dSPACE Release

New Features and Migration

- Release 4.1
- Product Release for dSPACE Release 4.1 July 2004
- MTest 1.1 and MTest 1.2
- MATLAB Compatibility Updates for dSPACE Release 4.1 for MATLAB R13SP2, R14 and R14SP1

December 2004

dSPACE

How to Contact dSPACE

Mail:	dSPACE GmbH Technologiepark 25 33100 Paderborn Germany
Tel.: Fax:	+49 5251 1638-0 +49 5251 66529
E-mail: Technical Support:	info@dspace.de support@dspace.de +49 5251 1638-941
Web: Subscription to e-mail newsletter:	http://www.dspace.de http://www.dspace.de/goto?SupportNewsletter

How to Contact dSPACE Support

dSPACE recommends that you use dSPACE Support Wizard to contact dSPACE support.

It is available

- On your dSPACE CD at \Diag\Tools\dSPACESupportWizard.exe
- Via Start – Programs – dSPACE Tools (after installation of the dSPACE software)
- At http://www.dspace.de/goto?supportwizard
  You can always find the latest version of dSPACE Support Wizard here.

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.de/goto?support for software updates and patches.

Important Notice

Contents

About This Document

This document provides you with an overview of the major new features of dSPACE Release 4.1 and other dSPACE products based on it. The following table helps you to find information on your dSPACE product:

For Information On	Remark
dSPACE Release 4.1, refer to *New features and enhancements of dSPACE Release 4.1* on page 6	For MATLAB R13.0.1, R13SP1, and R13SP1+
DS1103 Revision 09 for dSPACE Release 4.1, refer to *DS1103 Revision 09 for dSPACE Release 4.1* on page 7	For MATLAB R13.0.1, R13SP1, and R13SP1+; adds support for the DS1103 PPC Controller Board with board revision DS1103-09
MTest 1.1, refer to *MTest 1.1 for AutomationDesk 1.1* on page 7	For MATLAB R13.0.1, R13SP1, and R13SP1+; supports • dSPACE Release 4.1 • Automation Desk 1.1 • TargetLink 2.0 (optional)
MTest 1.2, refer to *MTest 1.2 for AutomationDesk 1.1* on page 7	For MATLAB R13.0.1, R13SP1, and R13SP1+; supports • dSPACE Release 4.1 • Automation Desk 1.1 • TargetLink 2.0 (optional)

For Information On	Remark
MATLAB R13SP2 Compatibility Update for Release 4.1, refer to *MATLAB R13SP2 Compatibility Update* on page 7	Updates dSPACE Release 4.1 for MATLAB R13SP2 compatibility and adds support for the DS1103 PPC Controller Board with board revision DS1103-09; can be installed on one of the following: • dSPACE Release 4.1 • dSPACE Release 4.1 updated with the DS1103 Revision 09 for dSPACE Release 4.1
MATLAB R14 Compatibility Update for Release 4.1, refer to *MATLAB R14 Compatibility Update* on page 7	Updates dSPACE Release 4.1 for MATLAB R14 compatibility and adds support for the DS1103 PPC Controller Board with board revision DS1103-09; can be installed on one of the following: • dSPACE Release 4.1 • dSPACE Release 4.1 updated with the DS1103 Revision 09 for dSPACE Release 4.1
MATLAB R14SP1 Compatibility Update for Release 4.1, refer to *MATLAB R14SP1 Compatibility Update* on page 8	Updates dSPACE Release 4.1 for MATLAB R14SP1 compatibility and adds support for the DS1103 PPC Controller Board with board revision DS1103-09; can be installed on one of the following: • dSPACE Release 4.1 • dSPACE Release 4.1 updated with the DS1103 Revision 09 for dSPACE Release 4.1 • dSPACE Release 4.1 updated with the MATLAB R13SP2 Compatibility Update • dSPACE Release 4.1 updated with the MATLAB R14 Compatibility Update

About dSPACE Releases

For information on how dSPACE software products are released, refer to *About dSPACE Releases* on page 9.

New features and enhancements of dSPACE Release 4.1

For a description of the key features and a summary of the major enhancements made since dSPACE Release 4.0, refer to *Key Features of dSPACE Release 4.1* on page 11.

In addition, this document provides you with information on the changes you may have to perform when you migrate from previous releases to dSPACE Release 4.1. Refer to *Migrating to dSPACE Release 4.1* on page 23.

DS1103 Revision 09 for dSPACE Release 4.1

For a description of the key features and a summary of the major enhancements made for the Product Release for dSPACE Release 4.1 July 2004, which contains DS1103 Revision 09 for dSPACE Release 4.1, refer to *Key Features of Product Release for dSPACE Release 4.1 July 2004* on page 27.

For details on migrating from DS1103 boards with board revision DS1103-07 and earlier, to boards with board revision DS1103-09, refer to *Migrating to DS1103 Revision 09 for dSPACE Release 4.1* on page 31.

MTest 1.1 for AutomationDesk 1.1

For a description of the key features and a summary of the major enhancements made for the MTest 1.1 product release, refer to *Key Features of MTest 1.1 for AutomationDesk 1.1* on page 33.

MTest projects created with an earlier version can be used without any changes in MTest 1.1. Projects created with MTest 1.1 or projects saved using MTest 1.1 are not downward compatible.

MTest 1.2 for AutomationDesk 1.1

For a description of the key features and a summary of the major enhancements made for the MTest 1.2 product release, refer to *Key Features of MTest 1.2 for AutomationDesk 1.1* on page 37.

MTest projects created with an earlier version can be used without any changes in MTest 1.2. Projects created with MTest 1.2 or projects saved using MTest 1.2 are not downward compatible.

MATLAB R13SP2 Compatibility Update

For a description of the changes introduced with the MATLAB R13SP2 Compatibility Update, refer to *Notes on the MATLAB R13SP2 Compatibility Update* on page 41.

MATLAB R14 Compatibility Update

For a description of the key features and a summary of the major enhancements made for the MATLAB R14 Compatibility Update for dSPACE Release 4.1, refer to *Key Features of the MATLAB R14 Compatibility Update for dSPACE Release 4.1* on page 43.

For details on the limitations you have to take into account when migrating to the MATLAB R14 Compatibility Update for dSPACE Release 4.1, refer to *Limitations in the MATLAB R14 Compatibility Update for dSPACE Release 4.1* on page 65.

MATLAB R14SP1 Compatibility Update

For a description of the key features and a summary of the major enhancements made for the MATLAB R14SP1 Compatibility Update for dSPACE Release 4.1, refer to *Key Features of the MATLAB R14SP1 Compatibility Update for dSPACE Release 4.1* on page 87.

For details on the limitations you have to take into account when migrating to the MATLAB R14SP1 Compatibility Update for dSPACE Release 4.1, refer to *Limitations in the MATLAB R14SP1 Compatibility Update for dSPACE Release 4.1* on page 113.

Legend

The following symbols are used in this document.

Warnings provide indispensable information to avoid severe damage to your system and/or your work.

Notes provide important information that should be kept in mind.

Tips show alternative and/or easier work methods.

Examples illustrate work methods and basic concepts, or provide ready-to-use templates.

About dSPACE Releases

dSPACE products that form a seamlessly integrated tool chain and require the same set of third-party products are assembled in a dSPACE Release.

dSPACE Release number

A dSPACE Release number consists of two digits to identify major and minor releases (major.minor, for example, dSPACE Release 3.4).

For patches, a patch number is appended to the dSPACE Release number (major.minor patch, for example, dSPACE Release 3.4p1). Patches are free of charge and can be downloaded from the dSPACE Web site.

Product Releases

Product Releases provide new products or new features for the current dSPACE Release.

MATLAB Compatibility Updates

MATLAB Compatibility Updates ensure that a dSPACE Release operates with a new MATLAB version until the availability of the next dSPACE Release (for example, MATLAB R14 Compatibility Update for dSPACE Release 4.1). They generally do not support new features of a new MATLAB version.
MATLAB Compatibility Updates can be downloaded from the dSPACE Web site.

Dependencies on third-party products

A compatibility matrix provided in the `CompList.txt` file (see the root folder of the dSPACE CD) describes the dependencies of a dSPACE Release on the products of other vendors (MATLAB, compilers, Windows, etc.).

Key Features of dSPACE Release 4.1

dSPACE Release 4.1 comes with the following key features:

Supported MATLAB releases

The software of dSPACE Release 4.1 offers full compatibility with the following releases of MATLAB from The MathWorks:

- MATLAB R13SP1+
- MATLAB R13SP1 (Release 13 with Service Pack 1)
- MATLAB Release 13.0.1.

It does not support MATLAB Release 13 (initial R13 release of August 2002), MATLAB R12.1 or earlier.

Supported operating systems

The software of dSPACE Release 4.1 supports Windows NT 4.0, Windows 2000, and Windows XP.

New supported hardware

The following new or improved hardware is supported by the software of dSPACE Release 4.1:

- DS1006 Processor Board
  The DS1006 is a processor board for a modular system using x86 processor technology. Due to its high computation power, it is ideally suited to hardware-in-the-loop applications. For details, refer to *Introduction to the Features of the DS1006* in the *DS1006 Features* document.
- DS2211 HIL I/O Board
  The DS2211 is the successor of the DS2210 HIL I/O Board. Many features are improved, for example, the signal conditioning now works with voltages up to 60 V, and the APU resolution is 16 bit. For details, refer to *DS2211 Versus DS2210* or *Feature Summary* in the *DS2211 Features* document.
- New MicroAutoBox variants
  MicroAutoBox now comes with a faster processor and a new enclosure.
- DS1005 PPC Board with 800 MHz processor clock
  The DS1005 PPC Board now comes with a processor running at 800 MHz.

Implementation: key feature summary

The new key features of the implementation software are:

- The new **RTI CAN MultiMessage Blockset** is a Simulink blockset for handling complex CAN setups in hardware-in-the-loop (HIL) applications. All the incoming RX messages and outgoing TX messages of an entire CAN controller can be controlled by a single Simulink block.
- The **RTI FlexRay Blockset 2.1** supports the FlexRay protocol 6.2 and DECOMSYS baseline 2 and can also be used in real-time models for multiprocessor systems.

For details on the key features and a list of more new features of dSPACE Release 4.1, refer to *New Implementation Features* on page 14.

Experiment and test: key feature summary

The new key features of the experiment and test software are:

- ControlDesk comes with the Remote Control feature. Using Remote Control, multiple ControlDesk client PCs can access one dSPACE real-time system with a ControlDesk server PC connected via Ethernet. Remote Control can be used to control a DS1104, which is plugged into a Server PC to have network-based access.
- AutomationDesk comes with improved usability features, for example, it now includes the Variable Browser known from ControlDesk.
- A new product, MTest 1.0.1, is an AutomationDesk add-on for convenient testing of Simulink models.

For details on the key features and a list of more new features of the experiment and test software, refer to *New Experiment and Test Features* on page 19.

For details on the update of MTest 1.0.1 to MTest 1.1, refer to *Key Features of MTest 1.1 for AutomationDesk 1.1* on page 33.

For details on updating MTest 1.1 to MTest 1.2, refer to *Key Features of MTest 1.2 for AutomationDesk 1.1* on page 37.

Documentation: key feature summary

The documentation on installing and configuring boards has been reorganized for faster access. Refer to *New Documentation Features* on page 21.

New Implementation Features

dSPACE Release 4.1 comes with several new implementation features, enhancements and changes that apply to all dSPACE boards, see below.

In addition, there are changes that apply to specific dSPACE boards and blocksets:

- *New RTI CAN MultiMessage Blockset* on page 16
- *New Implementation Features for the RTI FlexRay Blockset* on page 16
- *New Implementation Features for MicroAutoBox* on page 17

General Implementation Features, Enhancements and Changes

RTLib, RTI and RTI-MP of dSPACE Release 4.1 provide the following hardware-independent new features and enhancements. Unless stated otherwise, these features can be implemented on all dSPACE systems.

Compatibility with MATLAB

The software of dSPACE Release 4.1 offers full compatibility with the following releases of MATLAB from The MathWorks:

- MATLAB R13SP1+
- MATLAB R13SP1 (Release 13 with Service Pack 1)
- MATLAB Release 13.0.1

It does not support MATLAB Release 13 (initial R13 release of August 2002), MATLAB R12.1 or earlier.

DS1006 support

The new DS1006 Processor board is fully supported. For details, refer to the documentation for the DS1006:

- *DS1006 Features*
- *DS1006 RTI Reference*
- *DS1006 RTLib Reference*

Support for n-D Look-Up Table blocks

During real-time simulation, the parameters of multidimensional Look-Up Table (n-D), Direct Look-Up Table (n-D), and Interpolation (n-D) using PreLook-Up blocks can now be tuned via multiple 2-D slices.

Parameter representation by Real-Time Workshop Real-Time Workshop transforms the table parameters of the n-D table blocks (Look-Up Table (n-D), Direct Look-Up Table (n-D), and Interpolation (n-D) using PreLook-Up blocks) to arrays that typically differ from the dimensions of the corresponding block parameters. This also applies if you configure the table as a 1-D or 2-D table. Since dSPACE Release 4.1, ControlDesk's Table Editor instrument can handle the parameters of n-D table blocks.

Parameter representation by RTI Since dSPACE Release 4.1, RTI generates one or more `LookUpTableData` entries into the variable description file. For 1-D and 2-D tables, this corresponds to the table parameters of the standard Look-Up Table and Look-Up Table (2-D) blocks. You can therefore use ControlDesk's Table Editor instrument as usual, by connecting it with the `LookUpTableData` entry of the table. The other table data entry in the variable description file can be ignored.

Tables with more than 2 dimensions ControlDesk's Table Editor instrument cannot handle tables with more than two dimensions. To let you access the parameters of such a table, RTI subdivides it into several 2-dimensional table slices. This is especially useful if you use the 3^{rd} … n^{th} dimensions to switch between different 2-D tables.

Offline calibration using CalDesk

Generation of parameter initial values to the TRC file to support offline calibration of dSPACE targets using CalDesk.

New RTI CAN MultiMessage Blockset

dSPACE Release 4.1 provides the new RTI CAN MultiMessage Blockset with the following implementation features:

The RTI CAN MultiMessage Blockset is a new Simulink blockset for efficient and dynamic handling of complex CAN setups in hardware-in-the-loop (HIL) applications. All the incoming RX messages and outgoing TX messages of an entire CAN controller can be controlled by a single Simulink block. CAN communication is configured via database files (DBC file format or MAT file format). With the RTI CAN MultiMessage Blockset, you can configure and control a large number of CAN messages (more than 200) from a single Simulink block. This reduces the size of model files and the time required for code generation and the build process.

Supported hardware

The RTI CAN MultiMessage Blockset supports the following dSPACE platforms:

- DS1103 PPC Controller Board
- dSPACE modular systems based on the DS1005 and DS1006 containing one of the following I/O boards:
  - DS2210 HIL I/O Board
  - DS2211 HIL I/O Board
  - DS4302 CAN Interface Board

For details, refer to *CAN MultiMessage Support* in the *Features* document of your board.

New Implementation Features for the RTI FlexRay Blockset

dSPACE Release 4.1 provides the RTI FlexRay Blockset 2.1 with the following implementation features:

- FlexRay V6.2 is supported.
- The new FlexIM2+ modules are supported.
- RTIFLEXRAY supports the BASELINE 2 tool chain from DECOMSYS.
- RTIFLEXRAY can be used in a multiprocessor system.
- You can start/stop the execution of the FTCOM layer.

- You can start/stop the execution of FlexRay application tasks.
- The Timer Interrupts of the CC controller can be used as trigger sources in a Simulink model

Migrating to RTI FlexRay Blockset 2.1

FlexRay V4 is no longer supported. Some features are therefore no longer supported. Refer to *Migrating to RTI FlexRay Blockset 2.1* on page 24.

Supported hardware

The RTI FlexRay Blockset supports the following dSPACE hardware:

- DS1005 PPC Board and DS1006 Processor Board with a DS4501 containing a FlexRay IP module
- MicroAutoBox 1401/1505/1506 containing a FlexRay IP module

New Implementation Features for MicroAutoBox

dSPACE Release 4.1 provides the following new implementation features and enhancements for the new MicroAutoBox variants:

Processor update

All MicroAutoBox variants now have a faster processor. The details are as follows:

- The processor of MicroAutoBox has been changed from the PowerPC 603 to the PowerPC 750 FX processor.
- The processor clock has been increased from 300 MHz to 800 MHz.
- The bus clock has been increased from 50 MHz to 100 MHz.
- The processor provides a 512 KB level-2 cache running at 800 MHz.
- The boot firmware has been updated to Version 2.3.
- Speedup depends on the application (up to factor 10).
- The temperature ratings are basically unchanged (- 40° C to +85° C).
- Full object code compatibility is guaranteed.

Note the following points:

- Programs that require the entire computing time of an 800 MHz MicroAutoBox cause overruns on a 300 MHz MicroAutoBox.

- The faster CPU causes the power consumption to slightly increase to 9.5 W, resulting in a supply current of about 800 mA at 12 V.
- Due to the increased bus clock, the frequency of Timer B has doubled. The different values apply to Timer, Decrementer and Timer Interrupt Control functions. The software of these functions corrects this behavior to keep the timer definitions in an application unchanged. Because of this, a timer overrun will occur on the 800 MHz MicroAutoBox for events specified to be triggered at more than 50% of the maximum timer period defined for a 300 MHz MicroAutoBox. In this case, a higher prescaling rate must be specified.

New enclosures

All MicroAutoBox variants come with newly designed modular enclosures. The positions of the bolts have not changed.

New Experiment and Test Features

dSPACE Release 4.1 has many new features and enhancements for dSPACE's experiment and test software.

DS1006 support

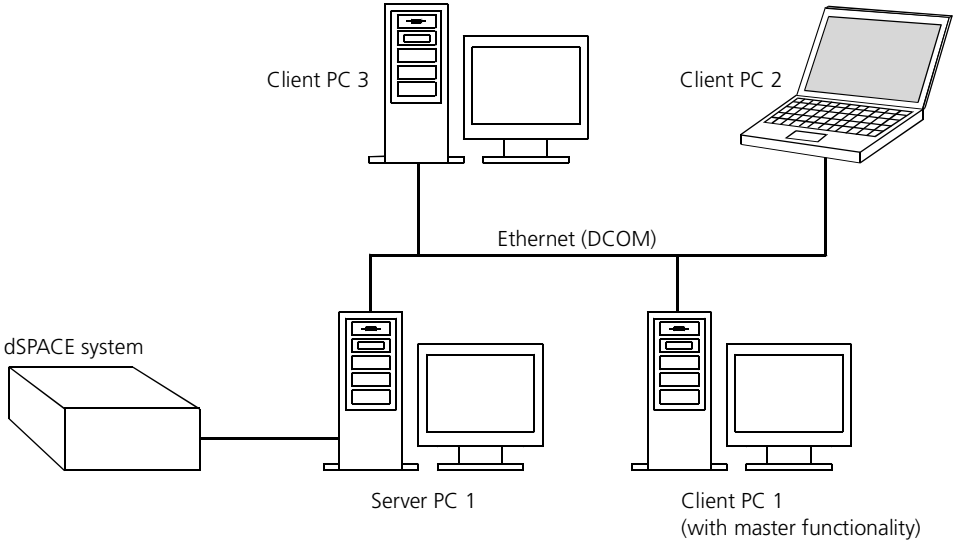
All dSPACE's experiment and test software supports the DS1006 Processor Board.

