generator's wave form during simulation. As of MATLAB R2010a, Real-Time Workshop changed the code generation of this block. As a result, the **WaveForm** parameter cannot be provided any longer in the TRC file.
- RTI: Up to dSPACE Release 7.0, signals and parameters with enumerated data types are not generated into the TRC file.
- Up to dSPACE Release 7.3: If you start a build process using MATLAB R2009b or newer, the generated TRC file does not contain entries for variables that are connected to a Simulink Outport block on the root level of the model. This applies to direct signal connections and to indirect connections via virtual blocks. The following entries are missing in the generated TRC file:
  - Signal labels for signals that are connected to Simulink Outport blocks on the model root level
  - Variables for output signals of virtual blocks that are connected directly or indirectly (via virtual blocks) to a Simulink Outport block on the model root level

To make the variables available in the TRC file, enable the 'Test point' option for the affected signals in the model.

- Up to dSPACE Release 7.3: Due to an optimization of Simulink Coder<sup>1</sup>, the output ports of Bus Selector blocks might not be written to the TRC file. If the Bus Selector blocks' outputs do not seem to be necessary for calculating another non-virtual block, no code is generated for the block. This is the case if the following conditions apply:
  - The Bus Selector block is fed by a Signal Conversion block.
  - The output of the Bus Selector block is not used in any non-virtual block.

Select the 'Test point' checkbox in the Signal Properties dialog of the Bus Selector block's output signal to have code generated for the Bus Selector block.

If you detect missing output in the Variable Browser and none of the documented reasons apply to you, send the Simulink model together with the version information of the active MATLAB Release and the active dSPACE Release and a short description about the missing output to dSPACE Support for further analysis.



## Tables

Using MATLAB R2014a and earlier:

Code Options Settings	Model Hierarchy in the Model Root Group
Inline Parameters = ON Signal Storage Reuse = ON Include only Simulink.Parameter and Simulink.Signal objects with global storage class = OFF	<ul style="list-style-type: none"> <li>Signals specified as test points and signals with a global storage class are available</li> <li>No parameters are available</li> </ul>
Inline Parameters = ON Signal Storage Reuse = ON Include only Simulink.Parameter and Simulink.Signal objects with global storage class = ON	<ul style="list-style-type: none"> <li>The model hierarchy is not available</li> <li>No LookupTable Data entries are available</li> </ul>
Inline Parameters = OFF Signal Storage Reuse = ON Include only Simulink.Parameter and Simulink.Signal objects with global storage class = OFF	<ul style="list-style-type: none"> <li>Signals specified as test points and signals with a global storage class are available</li> <li>Block parameters are available</li> </ul>
Inline Parameters = OFF Signal Storage Reuse = ON Include only Simulink.Parameter and Simulink.Signal objects with global storage class = ON	<ul style="list-style-type: none"> <li>The model hierarchy is not available</li> <li>No LookupTable Data entries are available</li> </ul>
Inline Parameters = ON Signal Storage Reuse = OFF Include only Simulink.Parameter and Simulink.Signal objects with global storage class = OFF	<ul style="list-style-type: none"> <li>All block output signals are available</li> <li>No block parameters are available</li> </ul>
Inline Parameters = ON Signal Storage Reuse = OFF Include only Simulink.Parameter and Simulink.Signal objects with global storage class = ON	<ul style="list-style-type: none"> <li>The model hierarchy is not available</li> <li>No LookupTable Data entries are available</li> </ul>
Inline Parameters = OFF Signal Storage Reuse = OFF Include only Simulink.Parameter and Simulink.Signal objects with global storage class = OFF	All block output signals and block parameters are available
Inline Parameters = OFF Signal Storage Reuse = OFF Include only Simulink.Parameter and Simulink.Signal objects with global storage class = ON	<ul style="list-style-type: none"> <li>The model hierarchy is not available</li> <li>No LookupTable Data entries are available</li> </ul>

## Using MATLAB R2014b or R2015a

Code Options Settings	Model Hierarchy in the Model Root Group
Inline Parameters = ON Signal storage reuse = ON Include only Simulink.Parameter and Simulink.Signal objects with global storage class = OFF	<ul style="list-style-type: none"> <li>• Signals specified as test points and signals with a global storage class are available.</li> <li>• No parameters are available.</li> </ul>
Inline Parameters = ON Signal storage reuse = ON Include only Simulink.Parameter and Simulink.Signal objects with global storage class = ON	<ul style="list-style-type: none"> <li>• The model hierarchy is not available.</li> <li>• No look-up table data entries are available</li> </ul>
Inline Parameters = OFF Signal storage reuse = ON Include only Simulink.Parameter and Simulink.Signal objects with global storage class = OFF	<ul style="list-style-type: none"> <li>• Signals specified as test points and signals with a global storage class are available.</li> <li>• Block parameters are available.</li> </ul>
Inline Parameters = OFF Signal storage reuse = ON Include only Simulink.Parameter and Simulink.Signal objects with global storage class = ON	<ul style="list-style-type: none"> <li>• The model hierarchy is not available.</li> <li>• No look-up table data entries are available.</li> </ul>
Inline Parameters = ON Signal storage reuse = OFF Include only Simulink.Parameter and Simulink.Signal objects with global storage class = OFF	<ul style="list-style-type: none"> <li>• All block outputs are available<sup>1)</sup>.</li> <li>• No block parameters are available.</li> </ul>
Inline Parameters = ON Signal storage reuse = OFF Include only Simulink.Parameter and Simulink.Signal objects with global storage class = ON	<ul style="list-style-type: none"> <li>• The model hierarchy is not available.</li> <li>• No look-up table data entries are available.</li> </ul>
Inline Parameters = OFF Signal storage reuse = OFF Include only Simulink.Parameter and Simulink.Signal objects with global storage class = OFF	<ul style="list-style-type: none"> <li>• All block output signals and block parameters are available<sup>1)</sup>.</li> </ul>
Inline Parameters = OFF Signal storage reuse = OFF Include only Simulink.Parameter and Simulink.Signal objects with global storage class = ON	<ul style="list-style-type: none"> <li>• The model hierarchy is not available.</li> <li>• No look-up table data entries are available.</li> </ul>

<sup>1)</sup> In individual cases, a signal might be unavailable if the 'Signal storage reuse' checkbox is cleared.