New Features for ControlDesk

ControlDesk on dSPACE Release 4.1 comes with the following new feature:

Remote Control

Using Remote Control, you can experiment with one real-time application from several PCs simultaneously via an Ethernet connection.



For details, refer to *Remote Control* in the *ControlDesk Experiment Guide*.

New Features for AutomationDesk

AutomationDesk on dSPACE Release 4.1 comes with the following new features and enhancements:

- The Data Object Editor can be used as a separate window, with a static display of the data objects from the selected automation block, or as a page in the Tool Window. In the latter display mode, the Data Object Editor's content changes simultaneously with the current block selection.
- The Data Object Selector replaces the Data Object Browser. It can now be called directly within the Data Object Editor.
- There are some new object-specific dialogs for editing data objects, for example, the Condition Editor for editing the conditions used by IfThenElse, While or Repeat automation blocks.
- The Variable Browser known from ControlDesk is integrated in the Platform Configuration dialog to make it easier to parameterize platform access programs.
- The syntax for accessing data objects has been made more simple.
- Errors that occur during an automation task are displayed graphically in the Result Browser.

New Documentation Features

With dSPACE Release 4.1, dSPACE completely reorganized the Installation and Configuration documentation for dSPACE systems (hardware and software).

Former Installation and Configuration documents describe the installation and configuration procedures for a system in one document. To keep the information on related procedures in one place, the information has been split into two separate documents:

- *dSPACE Software Installation and Management Guide*
  This guide provides detailed instructions for authorized users with administrator rights on installing and handling the dSPACE software for the different platforms. It also shows you how to manage dSPACE licenses.
- *DSxxxx Hardware Installation and Configuration* documents
  These guides describe hardware installation procedures and explain how to get started with your dSPACE system, using an appropriate example.

Migrating to dSPACE Release 4.1

After you install dSPACE Release 4.1, it may be necessary to carry out additional steps.

Migrating from dSPACE Release 4.0

Refer to *How to Migrate from dSPACE Release 4.0* on page 24.

Migrating from dSPACE Release 3.5 or earlier

Refer to *How to Migrate from dSPACE Release 3.5 or Earlier* on page 25.

How to Migrate from dSPACE Release 4.0

Note the following points when migrating from dSPACE Release 4.0 to dSPACE Release 4.1:

Migrating to RTI FlexRay Blockset 2.1

The RTI FlexRay Blockset 2.1 no longer supports the FlexRay V4 anymore. This concerns the following features:

- The following interrupt types are no longer supported:
  - Silence Violation Error
  - SOC Error
  - Protocol Error
- The following status information is no longer available:
  - Buffer lost bit
  - Cycle Counter error bit
  - Passive Flag
  - Slot mismatch bit
  - Synchronization lost bit

For this reason, RTIFLEXRAY Interrupt blocks and RTIFLEXRAY Status blocks which originate from RTIFLEXRAY 1.1 can no longer be used with RTIFLEXRAY 2.1. You must remove these blocks from your Simulink model.

The first time you open a model built under RTI FlexRay Blockset 1.1, the Host Setup block is automatically updated by the RTI FlexRay Blockset 2.1. No user interaction is therefore required to update the RTIFLEXRAY Host Setup block.

Note that models that were saved with RTI FlexRay Blockset 2.1 can no longer be used with RTI FlexRay Blockset 1.1.

MotionDesk Blockset

The Gigalink_Send, Gigalink_Receive, and Gigalink_Status Blocks in the MotionDesk Blockset are supported only for the DS1005. If you migrate a model from the DS1005 to the new DS1006, you must remove these blocks from your model.

How to Migrate from dSPACE Release 3.5 or Earlier

To migrate from dSPACE Release 3.5 or earlier to dSPACE Release 4.1, you have to migrate step by step via the intervening dSPACE Releases.

For example, if you want to migrate from dSPACE Release 3.4 to dSPACE Release 4.1, you have to follow the migration steps given in:

1. New Features and Migration of dSPACE Release 3.5
2. New Features and Migration of dSPACE Release 4.0
3. Finally, the migration steps given in *How to Migrate from dSPACE Release 4.0* on page 24.

The New Features and Migration documents for previous releases are available via Internet and on the dSPACE CD:

- Download them from http://www.dspace.de/goto?migration.
- Read them from the dSPACE CD (see the `\Doc\Print` folder). The PDF files are named `NewFeaturesAndMigrationxx.pdf`, where xx stands for the version or release number.

Key Features of Product Release for dSPACE Release 4.1 July 2004

The software of the DS1103 product release supports the new DS1103 PPC Controller Board with board revision DS1103-09. The board and its new software come with the following key features.

Improved hardware features

The DS1103 PPC Controller Board now has a faster processor. The details are as follows:

- The processor has been changed from the PowerPC 604e to the PowerPC 750GX.
- The processor clock has been increased from 400 MHz to 1 GHz.
- The bus clock has been increased from 66.63 MHz to 133.33 MHz.
- The processor provides a 1 MB level-2 cache running at 1 GHz.

- The board provides 128 MB DRAM:
  - 32 MB as application memory instead of 2 MB as with older board revisions
  - 96 MB communication memory
- The board has a new passive cooling system.
- Speedup depends on the application.

For further information on the technical characteristics, refer to *DS1103 Data Sheet* in the *DS1103 Hardware Installation and Configuration* document.

Improved I/O features

Some I/O components have been modified:

- Improved ADC unit:
  - The conversion time of the multiplexed channels has been improved from 4 µs to 1 µs (16 bit resolution).
  - The resolution of the parallel channels has been increased from 12 bit to 16 bit with the same conversion time of 800 ns.
- The resolution of the DAC unit has been increased from 14 bit to 16 bit with the same settling time.
- The UART of the serial interface is now driven by a PLL with a more precise baud rate setting.

Improved implementation features

The DS1103 software provides the following new features:

- Synchronous triggering of the I/O components by software or by an external signal:
  - External start of conversion of all ADCs.
  - External update of DAC outputs.
  - External strobe for the incremental encoder interface.

  This feature is provided by new RTI blocks and RTLib functions. For details, refer to:
  - *Synchronizing I/O Features of the Master PPC* in the *DS1103 Features* document
  - *Synchronizing I/O Unit* in the *DS1103 RTI Reference*
  - *Synchronous I/O Trigger* in the *DS1103 RTLib Reference*

The synchronous I/O trigger requires the DS1103 with board revision 09 or later, and RTLib1103 Version 2.0 or later.
Programs that contain the setup functions of the synchronous I/O trigger, for example `ds1103_syncin_trigger`, cannot be executed on older boards. The trigger setup functions of the I/O components, for example `ds1103_adc_trigger_setup`, can be used in the program but the trigger state parameter must be set to disabled. Otherwise the program exits with an error message.

- New RTLib functions for reading positions or delta positions of the incremental encoder within an interrupt service routine using the external signal line. For details, refer to:
  - *ds1103_inc_position_read_immediately*
  - *ds1103_inc_delta_position_read_immediately*
  - *ds1103_inc_counter_rd_immediately*

  in the *DS1103 RTLib Reference*.

These functions require DS1103 with board revision 09 or later, and RTLib1103 Version 2.0 or later.
Programs that contain these functions cannot be executed on older boards. The program exits with an error message.

- Interrupts have been improved for the use of synchronous I/O triggering. For details, refer to *Interrupts Provided by the DS1103* in the *DS1103 Features* document.
- There is a new RTLib function for reading from all A/D converters simultaneously. For details, refer to *ds1103_adc_multi_converter_read* in the *DS1103 RTLib Reference*.

This feature requires RTLib1103 Version 2.0 or later.
Programs that contain this function can be executed on all DS1103 board revisions.

AutoBoot firmware requirement

As of board revision DS1103-09 (1 GHz CPU clock, no fan), dSPACE AutoBoot Firmware Version 2.1 is required. For further information, refer to *Migrating to DS1103 Revision 09 for dSPACE Release 4.1*.

Migrating to DS1103 Revision 09 for dSPACE Release 4.1

If you already use a DS1103 PPC Controller Board with board revision DS1103-07 and earlier, note the following points when porting applications to boards with revision DS1103-09:

- Programs that were used with older boards must be recompiled.
- Programs that are compiled using RTLib1103 Version 2.0 and board revision DS1103-09 are executable on older boards with the following limitiations:
  - The external trigger source functions, for example, `ds1103_syncin_edge_setup`, cannot be used.
  - The external trigger for the I/O components cannot be disabled by using the trigger setup functions, for example, `ds1103_adc_trigger_setup`.

  Programs that use one of the described features affected by these limitations exit with an error message.

- Programs that require the entire computing time of a 1 GHz DS1103 cause overruns on a 400 MHz board.
- Due to the increased bus clock, the frequency of all timers has doubled. The different values apply to Timer, Decrementer and Timer Interrupt Control functions. The software of these functions corrects this behavior to keep the timer definitions in an application unchanged. Because of this, a timer overrun will occur on the 1 GHz DS1103 for events specified to be triggered at more than 50% of the maximum timer period defined for a 400 MHz DS1103. In this case, a higher prescaling rate must be specified. For further information, refer to *ds1103_timerB_init* in the *DS1103 RTLib Reference*.

AutoBoot firmware

The dSPACE AutoBoot firmware is located in the on-board flash memory of the slot CPU inside the expansion box/AutoBox. The firmware provides the necessary files (for example, application loaders) to support booting of the expansion box from the slot CPU.

As of board revision DS1103-09 (1 GHz CPU clock, no fan), dSPACE AutoBoot Firmware Version 2.1 is required.

- Expansion boxes (including a slot CPU) shipped since dSPACE Release 4.1 (April 2004) are already equipped with Firmware Version 2.1. No update is required.
- Expansion boxes shipped before dSPACE Release 4.1 need a firmware update. For details, refer to *Basics on Updating the AutoBoot Firmware* in the *DS1103 Hardware Installation and Configuration* document.

Key Features of MTest 1.1 for AutomationDesk 1.1

MTest is an AutomationDesk add-on for systematic testing of Simulink and TargetLink models. It consists of a special AutomationDesk project template to create tests in a predefined structure, like using an application wizard. It provides an interface to the CTE/ES (Classification Tree Editor for Embedded Systems) from RazorCat for creating systematic test data based on the classification tree method. To use the whole functionality of MTest, you need the following software.

Software requirements

- MATLAB R13 (R13.0.1 and R13 Service Pack 1)
- dSPACE Release 4.1
- AutomationDesk 1.1

Optional:

- TargetLink 2.0
- Simulink Performance Toolbox for using the model coverage.

- CTE/ES Version 2.3.18 or higher (Copyright by RazorCat Development GmbH) for using systematic test data.

  MTest can be distributed together with an OEM version of the CTE/ES on demand.

New Features

The main new feature of MTest 1.1 is TargetLink support, in addition to further new and improved features.

TargetLink support

Earlier versions of MTest support testing of Simulink models only. MTest 1.1 is approved for use with TargetLink 2.0.

This means that you can now use MTest to test a controller model within various test execution environments. You can analyze the model implementation in Simulink and its transitions to the TargetLink model and to TargetLink code. MTest provides test execution environments for:

- Model-in-the-loop (MIL) simulation mode
- Software-in-the-loop (SIL) simulation mode
- Processor-in-the-loop (PIL) simulation mode

For further information on the simulation modes, refer to the *TargetLink Production Code Generation Guide*.

Model and code coverage

Ideally, tests cover 100% of the test object to guarantee that the results are relevant under all conditions. The model coverage tool examines the test flow within a selected subsystem of a Simulink model according to the specified parameters and variables. With the code coverage tool, you can examine which parts of the target code were covered by the test. There are various analyses modes for model and code coverage which are supported by MTest, for example, the decision coverage mode for Simulink models or the statement coverage mode for TargetLink code. For further information, refer to the *AutomationDesk MTest Guide*.

Simulation properties

The simulation properties which you have specified for a test sequence can now be reset to the project-specific simulation properties by using the **Refresh Simulation Properties** command.

Online help

The online help has been improved to facilitate context-sensitive access to the data objects displayed in the Result Browser and to the test execution environment elements.

Preference settings

For setting certain MTest options, there is a separate tool that you can find in the **Start** menu at **dSPACE Tools - AutomationDesk - MTest Preferences**. You can customize various search paths, callback function handling, and whether the MATLAB output will be dumped.

Demo project

The MTest demo folder in *%DSPACE_ROOT%/Demos/AutomationDesk/MTest/* now contains three projects:

- *Demo1.adp* shows the handling of Simulink models.
- *Demo2.adp* is an example of using a TargetLink model in the different test execution environments.
- *SampleSession.adp* contains the test which you create when you follow the tutorial instructions in the AutomationDesk MTest Guide.

Key Features of MTest 1.2 for AutomationDesk 1.1

MTest is an AutomationDesk add-on for systematic testing of Simulink and TargetLink models. It consists of a special AutomationDesk project template to create tests in a predefined structure, like using an application wizard. It provides an interface to the Classification Tree Editor for Embedded Systems (CTE/ES) from RazorCat for creating systematic test data based on the classification tree method. To use the whole functionality of MTest, you need the following software.

Software requirements

- MATLAB R13 (R13.0.1 and R13 Service Pack 1)
- dSPACE Release 4.1
- AutomationDesk 1.1

Optional:

- TargetLink 2.0 with the TargetLink 2.0 Compatibility Update for MTest 1.2

- MATLAB support for model verification:
  - Simulink Performance Toolbox using MATLAB R13.0.1 or R13SP1
  - Simulink Verification and Validation using MATLAB R14 or R14SP1
- CTE/ES Version 2.3.18 or higher (copyright RazorCat Development GmbH) for using systematic test data.

  MTest is supplied together with an OEM version of the CTE/ES on request.

- MTest 1.2 can be also used if you have updated TargetLink 2.0 with the MATLAB R14 Compatibility Update for TargetLink 2.0.
- If you use MTest 1.2 with MATLAB R13 SP2 and R14 SP1, you have no TargetLink support.

  For latest compatibility information of MATLAB R13SP2 and R14SP1 check http://www.dspace.de/goto?MTest12.

New Features

MTest 1.2 provides the following new and improved features.

CTE Rule Manager

If the CTE/ES is open, you can import CTE rules which specify a default classification tree. With earlier MTest versions, you were able to create and edit CTE rules using an ASCII editor. With MTest 1.2, you can use the integrated CTE Rule Manager, which provides a graphical user interface for this purpose. You can manage rules conveniently and do not need to learn the syntax of CTE rules.

Data Editor

The Data Editor comes with an improved graphical user interface for editing signals:

- New commands to append and delete elements to the signal course
- New command to specify gridlines in the pattern window of the Data Editor
- Improved navigation in the table editor of the Data Editor via Ctrl + End, Ctrl + Home, and the arrow keys

- Improved highlighting of selected cells in the table editor and the pattern window

Global update of reference data

Output data from an executed test can be used as new reference data for the next test execution. You can copy the output data belonging to a specified test execution environment to the reference data of the own or another test execution environment. You can update the reference data for the entire MTest project with the new Update Reference Data command.

Improved Properties dialog of the test element

The Properties dialog of a test element has been improved by moving the management of corresponding model operation files to an additional dialog.

Support of Simulink Model Verification blocks

If your model contains blocks from the Simulink® Model Verification Library, you can now specify a callback function to log the first 30 assertions to the MTest result.

Improved report generation

The contents of an MTest result and the generated report have been improved:

- The test result contains new data objects for the difference data of output and reference data, and assertion data provided by model verification blocks.
- The test and output data in a report are preceded by a tabular overview of the signals used.
- The report does not contain diagrams of constant values. These values are documented only in the tabular overview. Nonconstant values are also plotted in a diagram.
- If you have specified reference data, test data and reference data are both plotted in one diagram. In addition, the differences between the data are plotted in another diagram.
- If you executed the test for several test execution environments with different output data, there is one result diagram containing the output data of all test execution environments.
- If your model contains model verification blocks, the first 30 assertions are stored in the report.

- If you test a TargetLink model, you can specify internal TargetLink variables, the TargetLink execution time, and the TargetLink stack size to be logged in the result and the report.

Testing of combined subsystems

If you have specified a TargetLink model to be tested, you can now deactivate the TargetLink filter to display non-TargetLink subsystems. For example, for indirect stimulation it is necessary to test a subsystem containing both controller and model plant.

Improved data handling for code generation

Because some target compilers have restrictions on the length of path names, all relevant data for code generation was copied to a temporary folder in earlier versions of MTest. With MTest 1.2, the folder of the test frame is mapped to a logical drive. Compilation is prepared and performed faster than before.

There must be a free logical drive name (A - Z).

Notes on the MATLAB R13SP2 Compatibility Update

The MATLAB R13SP2 Compatibility Update enables you to use dSPACE Release 4.1 together with MATLAB Release 13 with Service Pack 2. For simplicity, the term "MATLAB R13SP2" is used throughout the rest of this document.

- If you are a new dSPACE user, you can start working with MATLAB R13SP2 and dSPACE Release 4.1.
- If you have used MATLAB R13, R13.0.1, R13SP1, or R13SP1+ and dSPACE Release 4.1 before, you can update to MATLAB R13SP2. Models that you have created do not need to be modified.

- If you are installing MATLAB R13SP2 on a computer that already has an earlier version of MATLAB installed on it, you must install MATLAB R13SP2 in a new installation folder.

- After the MATLAB R13SP2 Compatibility Update has been installed, the interaction between MATLAB and dSPACE software is configured for MATLAB R13SP2 only. You can uninstall the older MATLAB release.
- It is not possible to install the MATLAB R13SP2 Compatibility Update on dSPACE Release 4.1 installations that have already been updated for MATLAB R14 or MATLAB R14SP1 compatibility.

Support of DS1103 Revision 09 for dSPACE Release 4.1

The MATLAB R13SP2 Compatibility Update includes support for the DS1103 PPC Controller Board with board revision DS1103-09. You can install it on one of the following product installations:

- dSPACE Release 4.1
- dSPACE Release 4.1 updated with the DS1103 Revision 09 for dSPACE Release 4.1

Do not install the DS1103 Revision 09 for dSPACE Release 4.1 after installing the MATLAB R13SP2 Compatibility Update.

Migration

Models created with dSPACE Release 4.1 or dSPACE Release 4.1 with the DS1103 Revision 09 for dSPACE Release 4.1 do not need to be modified for use with R13SP2.

Key Features of the MATLAB R14 Compatibility Update for dSPACE Release 4.1

The MATLAB R14 Compatibility Update for dSPACE Release 4.1 enables you to use dSPACE Release 4.1 together with MATLAB, Simulink, Real-Time Workshop and further products of The MathWorks R14 Release. For simplicity, the term "MATLAB R14" is used throughout the rest of this document. MATLAB R14 provides several new features. This document describes how to use the new features when working with RTI and RTI-MP. For details on the new MATLAB features, refer to the MATLAB R14 documentation. Note that RTI does not support all new MATLAB features. For details on the limitations that apply when working with the MATLAB R14 Compatibility Update, refer to *Limitations in the MATLAB R14 Compatibility Update for dSPACE Release 4.1* on page 65.

- Before you start working with the MATLAB R14 Compatibility Update, you have to install patches for MATLAB R14. For details, refer to *Real-Time Workshop patch required for code generation* on page 76 and *Patch required for correct execution of periodic tasks* on page 76.
- When you work with the MATLAB R14 Compatibility Update for dSPACE Release 4.1, it is assumed that you have knowledge in handling RTI, and that you know the RTI and RTI-MP Implementation Guide and the RTI and RTI-MP Implementation Reference. It is also assumed that you know the MATLAB R14 Release Notes for MATLAB, Simulink, and Real-Time Workshop.

General changes

The MATLAB R14 Compatibility Update has several new implementation features, enhancements and changes. Refer to *General Features, Enhancements and Changes* on page 45.

Creating models and configuring RTI

MATLAB R14 provides several new features for model creation and parameterization. Refer to *New Features for Creating Models and Configuring RTI* on page 46.

Handling tasks

When you work with the MATLAB R14 Compatibility Update, you should note some changes in task handling. Refer to *Changes in Task Handling* on page 54.

Building models

The MATLAB R14 Compatibility Update supports some of the new features of MATLAB R14 for building models. Refer to *New Features for Building Models* on page 55.

Simulating models

You should note some changes in MATLAB R14 regarding the simulation of models. Refer to *New Simulation Features* on page 61.

General Features, Enhancements and Changes

The MATLAB R14 Compatibility Update provides the following general enhancements and changes.

dSPACE Release 4.1 support of MATLAB R14

After you install the MATLAB R14 Compatibility Update, dSPACE Release 4.1 supports MATLAB R14. This includes all RTI products, ControlDesk and MLIB. All functions of the RTI products are available unless otherwise stated. For details on limitations, refer to *Limitations in the MATLAB R14 Compatibility Update for dSPACE Release 4.1* on page 65, specifically to *Real-Time Workshop patch required for code generation* on page 76 and to *Patch required for correct execution of periodic tasks* on page 76.

Support of dSPACE release

The MATLAB R14 Compatibility Update supports exclusively the following dSPACE products:

- dSPACE Release 4.1
- The DS1103 Revision 09 for dSPACE Release 4.1

MTest 1.1 is not supported by the MATLAB R14 Compatibility Update.

Compatibility of RTI models

RTI models created with dSPACE Release 4.1 or earlier provide full functionality when you work with the MATLAB R14 Compatibility Update. Nevertheless, you should be aware of the limitations described in *Limitations in the MATLAB R14 Compatibility Update for dSPACE Release 4.1* on page 65.

If you save an RTI model with the MATLAB R14 Compatibility Update, you can no longer use it with earlier dSPACE releases.

RTI MAT file support

RTI saves MAT files with the -v6 option to enable users with a MATLAB R13.x version to work with the MAT files. This applies to

- RTI's DS2210 and DS2211APU blocksets
- The layouter in the RTI CAN MultiMessage Blockset's MainBlock

New Features for Creating Models and Configuring RTI

This MATLAB Compatibility Update supports many features provided by MATLAB R14. The following new features for creating a Simulink model and configuring it for Simulink, Real-Time Workshop, and RTI are described in detail below:

- Model Explorer
- Configuration sets
- Signal & Scope Manager
- Duplicate input ports
- Environment Controller block

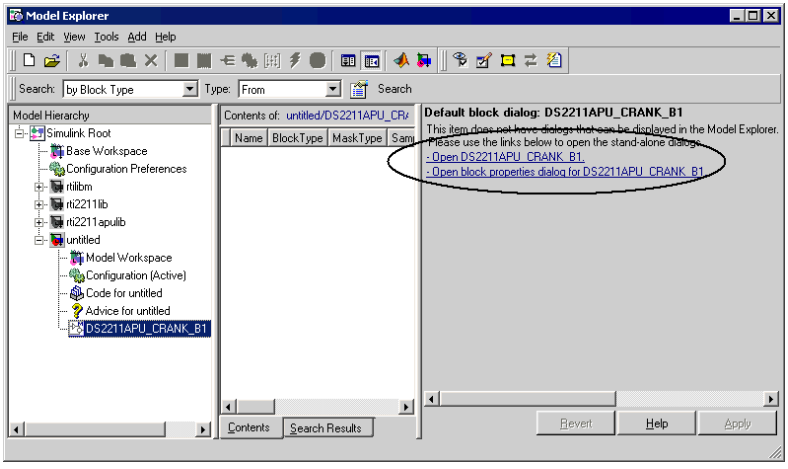
For detailed information on features which are not supported by this MATLAB Compatibility Update, refer to *Limitations in the MATLAB R14 Compatibility Update for dSPACE Release 4.1* on page 65.

Model Explorer

Configuring RTI RTI supports the Model Explorer provided by MATLAB R14. The Model Explorer allows you to locate and configure parameters in a Simulink model or a Stateflow chart. When you work with RTI, the Model Explorer helps you to perform the following actions:

- Specifying configuration preferences for RTI options
- Specifying RTI options for a model
- Configuring RTI blocks with a Simulink mask dialog

MATLAB provides a system of graphics objects for implementing visualization functions, for example, figure windows, axes, lines, etc., which is called MATLAB Handle Graphics. If an RTI block has a dialog implemented in MATLAB Handle Graphics, the Model Explorer displays a link for you to open the dialog with. The following illustration shows an example:



Configuring RTI-MP You can divide a large model into smaller parts and assign these parts to different CPUs. In this case, you have to use the Multiprocessor Setup dialog to configure the model for use on a multiprocessor system. In addition, you must configure the following options in the main model:

- All Simulink data import/export options
- All Simulink and Real-Time Workshop optimization options
- All Simulink diagnostic options
- All Simulink and Real-Time Workshop hardware implementation options
- The following Real-Time Workshop options:
  - The Documentation options on the General page
  - All options on the Comments page
  - All options on the Symbol page

- All options on the Custom Code page
- All options on the Debug page
- All options on the Interface page
- The **Generate code only** option of Real-Time Workshop

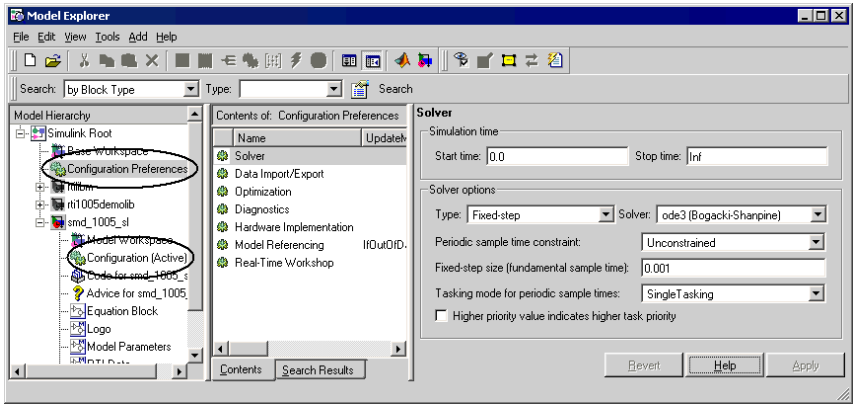
When the model is separated, RTI-MP ensures that the configuration of the main model is transferred to all submodels properly.

Model Referencing options are not supported. Refer to *Model referencing is not supported* on page 67.

Configuration sets

A configuration set is a set of values for your model's configuration parameters. You can create several configuration sets containing different values for any model configuration parameter, and associate them with the model. You can activate one of the configuration sets. The active set then contains the current values of the model's configuration parameters. There are two different types of configuration sets:

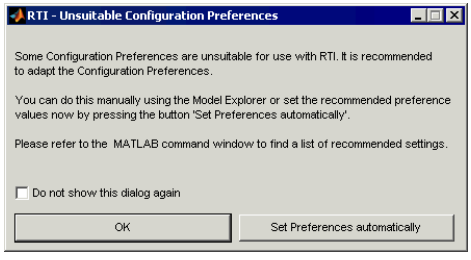
- The Configuration Preferences set
- Model-specific configuration sets



Configuration Preferences set In the Configuration Preferences set, you can specify default values for all model properties, which are transferred to the model's configuration set when you create a new model. When you start MATLAB or activate a different RTI platform, RTI checks if the settings in the Configuration Preferences set are suitable for RTI. The following table displays recommended settings for RTI-relevant properties:

Property	Value
Start time	0.0
Stop time	Inf
Solver type	Fixed-step
Periodic sample time constraint	Unconstrained
Fixed step-size	≠ auto
Higher priority value indicates higher task priority	Off
Block reduction optimization	Off
Signal storage reuse	Off
RTW system target file	rti<xxxx>.tlc (depending on the active target)

If the current values differ from the values which are recommended for RTI, a message dialog is displayed. You can set the recommended preferences automatically or edit the preferences manually after closing the message box.



If RTI detects unsuitable settings in your Preferences Configuration set, the current and the recommended settings are displayed in the MATLAB Command Window on MATLAB start-up, for example:

```
Some Configuration Preferences are unsuitable for use with RTI. It is recommended
to adapt the Configuration Preferences.

Note that, using the Model Explorer, you need to explicitly save manual changes to
the Configuration Preferences to restore them with the next MATLAB start.

  - Real-Time Workshop -            current setting            recommended setting
  RTW system target file:           'rti1104.tlc'          ->  'rti1005.tlc'
```

- The Configuration Preferences set is a Simulink feature and therefore holds the same values independently of the currently activated RTI platform.
- The RTI Task Configuration dialog lets you assign priorities to the different tasks of a model and configure the overrun strategy. You cannot open the RTI Task Configuration dialog for the Configuration Preferences set.

Model-specific configuration sets The model-specific configuration sets are saved together with the model file. In MATLAB R14, you can specify different configuration sets for your model, for example, one for RTI1103, and one for RTI1005. In this case, for example, if you change from RTI1103 to RTI1005 platform support, you can simply activate the corresponding configuration set and continue working with the same model and without altering the configuration used for RTI1103.

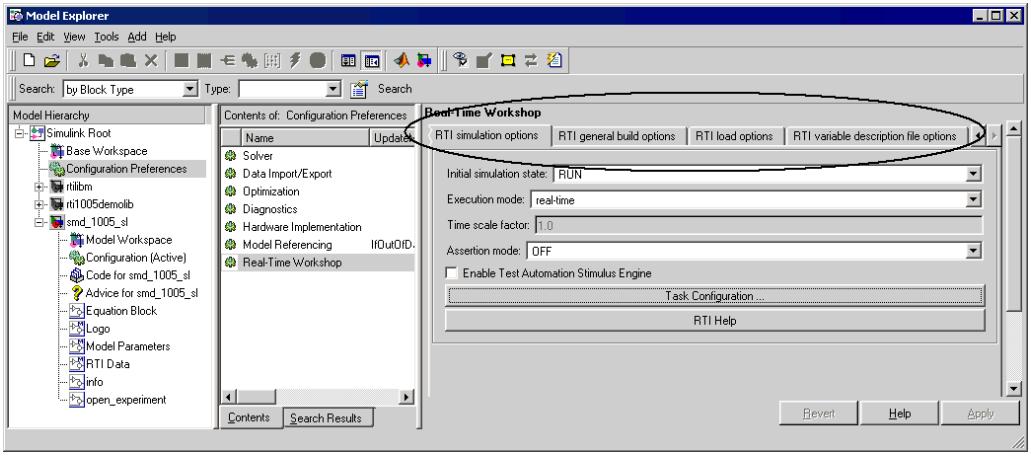
For RTI single-processor models, you can configure RTI options in active and inactive configuration sets for which you have specified an RTI<xxxx> target. Of course, you can also perform all standard Simulink operations on configuration sets, for example:

- Activating a specific configuration set
- Copying configuration sets within a model and to another model
- Copying the configuration preferences to a model

For single-processor models, you can specify the RTI options for the model-specific configuration either in the Model Explorer or in the Configuration Parameters dialog. Both methods lead to the same result.

To specify RTI options for multiprocessor systems, you have to use the Multiprocessor Setup dialog.

If you activate an RTI platform on the General page, for example, rti1005.tlc, the following RTI-specific pages appear in the Model Explorer's dialog:



- You can specify an RTI-specific configuration independently of the RTI platform that is currently activated. For example, while RTI1005 is the currently active platform support, you can create a new configuration set and configure it for RTI1103 (or any other RTI platform support) without the need to activate the RTI1103 platform support.
- Additionally, you can open a model that is configured for a specific RTI platform while another RTI platform is activated. For example, while RTI1005 is the currently active platform, you can open a model that is configured for RTI1103.

- You can start an RTI build process only if the RTI platform specified in the active configuration set is the same as the active platform. Otherwise, an error message is displayed.
- RTI task configuration is based on the activated RTI platform and the active configuration set of a model. Task configuration cannot be specified for each configuration set separately. Thus, RTI task configuration is the same for all RTI-specific configuration sets.
- You can open the RTI Task Configuration dialog only for an active configuration set and if the specified target is the same as the activated platform.

Target-specific default values If you change the RTI<xxxx> target in the Configuration Preferences set or in a model-specific configuration set, target-specific values are reset to their target-specific default values. All other values remain unchanged.

Signal & Scope Manager

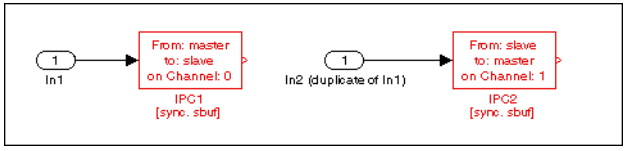
RTI fully supports the Signal & Scope Manager, introduced with MATLAB R14. It helps you manage signal generators and viewers.

Signal generators and viewers defined with the Signals & Scope Manager are not generated into the variable description file (TRC file).

Duplicate input ports

RTI and RTI-MP support duplicates of input ports in your model. The duplicates allow you to simplify block diagrams by eliminating unnecessary signal lines. In the variable description file (TRC file), they are displayed as common input ports. RTI-MP checks if the duplicate input ports are modeled correctly.

For example, specifying the same input signal with duplicate input ports as coming from different CPUs is not permitted.



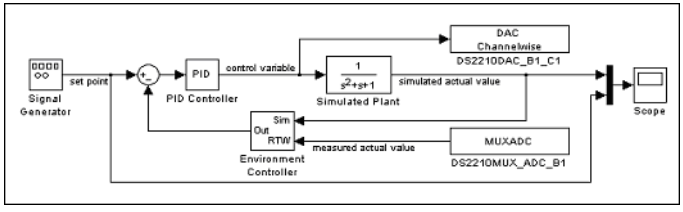
Environment Controller block

RTI supports the Environment Controller block. The Environment Controller block lets you create branches of a block diagram that apply only to simulation or only to code generation.

The Environment Controller block has the following inports:

- Sim
- RTW

If the model containing the block is simulated with Simulink, it outputs the signal at the Sim inport. If code is generated from the model, it outputs the signal at the RTW inport. The following illustration shows a simple example of how to use the Environment Controller block with RTI I/O blocks:

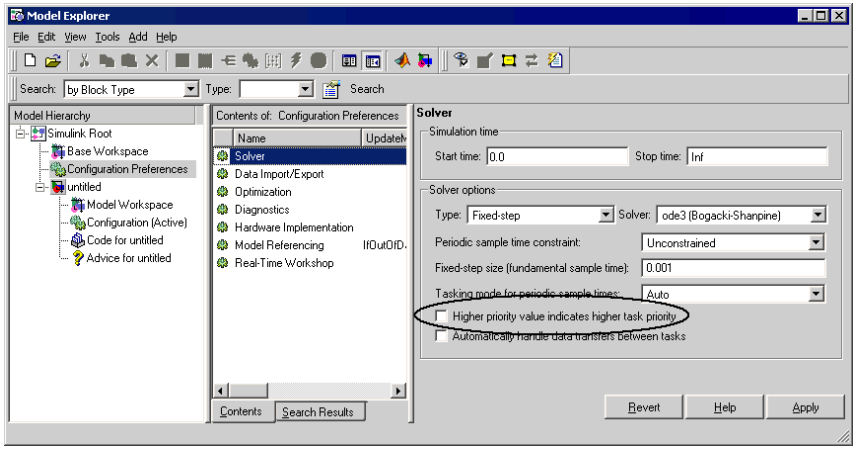


Changes in Task Handling

When migrating to the MATLAB R14 Compatibility Update, you have to note the following changes in task handling.

Handling task priorities

MATLAB R14 provides the **Higher priority value indicates higher task priority** option on the Solver dialog of the Model Explorer.



For RTI, you must clear this checkbox. RTI checks for the correct setting of this checkbox when starting the build process.