## Using MATLAB R2015b

Code Options Settings	Model Hierarchy in the Model Root Group
Inline Parameters = ON Signal storage reuse = ON Include only Simulink.Parameter and Simulink.Signal objects with global storage class = OFF	<ul style="list-style-type: none"> <li>Signals specified as test points and signals with a global storage class are available.</li> <li>References to the parameters in the Tunable Parameters group are available.</li> </ul>
Inline Parameters = ON Signal storage reuse = ON Include only Simulink.Parameter and Simulink.Signal objects with global storage class = ON	<ul style="list-style-type: none"> <li>The model hierarchy is not available.</li> <li>No look-up table data entries are available.</li> </ul>
Inline Parameters = OFF Signal storage reuse = ON Include only Simulink.Parameter and Simulink.Signal objects with global storage class = OFF	<ul style="list-style-type: none"> <li>Signals specified as test points and signals with a global storage class are available.</li> <li>Block parameters are available if the block was specified numerically. If the block was parameterized with a workspace parameter or with a Simulink.Parameter object, a parameter reference is available.</li> </ul>
Inline Parameters = OFF Signal storage reuse = ON Include only Simulink.Parameter and Simulink.Signal objects with global storage class = ON	<ul style="list-style-type: none"> <li>The model hierarchy is not available.</li> <li>No look-up table data entries are available.</li> </ul>
Inline Parameters = ON Signal storage reuse = OFF Include only Simulink.Parameter and Simulink.Signal objects with global storage class = OFF	<ul style="list-style-type: none"> <li>All block output signals are available<sup>1)</sup>.</li> <li>References to the parameters in the Tunable Parameters group are available.</li> </ul>
Inline Parameters = ON Signal storage reuse = OFF Include only Simulink.Parameter and Simulink.Signal objects with global storage class = ON	<ul style="list-style-type: none"> <li>The model hierarchy is not available.</li> <li>No look-up table data entries are available.</li> </ul>
Inline Parameters = OFF Signal storage reuse = OFF Include only Simulink.Parameter and Simulink.Signal objects with global storage class = OFF	<ul style="list-style-type: none"> <li>Block parameters are available if the block was specified numerically. If the block was parameterized with a workspace parameter or with a Simulink.Parameter object, a parameter reference is available.</li> <li>All block outputs are available<sup>1)</sup>.</li> </ul>
Inline Parameters = OFF Signal storage reuse = OFF Include only Simulink.Parameter and Simulink.Signal objects with global storage class = ON	<ul style="list-style-type: none"> <li>The model hierarchy is not available.</li> <li>No look-up table data entries are available.</li> </ul>

<sup>1)</sup> In individual cases, a signal might be unavailable if the 'Signal storage reuse' option is cleared.

The variable description file groups 'Labels' and 'Tunable Parameters' are affected differently by the three mentioned code option settings.

## Footnotes

- <sup>1</sup> In earlier Simulink Coder/Real-Time Workshop Releases, the mentioned parameters were localized on the Optimization page.
- <sup>2</sup> Formerly Real-Time Workshop. Simulink Coder is the successor of Real-Time Workshop, available since MATLAB R2011a.
- <sup>3</sup> For RTI-MP, the mentioned parameters are available on the Variable Description File Options page (CPU Options page) of the Multiprocessor Setup block.

## Related Documents

- *Variable Description File (TRC File) in the RTI and RTI-MP Implementation Reference.*
- *Available Variables in the Variable Description File in the RTI and RTI-MP Implementation Reference.*
- *Limitations with Variable Description File in the RTI and RTI-MP Implementation Guide*
- *Limitations Concerning the Build and Download Process in the ConfigurationDesk Real-Time Implementation Guide*
- *Conditions for the Inclusion of Variables From referenced Models in the ConfigurationDesk Real-Time Implementation Reference*
- *Limitations Regarding Simulink Variables in the Variable Description File in the ConfigurationDesk Real-Time Implementation Reference*

## Related FAQs

- -

## How to Contact dSPACE Support

dSPACE GmbH  
Rathenaustraße 26  
33102 Paderborn  
Germany

++49 5251 1638-941

<mailto:support@dspace.de>  
<http://www.dspace.com/support>

dSPACE recommends that you use the support request form on the Internet to contact dSPACE Support.

It is available at:

- <http://www.dspace.com/go/supportrequest>

## Software Updates and Patches

dSPACE strongly recommends that you download and install the most recent patches for your current dSPACE installation. Visit <http://www.dspace.com/go/patches> for software updates and patches.

## FAQ

FAQ documents are available at <http://www.dspace.com/go/faq>.

## Important Notice

This document contains proprietary information that is protected by copyright. All rights are reserved. Neither the documentation nor software may be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written consent of dSPACE GmbH.

© Copyright 2016 by:

dSPACE GmbH  
Rathenaustraße 26  
33102 Paderborn  
Germany

This publication and the contents hereof are subject to change without notice.

A list of registered dSPACE trademarks is available at:

<http://www.dspace.com/go/Trademarks>