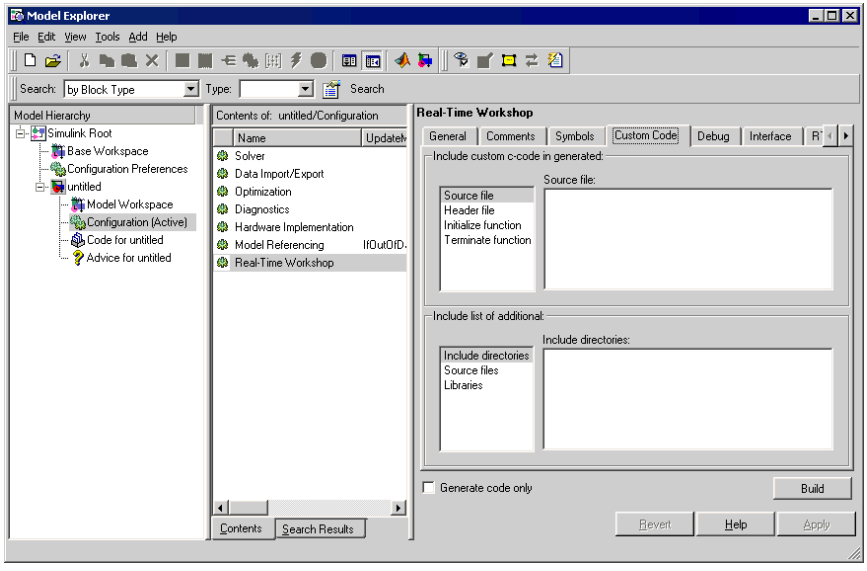
New Features for Building Models

The MATLAB R14 Compatibility Update supports new features of MATLAB R14 for building models. This section describes how to use the following new features with RTI and RTI-MP:

- Real-Time Workshop's Custom Code page
- Hardware Implementation dialog
- Target floating point math environment
- Shared utilities directory

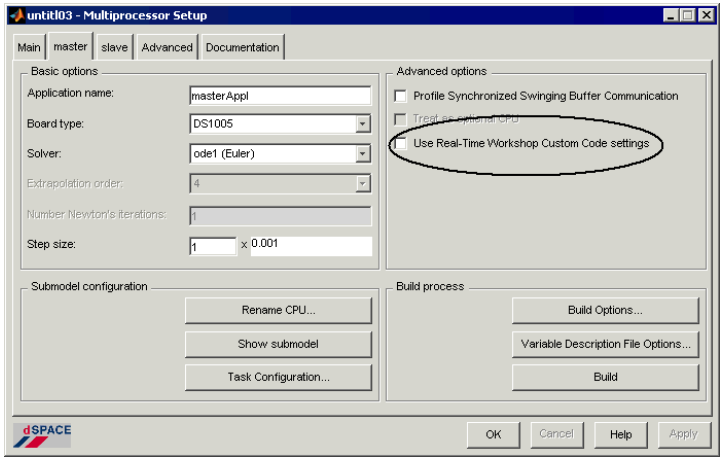
Custom Code page

RTI and RTI-MP support all the settings you can specify in Real-Time Workshop's Custom Code page.



- In earlier dSPACE releases, you could already use the RTI User-Code file (`USR.C`) to add hand-written C code to your real-time application. With this MATLAB Compatibility Update, you can use the settings in the **Include custom c-code in generated** frame in Real-Time Workshop's Custom Code page in addition to the settings in the RTI User-Code file (`USR.C`).
- In earlier dSPACE releases, you could already use the RTI User Makefile (`USR.MK`) to define extra search paths (for S-functions or user-defined C source code), user libraries, etc. With this MATLAB Compatibility Update, you can use the settings in the **Include list of additional** frame in Real-Time Workshop's Custom Code page instead of, or in addition to, the settings in the RTI User Makefile (`USR.MK`).

For RTI-MP, you have to specify the settings in Real-Time Workshop's Custom Code page of the main model. To have the settings transferred to a submodel, you have to select the **Use Real-Time Workshop Custom Code settings** checkbox on the relevant CPU pages of the Multiprocessor Setup dialog.

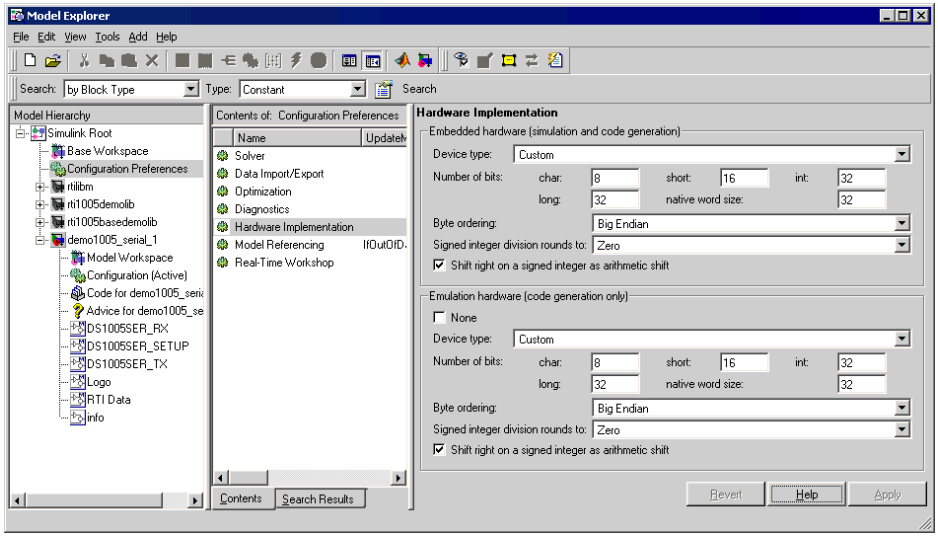


If you use Real-Time Workshop's Custom Code page, you cannot specify different custom code for each submodel of a multiprocessor system. If this is required, you must use the RTI User-Code files (`USR.C`) and RTI User Makefiles (`USR.MK`).

Hardware Implementation dialog

RTI supports the settings on the Hardware Implementation dialog, which lets you specify the characteristics of the hardware to be used to implement the system represented by your model. The Hardware Implementation dialog contains the following two groups of properties:

- **Embedded hardware (simulation and code generation)** lets you specify the properties of the production-type hardware.
- **Emulation hardware (code generation only)** lets you specify the properties of the hardware used to test the code generated from your model.



For details on these Simulink and Real-Time Workshop features, refer to the MATLAB R14 documentation.

Hardware Implementation configuration RTI and RTI-MP support code generation for the following configuration types:

- The RTI platform represents the production-type hardware, and no emulation hardware is used. You have to perform the following steps for configuration:
  - Under Embedded hardware, specify the properties applying to your dSPACE board.
  - Under Emulation hardware, select **None**.
- The behavior of arbitrary production-type hardware is emulated on the RTI platform. You have to perform the following steps for configuration:
  - Under Embedded hardware, specify arbitrary properties.
  - Under Emulation hardware, specify the properties applying to your dSPACE board.

    The behavior of the hardware specified under Embedded hardware is emulated on the RTI platform.

If you select an RTI platform on the General page of the Real-Time Workshop dialog, the following values are automatically configured on the Hardware Implementation dialog, depending on your dSPACE board:

Property	**DS1005, DS1103, DS1104, DS1401**	**DS1006**
Device type	Custom	Custom
Number of bits		
char	8	8
short	16	16
int	32	32
long	32	32
native word size	32	32
Byte ordering	Big Endian	Little Endian
Signed integer division rounds to	Zero	Zero
Shift right on a signed integer as arithmetic shift	True	True
Emulation hardware (code generation only)	None	None

RTI and RTI-MP check if the Hardware Implementation settings are specified correctly.

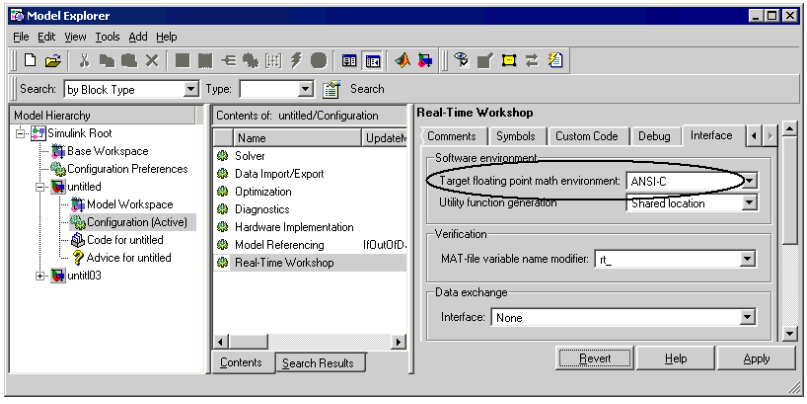
When you change the RTI system target file of your model to another RTI platform, for example, from rti1005.tlc to rti1103.tlc, all hardware characteristics are reset to the values specified in the above table.

Configuring hardware for RTI-MP When the model is separated, RTI-MP checks if the system target file specified in the main model's configuration set matches the dSPACE board type specified in the Multiprocessor Setup dialog. For example, if rti1005.tlc is specified in the configuration set, DS1005 must be specified in the Multiprocessor Setup dialog. If the settings do not match, model separation is aborted and an error message is displayed.

For RTI-MP, you cannot specify the Hardware Implementation properties separately for each submodel. When the model is separated for RTI-MP, the main model's Hardware Implementation settings are transferred to the corresponding settings of the submodels.

Target floating point math environment

RTI supports the **Target floating point math environment** option on the Real-Time Workshop Interface page of the Model Explorer.



With this option, you can specify the floating-point math library extensions that Real-Time Workshop uses for code generation. If you specify an RTI platform on Real-Time Workshop's General page, this property is set to the default value "ANSI-C". This applies to both the Configuration Preferences set and the model-specific configuration set.

The availability of math extensions is specific to the compiler's run-time libraries. In general, it is not recommended to use the ISO-C or GNU extensions. Thus, you should always specify ANSI-C if you do not have an explicit need to use ISO-C or GNU.

For RTI-MP, the settings specified for the main model are transferred to all submodels.

Shared utilities directory

The RTI and RTI-MP build process supports the shared utilities directory, which is generally used for model referencing. Although this MATLAB Compatibility Update does not support model referencing, you can use the shared utilities directory. For details, refer to the MATLAB documentation. You can use the shared utilities directory by selecting "Shared location" from the **Utility function generation** list on the Interface page of your model's configuration set. For RTI-MP, you have to specify this setting in the configuration set of the main model. It is transferred to all submodels automatically.

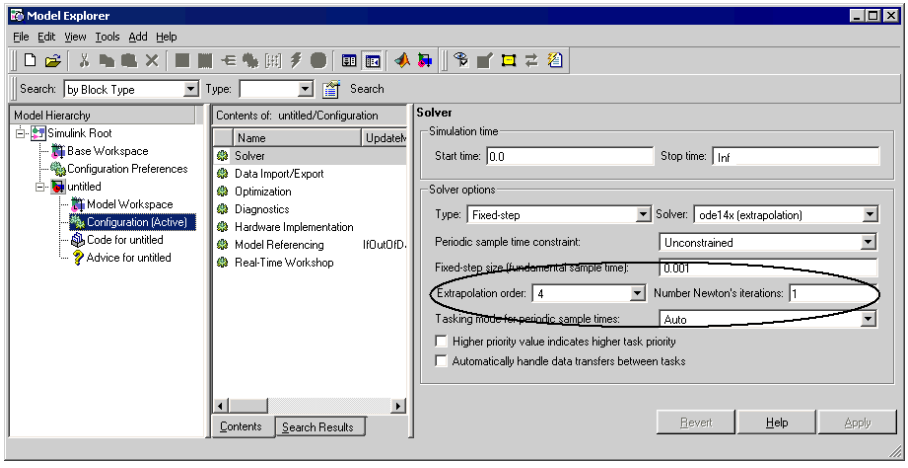
New Simulation Features

You should note the following changes in MATLAB R14 regarding the simulation of models.

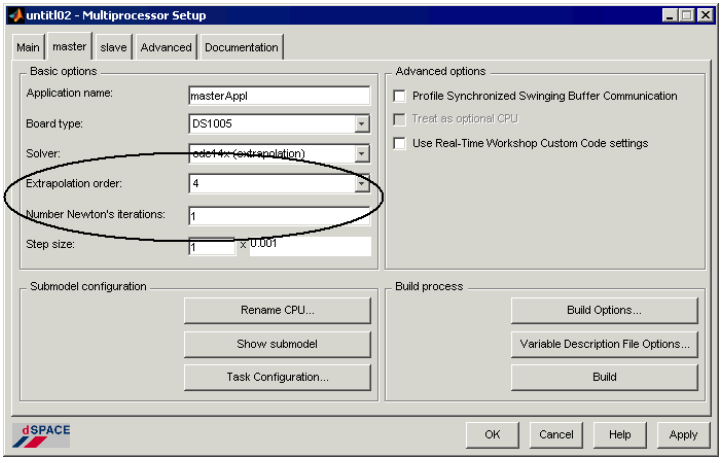
New ode14x solver

RTI and RTI-MP support the new implicit fixed-step continuous ode14x solver provided with MATLAB R14.

RTI and RTI-MP differ with regard to the dialog where the solver is specified. For RTI, you have to specify the Extrapolation order and the Number Newton's iterations on the Solver dialog of the Model Explorer.



For RTI-MP, you have to specify the Extrapolation order and the Number Newton's iterations on the CPU pages of the Multiprocessor Setup dialog.

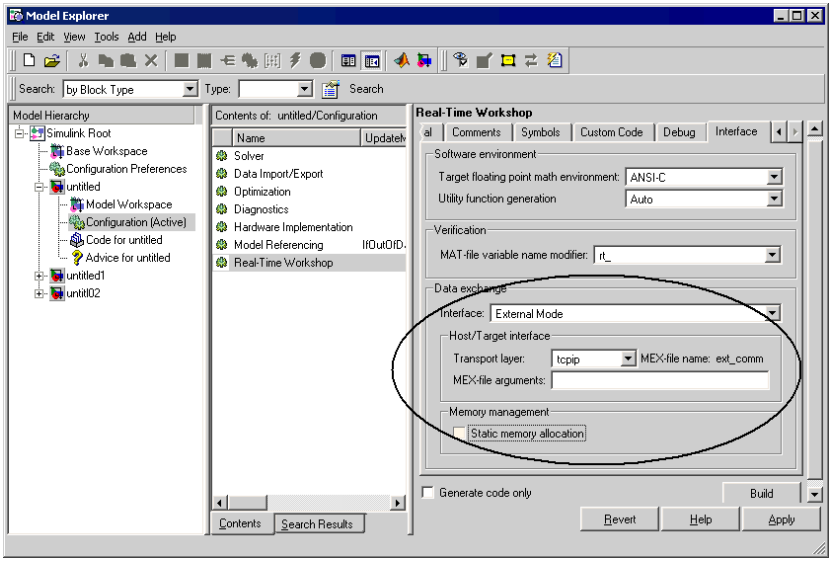


External simulation

To perform external simulation with the MATLAB R14 Compatibility Update, ensure that the following values are specified on the Interface page of the Real-Time Workshop dialog in the model configuration set:

Setting	Value
Interface	External mode
Transport layer	(Any value)
MEX file arguments	(Leave empty)
Static memory allocation	(Any value)

The following illustration shows a valid specification of the above settings for external simulation:



There are limitations applying to the support of external simulation. For details, refer to *Restricted support of external simulation* on page 82.

Limitations in the MATLAB R14 Compatibility Update for dSPACE Release 4.1

You should note some limitations when you migrate from dSPACE Release 4.1 to the MATLAB R14 Compatibility Update for dSPACE Release 4.1.

Unsupported features of MATLAB R14

There are some new features of MATLAB R14 which are not supported by this MATLAB Compatibility Update. Refer to *Unsupported Features of MATLAB R14* on page 67.

Software environment and compatibility

When you migrate to the MATLAB R14 Compatibility Update, there are some limitations in the software environment and compatibility. Refer to *Limitations in Software Environment and Compatibility* on page 69.

Modeling and configuration

There are some limitations when you create a model and configure it for RTI. Refer to *Limitations in Creating and Configuring Models* on page 70.

Handling tasks

Due to changes in MATLAB R14, there are some limitations in task handling. Refer to *Limitations in Task Handling* on page 72.

Building models

There are some limitations when you build and download your model. Refer to *Limitations in Building Models* on page 76.

Simulating models

When you use the MATLAB R14 Compatibility Update, support for external simulation is restricted. Refer to *Limitations in External Simulation* on page 82.

Miscellaneous limitations

In addition to the above limitations, there are miscellaneous restrictions you have to consider when you migrate to the MATLAB R14 Compatibility Update. Refer to *Miscellaneous Limitations* on page 84.

Unsupported Features of MATLAB R14

There is no support, or only very restricted support, for the following new features of MATLAB R14:

- Model referencing
- Model Workspaces
- New task features of MATLAB R14

Model referencing is not supported

With Simulink in MATLAB R14, you can include models in other models as blocks, which is called model referencing. The MATLAB R14 Compatibility Update does not support the model referencing feature and features based on it. For example, incremental code generation with Real-Time Workshop is also not supported.

Restricted support of Model Workspaces

MATLAB R14 provides Model Workspaces, which means that each model has its own workspace for storing variable values. For RTI, you have to take several limitations concerning Model Workspaces into account:

- RTI-MP does not support Model Workspaces.

  When the model is separated for several CPUs, the Model Workspaces are not transferred to the submodels.
- You cannot configure RTI blocks via MATLAB variables which are specified in Model Workspaces.

  These variables need to reside in the MATLAB workspace instead.
- You cannot make mask and workspace parameters available in the variable description file if the model uses Model Workspace variables.

  In this case, TRC file generation is aborted, and an error message is issued.

New task features of MATLAB R14 are not supported

RTI does not support the new task features of MATLAB R14. The following limitations apply:

- RTI does not make the task priorities known to Simulink and Real-Time Workshop.

- RTI does not support enhanced absolute and elapsed time computation.
- RTI does not support the following settings of the **Periodic sample time constraint** option on the Solver dialog:
  - Ensure sample time independent
  - Specified

  If you select one of the above settings, RTI issues an error message and aborts processing.

Limitations in Software Environment and Compatibility

The following limitations apply to the software environment and compatibility.

MATLAB R13.x is no longer supported

After you install the MATLAB R14 Compatibility Update for dSPACE Release 4.1, your dSPACE installation no longer supports MATLAB R13.x (R13.0.1, R13SP1 and R13SP1+). If you still need to use MATLAB R13.x, you must create another installation and switch between them using the dSPACE Installation Manager.

The MATLAB R14 Compatibility Update supports exclusively the following dSPACE products:

- dSPACE Release 4.1
- The DS1103 Revision 09 for dSPACE Release 4.1

Limitations in Creating and Configuring Models

The following limitations apply to modeling and configuration.

Inactive controls might be displayed as active

In some cases you might find that some controls previously disabled in the Model Explorer or in the Configuration Parameters dialog are enabled even if they should not be. This can also affect RTI options and is due to a problem in Simulink and Real-Time Workshop. For example, this might occur if:

- A build process fails in Real-Time Workshop
- You use the Revert button of the Model Explorer

Note that due to build checks of RTI and Real-Time Workshop, it is ensured that models are not built with inconsistent settings.

Display errors in RTI dialogs

Due to a problem in MATLAB R14, the following display errors can occur in RTI dialogs which are implemented as MATLAB Handle Graphics dialogs:

- With Windows XP, disabled edit fields look enabled.
- With Windows XP, the background color of dialogs is incorrect.
- Control names appear slightly outside of frames.
- Checkboxes have no right border.
- Frames might be slightly displaced on dialog pages.
- Text might be truncated at the top.
- The alignment of controls might be slightly displaced.

You can avoid some problems that occur with Windows XP by selecting “Windows Classic” from the Theme list on Windows’ Display Properties dialog.

These display errors do not impact the RTI dialogs’ functionality.

Dialog layout

With MATLAB R14, it is not possible to influence the positions of dialog controls belonging to Real-Time Workshop custom targets like RTI. The resulting page layout for the RTI options in the Model Explorer and the Configuration Parameters dialog looks rather untidy, but this does not affect the functionality.

Migrating old RTI models of Texas Instruments DSP based targets

If an older model is configured for an RTI platform that is no longer supported by dSPACE Release 4.1 (that is, DS1003 and DS1102 models from dSPACE Release 3.5 or earlier), all the values of RTI-specific properties are lost when the model is migrated to the MATLAB R14 Compatibility Update. You must reconfigure the RTI options manually.

Limitations in Task Handling

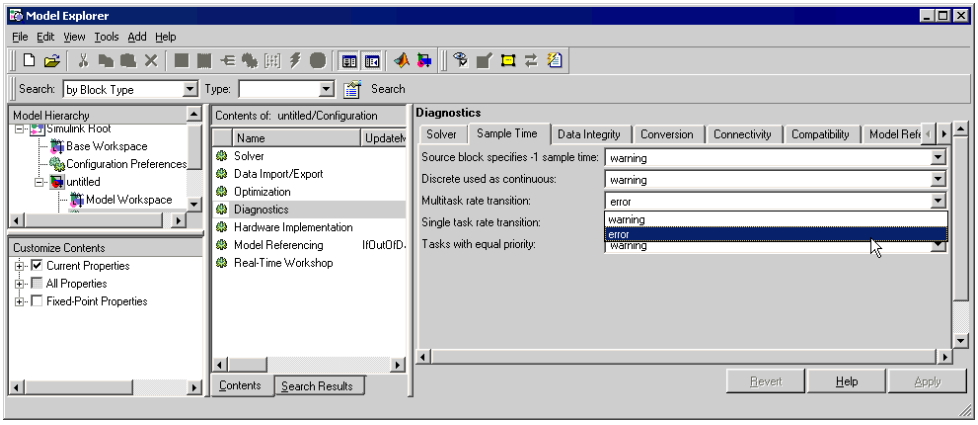
The following limitations apply to task handling.

Interrupt-driven subsystems that access time

An interrupt-driven subsystem which accesses the time reads the time value from the base rate task. In earlier MATLAB versions, this time value was represented by a variable of double type. However, in MATLAB R14, the time value is represented by two 32-bit counters. Thus, reading the time value can no longer be performed by atomic access but must be done in several steps (reading a variable of type double is not interruptible on all dSPACE platforms). If the interrupt-driven subsystem interrupts the base rate task while the latter updates its time value, the interrupt-driven subsystem might read inconsistent data. The same could happen if the base rate task interrupts the interrupt-driven task when reading the time. Since Simulink and Real-Time Workshop do not know the task priorities, they cannot generate code to ensure data integrity for reading the time value. For this reason, if your model contains such a constellation, Real-Time Workshop issues the following message:

```
Asynchronous task associated with the function-call subsystem
'MyModel/Function-Call Subsystem' accesses absolute time from
the fundamental sample time task without ensuring the integrity
of the value accessed. You can avoid this problem by assigning
a priority to this asynchronous task via the asynchronous S-
function 'untitled/Timer Interrupt/S-Function1'. If you need to
proceed without guaranteeing the integrity of the value used
for time, you may set the Diagnostics option 'Multitask rate
transition' to 'warning'.
```

It depends on the setting of the **Multitask rate transition** option on the Sample Time page whether the message is treated as an error or a warning.



Subsystems triggered by several interrupts

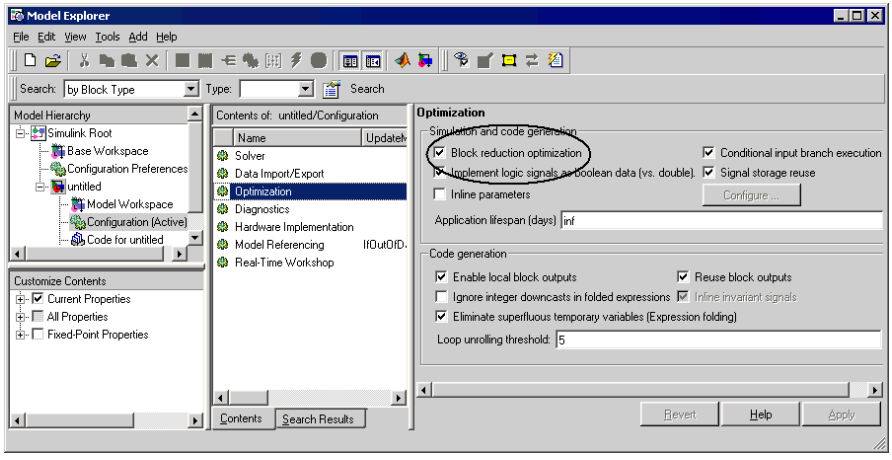
If a subsystem is triggered by several RTI interrupts (hardware, software, and/or timer interrupts), Simulink issues the following warning:

```
Warning: The function-call subsystem
'combined_interrupts/Subsystem' has multiple asynchronous
triggers that do not specify priority.  Data integrity will not
be maintained if these triggers can preempt one another.
```

Simulink always issues the above warning, because RTI does not make the task priorities known to Simulink. Refer to *New task features of MATLAB R14 are not supported* on page 67. In the case of such a modeling constellation, you must configure the priorities of the involved tasks to have the same value.

Using the Simulink Rate Transition block

In the case of data transfer between tasks, the Simulink Rate Transition block shows the behavior described below if the task priorities are equal or unknown to Simulink and Real-Time Workshop (as is currently the case with RTI, refer to *New task features of MATLAB R14 are not supported* on page 67). The behavior depends on the setting of the **Block reduction optimization** checkbox on the Optimization dialog:



- Block reduction optimization is disabled:

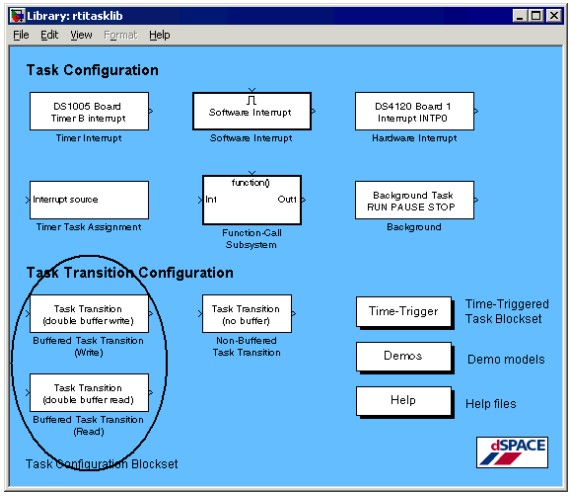
  The Simulink Rate Transition block switches to the **Copy** mode and generates the corresponding code for data transfer. However, the data transfer code generated by Real-Time Workshop ensures data integrity only if both tasks have the same priority, and cannot interrupt each other. If you use the Rate Transition block for data transfer from or to interrupt-driven tasks, you must configure the priorities of the involved tasks to fulfill this condition.

- Block reduction optimization is enabled:

  The Simulink Rate Transition block switches to the **NoOp** mode.

  Note that RTI does not support the Block reduction option. This mode can therefore be used only for Simulink simulations.

It is recommended to use the Simulink Rate Transition block only for data transfer between periodic tasks. For protected data transfer from, to, or between interrupt-driven tasks, it is recommended to use the rate transition blocks from the RTI TaskLib.



Limitations in Building Models

The following limitations apply to the build process.

Real-Time Workshop patch required for code generation

Due to problems in Real-Time Workshop, the code generated for your model might be incorrect and the real-time application might not work properly. The following problems are known:

- Code generated for blocks residing in a triggered subsystem on the topmost level of the model is duplicated into the code of all interrupt-driven subsystems of the model. The following RTI blocks are known to be affected by this problem:
  - Most of the RTI I/O blocks
  - The RTI Data Capture block from RTI's Extras Library
  - The Time-Trigger Set block from the RTI's TaskLib
- It can occur that Real-Time Workshop does not generate code for blocks inside an atomic subsystem. This might be influenced by the following parameters:
  - The setting of the **RTW system code** parameter of the atomic subsystem.
  - The atomic subsystem contains one or more sample times.
  - The atomic subsystem resides inside an interrupt-driven subsystem.

The MathWorks provides a patch that fixes most of these problems. You have to install the patch to work properly with RTI. You can download the patch under http://www.mathworks.com/support/solutions/data/1-OGY14.html?solution=1-OGY14.

Patch required for correct execution of periodic tasks

If a model contains several periodically executed tasks and at least one asynchronous task, the periodically executed tasks may be calculated with an incorrect sample time. This applies to tasks with the following sample times:

t_s > 255 * fixed-step size of the model

The MathWorks provides a patch that fixes this problem. You can download the patch under http://www.mathworks.com/support/solutions/data/1-P36G8.html?solution=1-P36G8.

Code generation problem for interrupt-driven subsystems

Due to a problem in Real-Time Workshop, a situation can arise in which the code generated for interrupt-driven subsystems would not be called in the real-time application. This happens only in some specific modeling cases. RTI checks if this problem occurs and aborts the build process with the following error message:

```
SUMMARY: Encountered separate Output and Update functions for
triggered subsystem

DETAILS: Due to an internal problem in Real-Time Workshop,
separate Output and Update functions are generated for the
triggered subsystem:

--> <subsystem_name>

The absence of a combined OutputUpdate function prevents RTI
from triggering the subsystem code by Hardware or Software
Interrupt blocks, or by the Background block. To work around
this problem try inserting unity Gain blocks after the inports
of the affected subsystem.
```

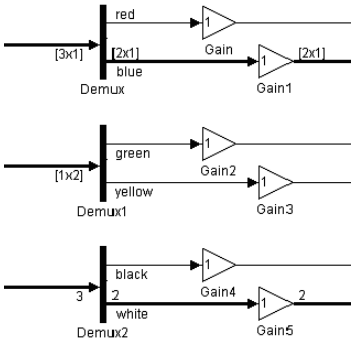
Block or subsystem names containing a "/"

Existing ControlDesk experiments may access block signals or parameters where the blocks or subsystems have a "/" in the model path they reside in. When you migrate to MATLAB R14, you have to recreate the connections to or from these blocks or subsystems if you want to access variables which reside in the path concerned. This limitation results from a problem in Simulink. It applies to MLIB or Python accordingly.

Dimension of Demux block output signals

Due to a problem in Real-Time Workshop, output signals from Demux blocks have incorrect dimensions in the variable description file (TRC file) if the input signal of the Demux block is a directed vector of the type 1xN or Nx1.

Consider the following example model:



The following table compares the dimensions of the above signals in the model to their dimensions in the variable description file (TRC file):

Signal Name	Dimension in the Model	Dimension in the TRC File
red	Scalar	[3x1] (incorrect)
blue	[2x1]	[3x1] (incorrect)
green/yellow	Scalar	[1x2] (incorrect)
black	Scalar	Scalar
white	Vector of length 2	Vector of length 2

The black and white signals have the correct dimensions in the variable description file (TRC file) because they are undirected vectors.

The following problems result from incorrect dimensions in the variable description file (TRC file):

- When you migrate to MATLAB R14, you may have to recreate some connections in ControlDesk layouts.

  This applies to accesses from MLIB or Python accordingly.
- Superfluous elements of the Demux block's output signals may point to data of other blocks or other memory units.

Vectorial block parameters

Due to an intentional change in Real-Time Workshop, vectorial block parameters with n elements (row or column vectors) are no longer interpreted as a matrix, but as a vector of length n. As a result, all dimension-specific data in the variable description file (TRC file) generated with MATLAB R14 differs from the data generated with MATLAB R13.

For example, if you want to access the m^{th} element of the myVec vector, you have to use myVec[m] instead of myVec[m][1].

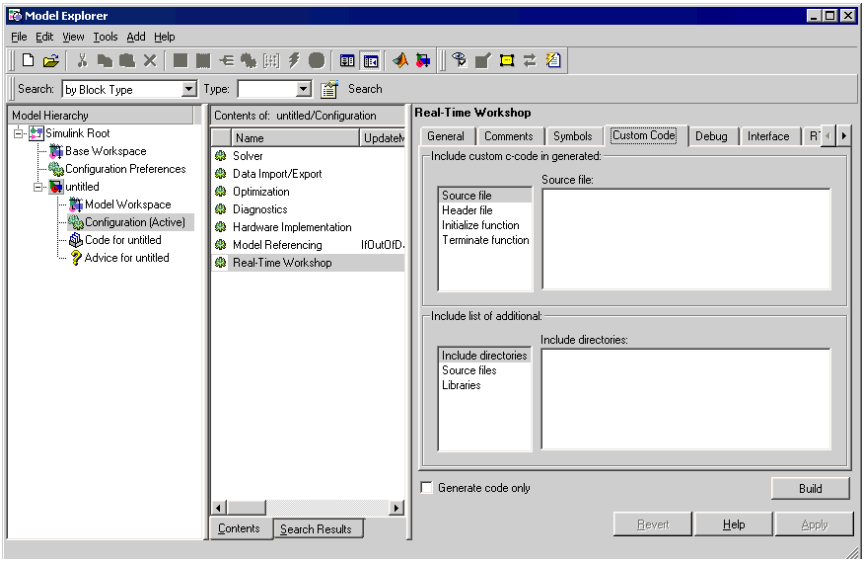
You may have to recreate all connections between vector elements to ControlDesk instruments in existing ControlDesk experiments. In a similar way, you may have to adapt MLIB and Python scripts.

This limitation applies only to parameters, not to signals.

Real-Time Workshop/ Stateflow Custom Code

Due to changes in MATLAB R14, the handling of custom code in Stateflow has changed.

With R13 and earlier versions of MATLAB, you can add custom code to a Stateflow chart directly in the so-called Real-Time Workshop target of the Stateflow chart. With MATLAB R14, you have to specify this custom code using the Custom Code settings of Real-Time Workshop:



If you open an existing model created with a Simulink version earlier than Simulink 6.0 (R14), Simulink automatically transfers the settings from the Stateflow chart to the Custom Code settings of Real-Time Workshop.

For RTI-MP, you must now use the **Use Real-Time Workshop Custom Code settings** checkbox on the CPU pages of the Multiprocessor Setup dialog to configure the CPUs for which the Real-Time Workshop custom code settings are used.

Compiler options containing assignments

Due to a problem in Real-Time Workshop, you cannot use compiler options which contain an assignment of the format `OPTION=value`. If you do, the build process is aborted and the following error message is displayed:

```
Error using ==> RTW.makertw.make_rtw

Operands to the || and && operators must be convertible to
logical scalar values.
```

SimMechanics Blockset is not supported for DS1006

The generated SimMechanics code uses functions from `stdio.h`, for example, `printf()`. The DS1006 compiler does not support these functions.

This limitation does not apply to PowerPC systems using the Microtec compiler (DS1005, DS1103, DS1104, DS1401).

Limitations in External Simulation

The following limitations apply to external simulation.

Restricted support of external simulation

Due to problems in Simulink and Real-Time Workshop, external simulation is subject to the following limitations:

- External simulation cannot be used with RTI-MP.
- For external simulations with RTI (single processor), you have to perform the following steps:

1. Remove the `no_ext_comm.dll` from the MATLAB memory by entering **`clear all`** or **`clear functions`** in the MATLAB Command Window.
2. Rename the `rti_comm.dll` in `no_ext_comm.dll`. The `rti_comm.dll` resides in the following path:

   `<DSPACE_ROOT>\MATLAB\RTI<xxxx>\M`

   If you start the external simulation without performing the above DLL renaming steps, Simulink issues the following error message:

   ```
   Error occurred while executing External Mode MEX-file
   'no_ext_comm': No external mode support for the current
   Real-Time Workshop target
   ```

- You cannot perform external simulation if your model contains Stateflow variables with the following characteristics:
  - The variables are not assigned to a specific Stateflow chart but to the model.
  - The scope of the variables is set to 'exported'.

  In this case, the build process aborts during code generation with the following error messages:

```
Error: File: D:\MATLAB\R14\rtw\c\tlc\mw\globalmaplib.tlc Line: 3949 Column: 18
Unable to find SL_ExtModeType within the Data scope
Error: File: D:\MATLAB\R14\rtw\c\tlc\mw\globalmaplib.tlc Line: 3949 Column: 36
The == and != operators can only be used to compare values of the same type
Error: File: D:\MATLAB\R14\rtw\c\tlc\mw\extmodemaplib.tlc Line: 115 Column: 18
Unable to find SL_ExtModeType within the Data scope
Error: File: D:\MATLAB\R14\rtw\c\tlc\mw\extmodemaplib.tlc Line: 115 Column: 36
The == and != operators can only be used to compare values of the same type
```

You cannot specify the RTI platform using the `BOARD=<board_name>` command. As a result, you can perform external simulation for ControlDesk's Working Board only.

Miscellaneous Limitations

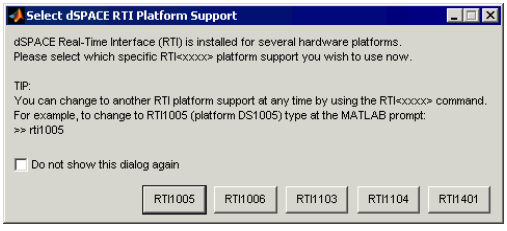
In addition to the above limitations, there are miscellaneous restrictions you have to consider when you migrate to the MATLAB R14 Compatibility Update.

Accessing RTI/RTI blocksets via MATLAB Start button

Due to a problem in MATLAB R14, RTI and the RTI blocksets may not be accessible via the MATLAB Start button (formerly known as Launch Pad in R13.x).

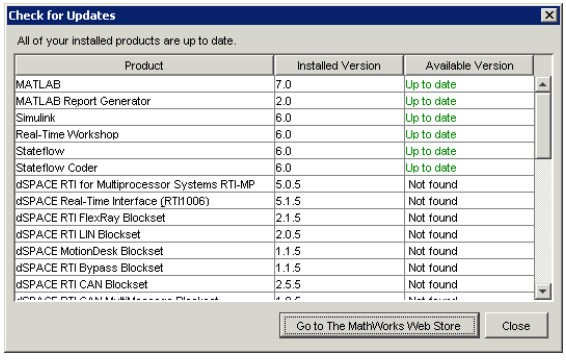
In the following cases, you can access RTI and the RTI blocksets via the MATLAB start button:

- After changing the RTI<xxxx> platform
- If several RTI<xxxx> platforms are installed and the following dialog opens at MATLAB start-up:



Check for Updates

MATLAB R14 allows you to check if more recent versions of The MathWorks products are available. The check for product updates also includes dSPACE products, for example, RTI, RTI blocksets, MLIB, etc. which are not products of The MathWorks. The result of the check for dSPACE product updates is therefore "not found". This is not a malfunction of your installation.



Model Advisor messages

With MATLAB's Model Advisor, you can analyze your model to configure Simulink and Real-Time Workshop appropriately. Since the Model Advisor is a tool from The MathWorks, it cannot take into account RTI specifics. Therefore, the Model Advisor may issue tips which are unsuitable or inapplicable for RTI. Double-check against the RTI documentation in case of doubt.

Error messages concerning S-function upgrades

If you select "error" or "warning" from the **S-function upgrades needed** list on the Compatibility page of the Diagnostics dialog, an error message or warning may be displayed for RTI blocks. You can ignore these messages.

It is recommended to select "none" or "warning" from the **S-function upgrades needed** list.

Key Features of the MATLAB R14SP1 Compatibility Update for dSPACE Release 4.1

The MATLAB R14SP1 Compatibility Update for dSPACE Release 4.1 enables you to use dSPACE Release 4.1 together with MATLAB, Simulink, Real-Time Workshop and further products of The MathWorks R14 with Service Pack 1. For simplicity, the term "MATLAB R14SP1" is used throughout the rest of this document. For details on MATLAB R14SP1, refer to the MATLAB R14SP1 documentation. The MATLAB R14SP1 Compatibility Update provides the same RTI functionality as the MATLAB R14 Compatibility Update, for example, support of the Model Explorer, Configuration Sets, etc.

This chapter and the chapter *Limitations in the MATLAB R14SP1 Compatibility Update for dSPACE Release 4.1* contain all information relevant for users who update directly from MATLAB R13.x to MATLAB R14SP1, without installing MATLAB R14. If you update from R13.x to R14SP1, you do not need to read the chapters about the MATLAB R14 Compatibility Update. If you update from R14 to R14SP1, the changes mentioned in the next paragraph might be of interest to you.

The following key features are new or have changed compared with the MATLAB R14 Compatibility Update:

- *dSPACE Release 4.1 support of MATLAB R14SP1* on page 90
- *Stateflow test point* on page 99
- *Code Reuse* on page 106

If you already know the MATLAB R14 Compatibility Update, and are now switching to MATLAB R14SP1, it might be sufficient for you to read only the above sections.

Note that RTI does not support all new MATLAB features. For details on the limitations that apply when working with the MATLAB R14SP1 Compatibility Update, refer to *Limitations in the MATLAB R14SP1 Compatibility Update for dSPACE Release 4.1* on page 113.

- Before you start working with the MATLAB R14SP1 Compatibility Update, you should install patches for MATLAB R14SP1. You can find information on the recommended patches on http://www.dspace.de/goto?MLCU_R14SP1.
- When you work with the MATLAB R14SP1 Compatibility Update for dSPACE Release 4.1, it is assumed that you have knowledge in handling RTI, and that you know the RTI and RTI-MP Implementation Guide and the RTI and RTI-MP Implementation Reference. It is also assumed that you know the MATLAB R14SP1 Release Notes for MATLAB, Simulink, and Real-Time Workshop

General changes

The MATLAB R14SP1 Compatibility Update has several new implementation features, enhancements and changes. Refer to *General Features, Enhancements and Changes* on page 90.

Creating models and configuring RTI

MATLAB R14SP1 provides several new features for model creation and parameterization. Refer to *New Features for Creating Models and Configuring RTI* on page 92.

Handling tasks

When you work with the MATLAB R14SP1 Compatibility Update, you should note some changes in task handling. Refer to *Changes in Task Handling* on page 100.

Building models

The MATLAB R14SP1 Compatibility Update supports some of the new features of MATLAB R14 for building models. Refer to *New Features for Building Models* on page 101.

Simulating models

You should note some changes in MATLAB R14SP1 regarding the simulation of models. Refer to *New Simulation Features* on page 108.

General Features, Enhancements and Changes

The MATLAB R14SP1 Compatibility Update provides the following general enhancements and changes.

dSPACE Release 4.1 support of MATLAB R14SP1

After you install the MATLAB R14SP1 Compatibility Update, dSPACE Release 4.1 supports MATLAB R14SP1. This includes all RTI products, ControlDesk and MLIB. All functions of the RTI products are available unless otherwise stated. For details on limitations, refer to *Limitations in the MATLAB R14SP1 Compatibility Update for dSPACE Release 4.1* on page 113.

The MATLAB R14SP1 Compatibility Update can be installed on one of the following dSPACE Release 4.1 installations:

- dSPACE Release 4.1

  If you update dSPACE Release 4.1 with the MATLAB R14SP1 Compatibility Update, the DS1103 Revision 09 for dSPACE Release 4.1 is supported.

  Do not install the DS1103 Revision 09 for dSPACE Release 4.1 again.

- dSPACE Release 4.1 updated with the DS1103 Revision 09 for dSPACE Release 4.1
- dSPACE Release 4.1 updated with the MATLAB R13SP2 Compatibility Update
- dSPACE Release 4.1 updated with the MATLAB R14 Compatibility Update

- After you install the MATLAB R14SP1 Compatibility Update for dSPACE Release 4.1, your dSPACE installation no longer supports MATLAB R13.x (R13.0.1, R13SP1, R13SP1+, and R13SP2) and MATLAB R14. If you still need to use MATLAB R13.x or R14, you must create another installation and switch between them using the dSPACE Installation Manager.

- MTest 1.1 is not supported by the MATLAB R14SP1 Compatibility Update. It is disabled during the installation of the MATLAB R14SP1 Compatibility Update.

Compatibility of RTI models

RTI models created with dSPACE Release 4.1 or earlier provide full functionality when you work with the MATLAB R14SP1 Compatibility Update. Nevertheless, you should be aware of the limitations described in *Limitations in the MATLAB R14SP1 Compatibility Update for dSPACE Release 4.1* on page 113.

If you save an RTI model with the MATLAB R14SP1 Compatibility Update, you can no longer use it with earlier dSPACE releases.

RTI MAT file support

RTI saves MAT files with the -v6 option to enable users with a MATLAB R13.x version to work with the MAT files. This applies to

- RTI's DS2210 and DS2211APU blocksets
- The layouter in the RTI CAN MultiMessage Blockset's MainBlock

RTI FlexRay Blockset support

The MATLAB R14SP1 Compatibility Update supports the RTI FlexRay Blockset.

Support of the RTI FlexRay Blockset depends not only on MATLAB, but requires the DECOMSYS tools as well. Up to the time of the MATLAB R14SP1 Compatibility Update release, no DECOMSYS tools released for MATLAB R14SP1 were available. Contact DECOMSYS GmbH for appropriate versions of the DECOMSYS tools.

New Features for Creating Models and Configuring RTI

This MATLAB Compatibility Update supports many features introduced with MATLAB R14. The following new features for creating a Simulink model and configuring it for Simulink, Real-Time Workshop, and RTI are described in detail below:

- Model Explorer
- Configuration sets
- Signal & Scope Manager
- Duplicate input ports
- Environment Controller block

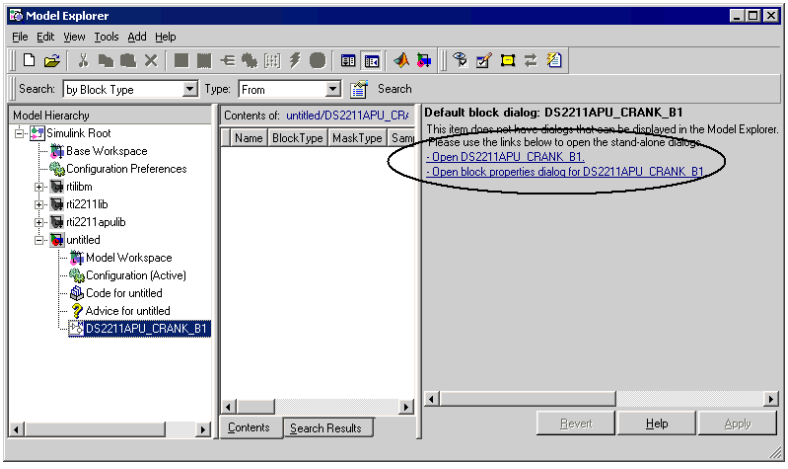
For detailed information on features which are not supported by this MATLAB Compatibility Update, refer to *Limitations in the MATLAB R14SP1 Compatibility Update for dSPACE Release 4.1* on page 113.

Model Explorer

Configuring RTI RTI supports the Model Explorer introduced with MATLAB R14. The Model Explorer allows you to locate and configure parameters in a Simulink model or a Stateflow chart. When you work with RTI, the Model Explorer helps you to perform the following actions:

- Specifying configuration preferences for RTI options
- Specifying RTI options for a model
- Configuring RTI blocks with a Simulink mask dialog

MATLAB provides a system of graphics objects for implementing visualization functions, for example, figure windows, axes, lines, etc., which is called MATLAB Handle Graphics. If an RTI block has a dialog implemented in MATLAB Handle Graphics, the Model Explorer displays a link for you to open the dialog with. The following illustration shows an example:



Configuring RTI-MP You can divide a large model into smaller parts and assign these parts to different CPUs. In this case, you have to use the Multiprocessor Setup dialog to configure the model for use on a multiprocessor system. In addition, you must configure the following options in the main model:

- All Simulink data import/export options
- All Simulink and Real-Time Workshop optimization options
- All Simulink diagnostic options
- All Simulink and Real-Time Workshop hardware implementation options
- The following Real-Time Workshop options:
  - The Documentation options on the General page
  - All options on the Comments page
  - All options on the Symbol page

- All options on the Custom Code page
- All options on the Debug page
- All options on the Interface page
- The **Generate code only** option of Real-Time Workshop

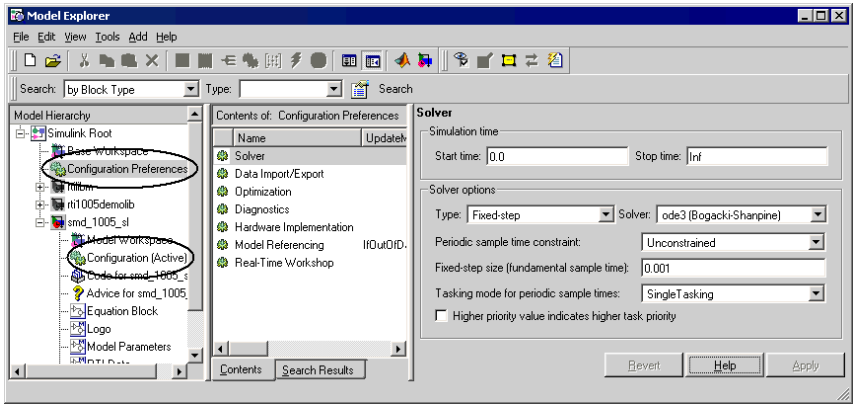
When the model is separated, RTI-MP ensures that the configuration of the main model is transferred to all submodels properly.

Model Referencing options are not supported. Refer to *Model referencing is not supported* on page 115.

Configuration sets

A configuration set is a set of values for your model's configuration parameters. You can create several configuration sets containing different values for any model configuration parameter, and associate them with the model. You can activate one of the configuration sets. The active set then contains the current values of the model's configuration parameters. There are two different types of configuration sets:

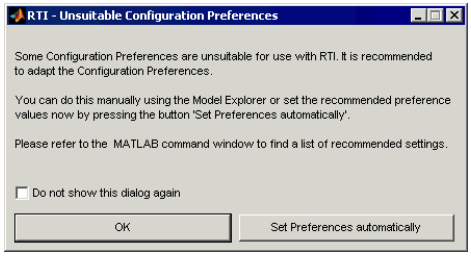
- The Configuration Preferences set
- Model-specific configuration sets



Configuration Preferences set In the Configuration Preferences set, you can specify default values for all model properties, which are transferred to the model's configuration set when you create a new model. When you start MATLAB or activate a different RTI platform, RTI checks if the settings in the Configuration Preferences set are suitable for RTI. The following table displays recommended settings for RTI-relevant properties:

Property	Value
Start time	0.0
Stop time	Inf
Solver type	Fixed-step
Periodic sample time constraint	Unconstrained
Fixed step-size	≠ auto
Higher priority value indicates higher task priority	Off
Block reduction optimization	Off
Signal storage reuse	Off
RTW system target file	rti<xxxx>.tlc (depending on the active target)

If the current values differ from the values which are recommended for RTI, a message dialog is displayed. You can set the recommended preferences automatically or edit the preferences manually after closing the message box.



If RTI detects unsuitable settings in your Preferences Configuration set, the current and the recommended settings are displayed in the MATLAB Command Window on MATLAB start-up, for example:

```
Some Configuration Preferences are unsuitable for use with RTI. It is recommended
to adapt the Configuration Preferences.

Note that, using the Model Explorer, you need to explicitly save manual changes to
the Configuration Preferences to restore them with the next MATLAB start.

  - Real-Time Workshop -            current setting            recommended setting
  RTW system target file:           'rti1104.tlc'          ->  'rti1005.tlc'
```

- The Configuration Preferences set is a Simulink feature and therefore holds the same values independently of the currently activated RTI platform.
- The RTI Task Configuration dialog lets you assign priorities to the different tasks of a model and configure the overrun strategy. You cannot open the RTI Task Configuration dialog for the Configuration Preferences set.

Model-specific configuration sets The model-specific configuration sets are saved together with the model file. In MATLAB R14SP1, you can specify different configuration sets for your model, for example, one for RTI1103, and one for RTI1005. In this case, for example, if you change from RTI1103 to RTI1005 platform support, you can simply activate the corresponding configuration set and continue working with the same model and without altering the configuration used for RTI1103.

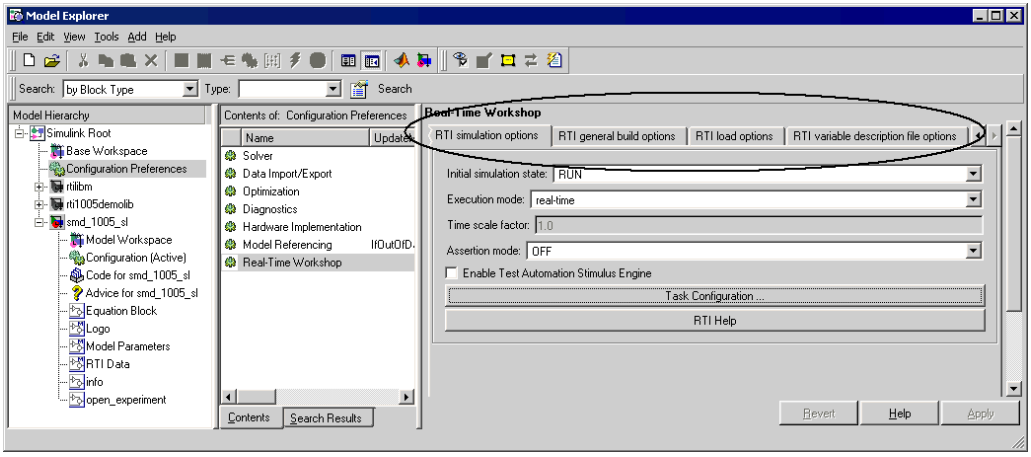
For RTI single-processor models, you can configure RTI options in active and inactive configuration sets for which you have specified an RTI<xxxx> target. Of course, you can also perform all standard Simulink operations on configuration sets, for example:

- Activating a specific configuration set
- Copying configuration sets within a model and to another model
- Copying the configuration preferences to a model

For single-processor models, you can specify the RTI options for the model-specific configuration either in the Model Explorer or in the Configuration Parameters dialog. Both methods lead to the same result.

To specify RTI options for multiprocessor systems, you have to use the Multiprocessor Setup dialog.

If you activate an RTI platform on the General page, for example, rti1005.tlc, the following RTI-specific pages appear in the Model Explorer's dialog:



- You can specify an RTI-specific configuration independently of the RTI platform that is currently activated. For example, while RTI1005 is the currently active platform support, you can create a new configuration set and configure it for RTI1103 (or any other RTI platform support) without the need to activate the RTI1103 platform support.
- Additionally, you can open a model that is configured for a specific RTI platform while another RTI platform is activated. For example, while RTI1005 is the currently active platform, you can open a model that is configured for RTI1103.

- You can start an RTI build process only if the RTI platform specified in the active configuration set is the same as the active platform. Otherwise, an error message is displayed.
- RTI task configuration is based on the activated RTI platform and the active configuration set of a model. Task configuration cannot be specified for each configuration set separately. Thus, RTI task configuration is the same for all RTI-specific configuration sets.
- You can open the RTI Task Configuration dialog only for an active configuration set and if the specified target is the same as the activated platform.

Target-specific default values If you change the RTI<xxxx> target in the Configuration Preferences set or in a model-specific configuration set, target-specific values are reset to their target-specific default values. All other values remain unchanged.

Signal & Scope Manager

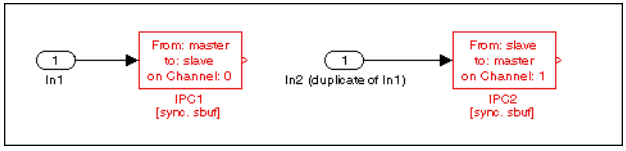
RTI fully supports the Signal & Scope Manager, introduced with MATLAB R14. It helps you manage signal generators and viewers.

Signal generators and viewers defined with the Signals & Scope Manager are not generated into the variable description file (TRC file).

Duplicate input ports

RTI and RTI-MP support duplicates of input ports in your model. The duplicates allow you to simplify block diagrams by eliminating unnecessary signal lines. In the variable description file (TRC file), they are displayed as common input ports. RTI-MP checks if the duplicate input ports are modeled correctly.

For example, specifying the same input signal with duplicate input ports as coming from different CPUs is not permitted.



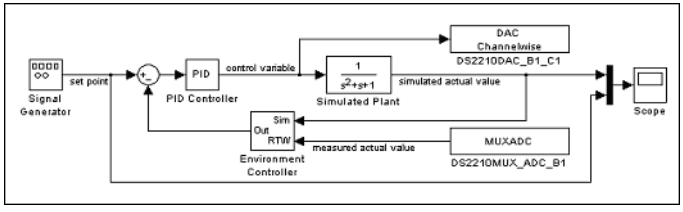
Environment Controller block

RTI supports the Environment Controller block. The Environment Controller block lets you create branches of a block diagram that apply only to simulation or only to code generation.

The Environment Controller block has the following inports:

- Sim
- RTW

If the model containing the block is simulated with Simulink, it outputs the signal at the Sim inport. If code is generated from the model, it outputs the signal at the RTW inport. The following illustration shows a simple example of how to use the Environment Controller block with RTI I/O blocks:



Stateflow test point

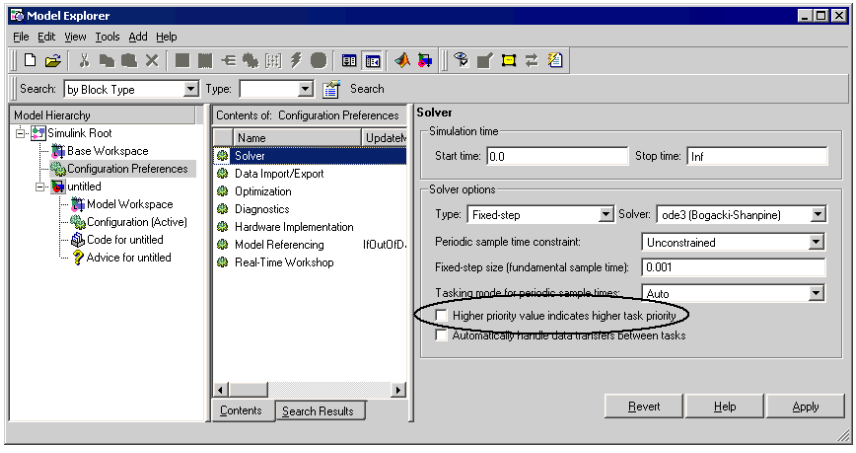
Stateflow test points are new with the MATLAB R14SP1 Compatibility Update. A Stateflow test point is a signal that is observable during simulation. The signal's observability is assured by Stateflow. RTI generates read-only entries in the appropriate Stateflow chart group in the variable description file (TRC file) for these signals.

Changes in Task Handling

When migrating to the MATLAB R14SP1 Compatibility Update, you have to note the following changes in task handling.

Handling task priorities

MATLAB R14SP1 provides the **Higher priority value indicates higher task priority** option on the Solver dialog of the Model Explorer.



For RTI, you must clear this checkbox. RTI checks for the correct setting of this checkbox when starting the build process.

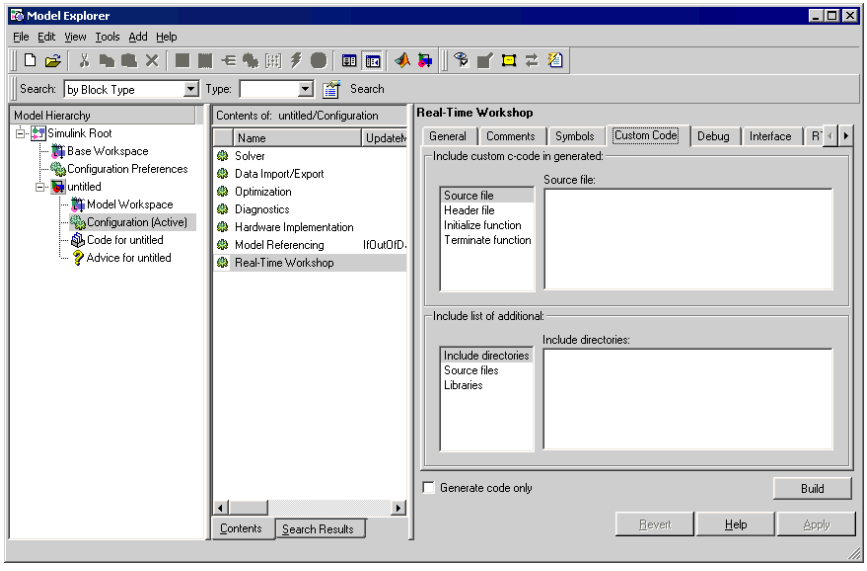
New Features for Building Models

The MATLAB R14SP1 Compatibility Update supports new features introduced with MATLAB R14 for building models. This section describes how to use the following new features with RTI and RTI-MP:

- Real-Time Workshop's Custom Code page
- Hardware Implementation dialog
- Target floating point math environment
- Shared utilities directory

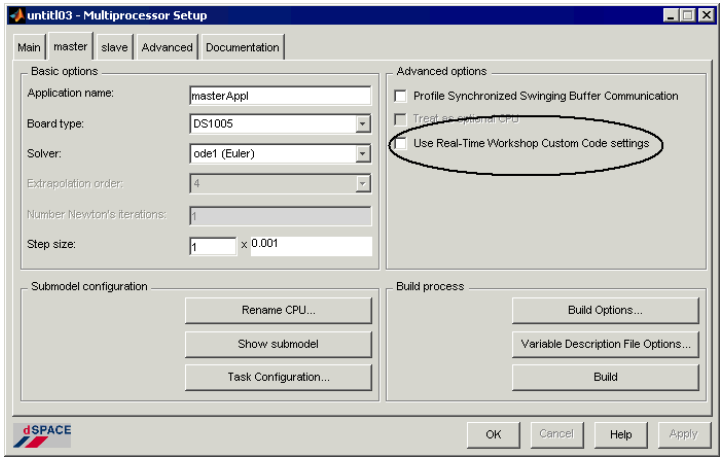
Custom Code page

RTI and RTI-MP support all the settings you can specify in Real-Time Workshop's Custom Code page.



- In earlier dSPACE releases, you could already use the RTI User-Code file (`USR.C`) to add hand-written C code to your real-time application. With this MATLAB Compatibility Update, you can use the settings in the **Include custom c-code in generated** frame in Real-Time Workshop's Custom Code page in addition to the settings in the RTI User-Code file (`USR.C`).
- In earlier dSPACE releases, you could already use the RTI User Makefile (`USR.MK`) to define extra search paths (for S-functions or user-defined C source code), user libraries, etc. With this MATLAB Compatibility Update, you can use the settings in the **Include list of additional** frame in Real-Time Workshop's Custom Code page instead of, or in addition to, the settings in the RTI User Makefile (`USR.MK`).

For RTI-MP, you have to specify the settings in Real-Time Workshop's Custom Code page of the main model. To have the settings transferred to a submodel, you have to select the **Use Real-Time Workshop Custom Code settings** checkbox on the relevant CPU pages of the Multiprocessor Setup dialog.

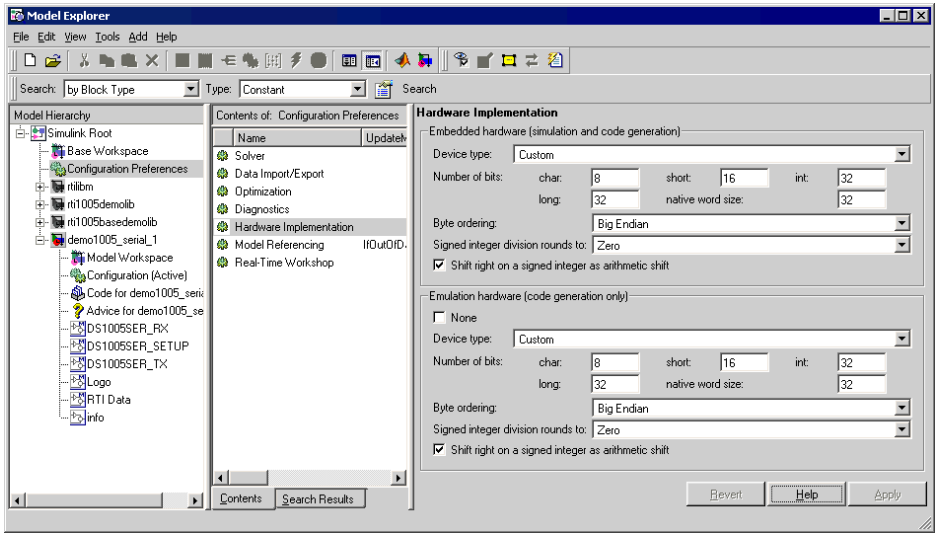


If you use Real-Time Workshop's Custom Code page, you cannot specify different custom code for each submodel of a multiprocessor system. If this is required, you must use the RTI User-Code files (`USR.C`) and RTI User Makefiles (`USR.MK`).

Hardware Implementation dialog

RTI supports the settings on the Hardware Implementation dialog, which lets you specify the characteristics of the hardware to be used to implement the system represented by your model. The Hardware Implementation dialog contains the following two groups of properties:

- **Embedded hardware (simulation and code generation)** lets you specify the properties of the production-type hardware.
- **Emulation hardware (code generation only)** lets you specify the properties of the hardware used to test the code generated from your model.



For details on these Simulink and Real-Time Workshop features, refer to the MATLAB R14 documentation.

Hardware Implementation configuration RTI and RTI-MP support code generation for the following configuration types:

- The RTI platform represents the production-type hardware, and no emulation hardware is used. You have to perform the following steps for configuration:
  - Under Embedded hardware, specify the properties applying to your dSPACE board.
  - Under Emulation hardware, select **None**.
- The behavior of arbitrary production-type hardware is emulated on the RTI platform. You have to perform the following steps for configuration:
  - Under Embedded hardware, specify arbitrary properties.
  - Under Emulation hardware, specify the properties applying to your dSPACE board.

    The behavior of the hardware specified under Embedded hardware is emulated on the RTI platform.

If you select an RTI platform on the General page of the Real-Time Workshop dialog, the following values are automatically configured on the Hardware Implementation dialog, depending on your dSPACE board:

Property	**DS1005, DS1103, DS1104, DS1401**	**DS1006**
Device type	Custom	Custom
Number of bits		
char	8	8
short	16	16
int	32	32
long	32	32
native word size	32	32
Byte ordering	Big Endian	Little Endian
Signed integer division rounds to	Zero	Zero
Shift right on a signed integer as arithmetic shift	True	True
Emulation hardware (code generation only)	None	None

RTI and RTI-MP check if the Hardware Implementation settings are specified correctly.

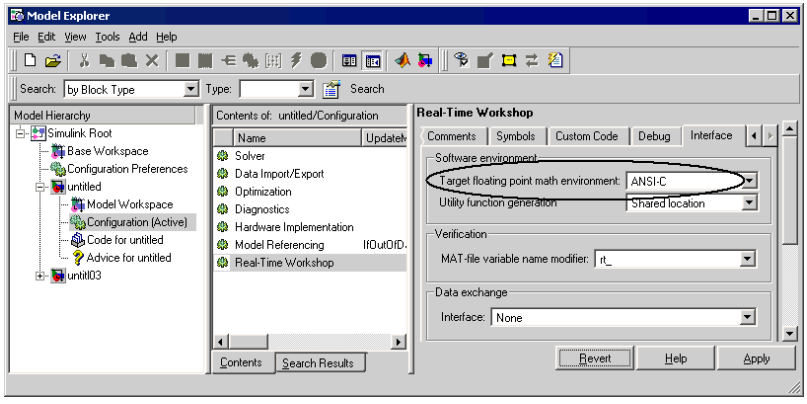
When you change the RTI system target file of your model to another RTI platform, for example, from rti1005.tlc to rti1103.tlc, all hardware characteristics are reset to the values specified in the above table.

Configuring hardware for RTI-MP When the model is separated, RTI-MP checks if the system target file specified in the main model's configuration set matches the dSPACE board type specified in the Multiprocessor Setup dialog. For example, if rti1005.tlc is specified in the configuration set, DS1005 must be specified in the Multiprocessor Setup dialog. If the settings do not match, model separation is aborted and an error message is displayed.

For RTI-MP, you cannot specify the Hardware Implementation properties separately for each submodel. When the model is separated for RTI-MP, the main model's Hardware Implementation settings are transferred to the corresponding settings of the submodels.

Target floating point math environment

RTI supports the **Target floating point math environment** option on the Real-Time Workshop Interface page of the Model Explorer.



With this option, you can specify the floating-point math library extensions that Real-Time Workshop uses for code generation. If you specify an RTI platform on Real-Time Workshop's General page, this property is set to the default value "ANSI-C". This applies to both the Configuration Preferences set and the model-specific configuration set.

The availability of math extensions is specific to the compiler's run-time libraries. In general, it is not recommended to use the ISO-C or GNU extensions. Thus, you should always specify ANSI-C if you do not have an explicit need to use ISO-C or GNU.

For RTI-MP, the settings specified for the main model are transferred to all submodels.

Shared utilities directory

The RTI and RTI-MP build process supports the shared utilities directory, which is generally used for model referencing. Although this MATLAB Compatibility Update does not support model referencing, you can use the shared utilities directory. For details, refer to the MATLAB documentation. You can use the shared utilities directory by selecting "Shared location" from the **Utility function generation** list on the Interface page of your model's configuration set. For RTI-MP, you have to specify this setting in the configuration set of the main model. It is transferred to all submodels automatically.

Code Reuse

RTI support for the Real-Time Workshop Code Reuse feature is new with the MATLAB R14SP1 Compatibility Update. By default, RTI now retains MATLAB's default setting (on) for the Real-Time Workshop Code Reuse feature. This is possible since the Code Reuse feature no longer conflicts with RTI's Include mask and workspace parameters option.

If required, you can disable the Code Reuse feature by typing the following command in the MATLAB Command Window:

```
rti_option EnableRTWCodeReuse 0
```

In some rare cases, Real-Time Workshop might generate wrong code in conjunction with the code reuse optimizations it performs. Check The MathWorks Web site or http://www.dspace.de/goto?MLCU_R14SP1 to see whether there is a patch available for this kind of problems.

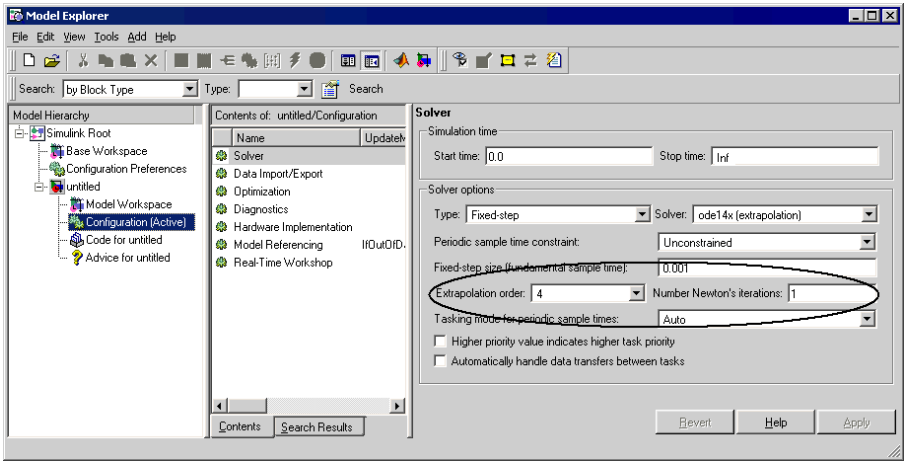
New Simulation Features

You should note the following changes regarding the simulation of models.

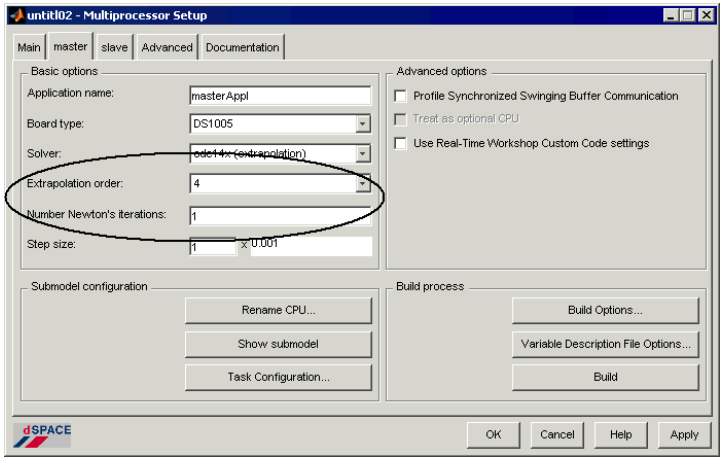
New ode14x solver

RTI and RTI-MP support the new implicit fixed-step continuous ode14x solver introduced with MATLAB R14.

RTI and RTI-MP differ with regard to the dialog where the solver is specified. For RTI, you have to specify the Extrapolation order and the Number Newton's iterations on the Solver dialog of the Model Explorer.



For RTI-MP, you have to specify the Extrapolation order and the Number Newton's iterations on the CPU pages of the Multiprocessor Setup dialog.



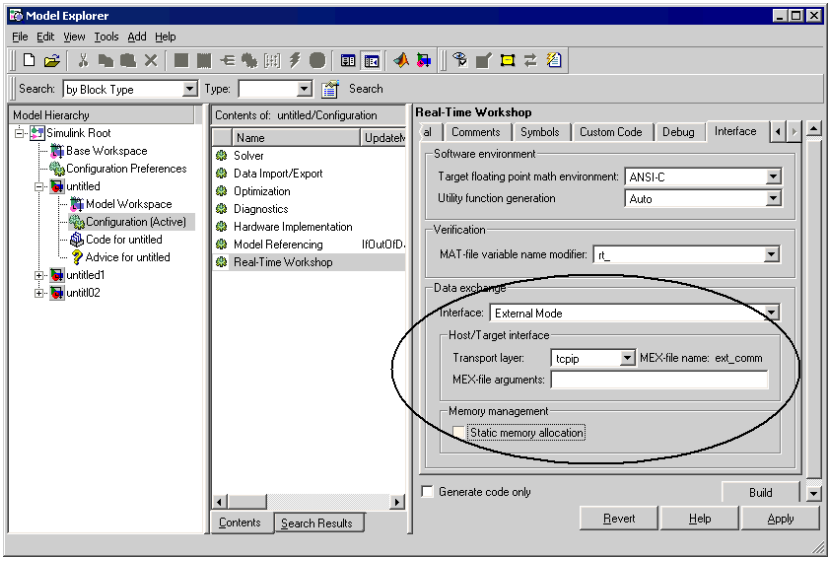
External simulation

To perform external simulation with the MATLAB R14SP1 Compatibility Update, ensure that the following values are specified on the Interface page of the Real-Time Workshop dialog in the model configuration set:

Setting	Value
Interface	External mode
Transport layer	(Any value)

Setting	Value
MEX file arguments	The MEX file arguments are optional and let you specify the desired target: • If left blank (default), RTI's external mode interface connects to the working board that was defined in ControlDesk. • If a board is specified using the syntax: 'BOARD=<board_name>' and this board is available in the connection defined by the working board (bus/network), this board is used as the target. MEX file arguments have to be enclosed by single quotes and separated by a comma.
Static memory allocation	(Any value)

The following illustration shows a valid specification of the above settings for external simulation:



There are limitations applying to the support of external simulation. For details, refer to *Restricted support of external simulation* on page 131.

Limitations in the MATLAB R14SP1 Compatibility Update for dSPACE Release 4.1

You should note some limitations when you migrate from dSPACE Release 4.1 to the MATLAB R14SP1 Compatibility Update for dSPACE Release 4.1.

Unsupported features of MATLAB R14SP1

There are some new features introduced with MATLAB R14 which are not supported by this MATLAB Compatibility Update. Refer to *Unsupported Features of MATLAB R14SP1* on page 115.

Software environment and compatibility

When you migrate to the MATLAB R14SP1 Compatibility Update, there are some limitations in the software environment and compatibility. Refer to *Limitations in Software Environment and Compatibility* on page 117.

Modeling and configuration

There are some limitations when you create a model and configure it for RTI. Refer to *Limitations in Creating and Configuring Models* on page 118.

Handling tasks

Due to changes in MATLAB R14SP1, there are some limitations in task handling. Refer to *Limitations in Task Handling* on page 123.

Building models

There are some limitations when you build and download your model. Refer to *Limitations in Building Models* on page 128.

Simulating models

When you use the MATLAB R14SP1 Compatibility Update, support for external simulation is restricted. Refer to *Limitations in External Simulation* on page 131.

Miscellaneous limitations

In addition to the above limitations, there are miscellaneous limitations you have to consider when you migrate to the MATLAB R14SP1 Compatibility Update. Refer to *Miscellaneous Limitations* on page 133.

Unsupported Features of MATLAB R14SP1

There is no support, or only very restricted support, for the following features of MATLAB R14SP1:

- Model referencing
- Model Workspaces
- New task features introduced with MATLAB R14

Model referencing is not supported

With Simulink in MATLAB R14SP1, you can include models in other models as blocks, which is called model referencing. The MATLAB R14SP1 Compatibility Update does not support the model referencing feature and features based on it. For example, incremental code generation with Real-Time Workshop is also not supported.

Restricted support of Model Workspaces

MATLAB R14SP1 provides Model Workspaces, which means that each model has its own workspace for storing variable values. For RTI, you have to take several limitations concerning Model Workspaces into account:

- RTI-MP does not support Model Workspaces.

  When the model is separated for several CPUs, the Model Workspaces are not transferred to the submodels.
- You cannot configure RTI blocks via MATLAB variables which are specified in Model Workspaces.

  These variables need to reside in the MATLAB workspace instead.
- You cannot make mask and workspace parameters available in the variable description file if the model uses Model Workspace variables.

  In this case, TRC file generation is aborted, and an error message is issued.

New task features introduced with MATLAB R14 are not supported

RTI does not support the new task features introduced with MATLAB R14. The following limitations apply:

- RTI does not make the task priorities known to Simulink and Real-Time Workshop.

- RTI does not support enhanced absolute and elapsed time computation.
- RTI does not support the following settings of the **Periodic sample time constraint** option on the Solver dialog:
  - Ensure sample time independent
  - Specified

  If you select one of the above settings, RTI issues an error message and aborts processing.

Limitations in Software Environment and Compatibility

The following limitations apply to the software environment and compatibility.

MATLAB R13.x and R14 are no longer supported

After you install the MATLAB R14SP1 Compatibility Update for dSPACE Release 4.1, your dSPACE installation no longer supports MATLAB R13.x (R13.0.1, R13SP1, R13SP1+, and R13SP2) and MATLAB R14. If you still need to use MATLAB R13.x or R14, you must create another installation and switch between them using the dSPACE Installation Manager.

The MATLAB R14SP1 Compatibility Update can be installed on one of the following dSPACE Release 4.1 installations:

- dSPACE Release 4.1

  If you update dSPACE Release 4.1 with the MATLAB R14SP1 Compatibility Update, the DS1103 Revision 09 for dSPACE Release 4.1 is supported.

  Do not install the DS1103 Revision 09 for dSPACE Release 4.1 again.

- dSPACE Release 4.1 updated with the DS1103 Revision 09 for dSPACE Release 4.1
- dSPACE Release 4.1 updated with the MATLAB R13SP2 Compatibility Update
- dSPACE Release 4.1 updated with the MATLAB R14 Compatibility Update

MTest 1.1 is not supported by the MATLAB R14SP1 Compatibility Update. It is disabled during the installation of the MATLAB R14SP1 Compatibility Update.

Limitations in Creating and Configuring Models

The following limitations apply to modeling and configuration.

Missing controls and interchanged channel numbers

Due to a problem in MATLAB R14SP1, the following failures occur on some RTI dialog pages:

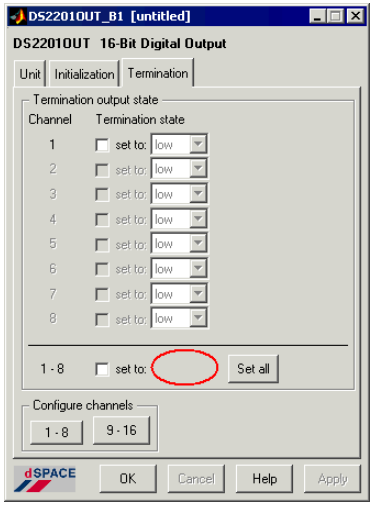
- Missing controls
- Interchanged channel numbers

The following table gives an overview of the relevant RTI blocks and dialog pages:

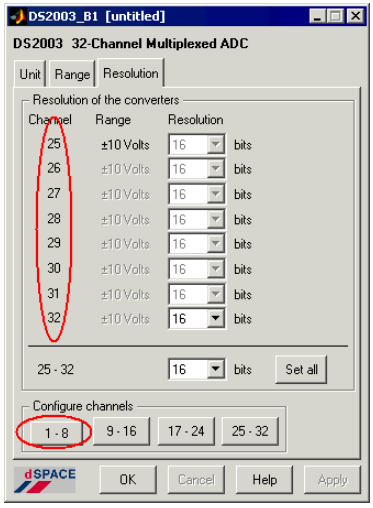
RTI Block	Dialog
DS2003_Bx	Range and Resolution pages
DS2103_Bx	Range, Initialization, and Termination pages
DS2201OUT_Bx	Initialization and Termination pages

The failures are explained in detail below.

Missing controls Edit fields or lists on the relevant pages might be missing. As a result, you cannot specify values for several channels in one step. You have to specify the values for each channel separately. The following illustration shows an example:



Interchanged channel numbers If you use the buttons in the **Configure channels** frame to switch between the channels (1 - 8, 9 - 16, etc.), the channels displayed in the **Resolution of the converters** frame might be incorrect. The following illustration shows an example:



This problem affects RTI's I/O code generation due to incorrect specification of the RTI block options. The following table gives an overview of the options which might be applied incorrectly for initializing the I/O hardware devices:

RTI Block	Option
DS2003_Bx	Options in the **Resolution of the converters** frame (on the Resolution page)
DS2103_Bx	Options in the **Initial voltage value** and **Termination voltage value** frames (on the Initialization page and the Termination page)
DS2201OUT_Bx	Options in the **Termination output state** frame (on the Termination page)

The code is generated correctly if you specify identical values for all channels.

Check The MathWorks web site or http://www.dspace.de/goto?MLCU_R14SP1 to see whether there is a patch available for this kind of problems.

Incorrect specification of RTI options

Due to problems in MATLAB R14SP1, incorrect code might be generated from the DS2002_Bx RTI block. This problem occurs if you specify different values for ADC1 and ADC2 on the block dialog's **Parameters** page.

Check The MathWorks web site or http://www.dspace.de/goto?MLCU_R14SP1 to see whether there is a patch available for this kind of problems.

Display errors in RTI dialogs

Due to a problem in MATLAB R14SP1, the following general and harmless display errors can occur in RTI dialogs which are implemented as MATLAB Handle Graphics dialogs:

- Control names appear slightly outside of frames.
- Checkboxes have no right border.
- Frames might be slightly displaced on dialog pages.
- Text might be truncated at the top.
- The alignment of controls might be slightly displaced.

These display errors do not impact the RTI dialogs' functionality.

Edit field changes can be lost

If you work with an RTI dialog which is implemented as a MATLAB Handle Graphics dialog and consists of several pages, the following problem can arise if you change the page:

If an edit field loses the focus because of a page change, the changes you made in that edit field are lost.

To avoid the above problem, you can click **Apply** before you change to another page. Alternatively, you can change to another control on the page the edit field resides in, or click an arbitrary point on that page, before you change to another page.

Check The MathWorks web site or http://www.dspace.de/goto?MLCU_R14SP1 to see whether there is a patch available for this kind of problems.

Dialog layout

With MATLAB R14SP1, it is not possible to influence the positions of dialog controls belonging to Real-Time Workshop custom targets like RTI. The resulting page layout for the RTI options in the Model Explorer and the Configuration Parameters dialog looks rather untidy, but this does not affect the functionality.

Migrating old RTI models of Texas Instruments DSP based targets

If an older model is configured for an RTI platform that is no longer supported by dSPACE Release 4.1 (that is, DS1003 and DS1102 models from dSPACE Release 3.5 or earlier), all the values of RTI-specific properties are lost when the model is migrated to the MATLAB R14SP1 Compatibility Update. You must reconfigure the RTI options manually.

Limitations in Task Handling

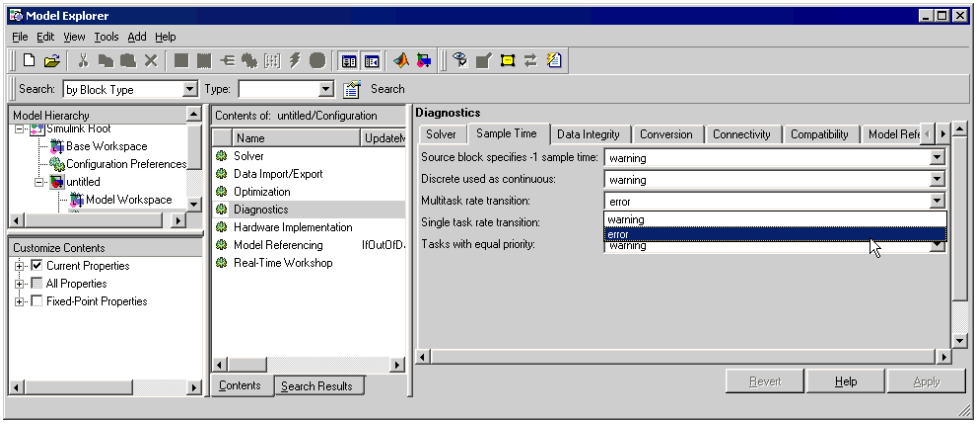
The following limitations apply to task handling.

Interrupt-driven subsystems that access time

An interrupt-driven subsystem which accesses the time reads the time value from the base rate task. In earlier MATLAB versions, this time value was represented by a variable of double type. However, in MATLAB R14SP1, the time value is represented by two 32-bit counters. Thus, reading the time value can no longer be performed by atomic access but must be done in several steps (reading a variable of type double is not interruptible on all dSPACE platforms). If the interrupt-driven subsystem interrupts the base rate task while the latter updates its time value, the interrupt-driven subsystem might read inconsistent data. The same could happen if the base rate task interrupts the interrupt-driven task when reading the time. Since Simulink and Real-Time Workshop do not know the task priorities, they cannot generate code to ensure data integrity for reading the time value. For this reason, if your model contains such a constellation, Real-Time Workshop issues the following message:

```
Asynchronous task associated with the function-call subsystem
'MyModel/Function-Call Subsystem' accesses absolute time from
the fundamental sample time task without ensuring the integrity
of the value accessed. You can avoid this problem by assigning
a priority to this asynchronous task via the asynchronous S-
function 'untitled/Timer Interrupt/S-Function1'. If you need to
proceed without guaranteeing the integrity of the value used
for time, you may set the Diagnostics option 'Multitask rate
transition' to 'warning'.
```

It depends on the setting of the **Multitask rate transition** option on the Sample Time page whether the message is treated as an error or a warning.



Subsystems triggered by several interrupts

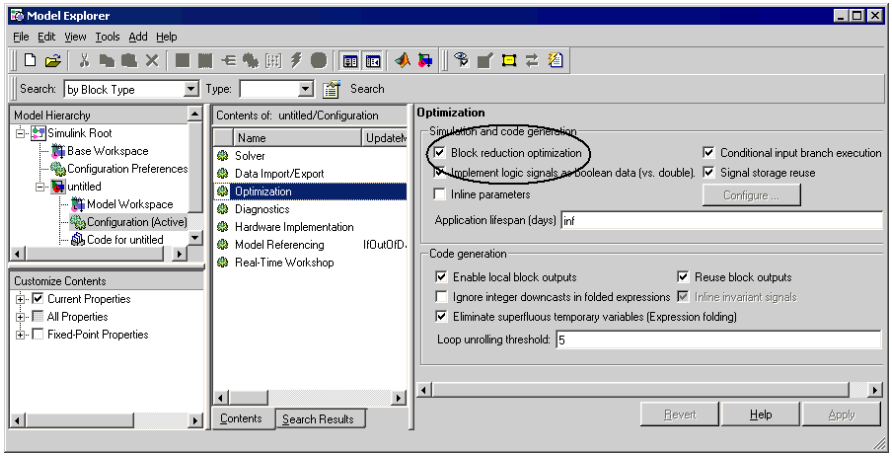
If a subsystem is triggered by several RTI interrupts (hardware, software, and/or timer interrupts), Simulink issues the following warning:

```
Warning: The function-call subsystem
'combined_interrupts/Subsystem' has multiple asynchronous
triggers that do not specify priority.  Data integrity will not
be maintained if these triggers can preempt one another.
```

Simulink always issues the above warning, because RTI does not make the task priorities known to Simulink. Refer to *New task features introduced with MATLAB R14 are not supported* on page 115. In the case of such a modeling constellation, you must configure the priorities of the involved tasks to have the same value.

Using the Simulink Rate Transition block

In the case of data transfer between tasks, the Simulink Rate Transition block shows the behavior described below if the task priorities are equal or unknown to Simulink and Real-Time Workshop (as is currently the case with RTI, refer to *New task features introduced with MATLAB R14 are not supported* on page 115). The behavior depends on the setting of the **Block reduction optimization** checkbox on the Optimization dialog:



- Block reduction optimization is disabled:

  The Simulink Rate Transition block switches to the **Copy** mode and generates the corresponding code for data transfer. However, the data transfer code generated by Real-Time Workshop ensures data integrity only if both tasks have the same priority, and cannot interrupt each other. If you use the Rate Transition block for data transfer from or to interrupt-driven tasks, you must configure the priorities of the involved tasks to fulfill this condition.

- Block reduction optimization is enabled:

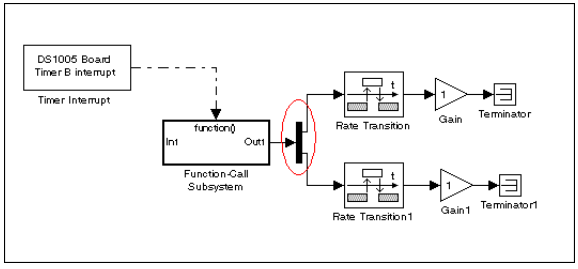
  The Simulink Rate Transition block switches to the **NoOp** mode.

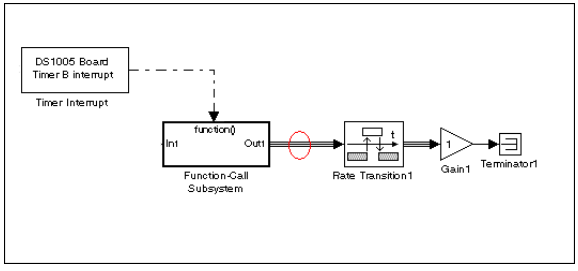
  Note that RTI does not support the Block reduction option. This mode can therefore be used only for Simulink simulations.

- If you use the Simulink Rate Transition block for data transfer from an interrupt-driven task to another task, a warning similar to the following may arise:

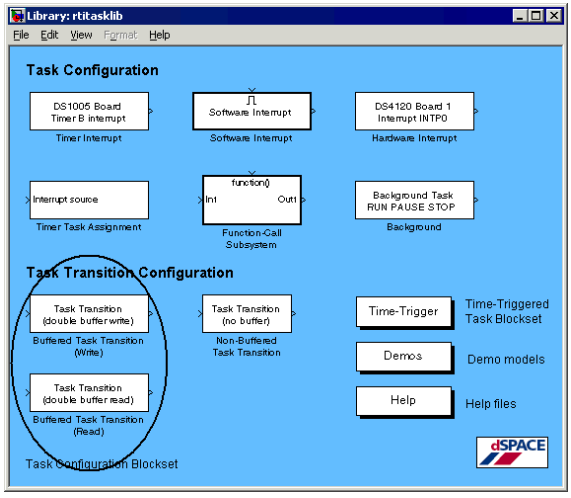
```
Warning: Inconsistent sample times. Sample time of signal
([-1, -1]) driving input port 1 of
'untitled/RateTransition1' differs from the expected sample
time at this input port ([-1, -2]).
```

Due to a bug in Simulink, the warning is displayed if a virtual block (which can be visible or inserted invisibly by Simulink) resides between an interrupt-driven subsystem and a Simulink Rate Transition block. Two examples are shown below. The first illustration shows a visible virtual block residing between an interrupt-driven subsystem and a Simulink Rate Transition block. The second illustration shows an invisible virtual block which is inserted automatically if a bus signal is used.





It is recommended to use the Simulink Rate Transition block only for data transfer between periodic tasks. For protected data transfer from, to, or between interrupt-driven tasks, it is recommended to use the rate transition blocks from the RTI TaskLib.



Limitations in Building Models

The following limitations apply to the build process.

Code generation problem for interrupt-driven subsystems

It can occur that Real-Time Workshop does not generate code for blocks inside an interrupt-driven subsystem. This problem has been observed in conjunction with RTI Software Interrupt blocks, but the possibility of it occuring in other modeling situations cannot be excluded. If you encounter this problem with the RTI Software Interrupt block, you have to make sure that a subsystem triggered by a Software Interrupt block resides outside the interrupt-driven subsystem containing the Software Interrupt block. If necessary, you have to change your model to meet this requirement.

Block or subsystem names containing a "/"

Existing ControlDesk experiments may access block signals or parameters where the blocks or subsystems have a "/" in the model path they reside in. When you migrate to MATLAB R14SP1, you possibly have to recreate the connections to or from these blocks or subsystems if you want to access variables which reside in the path concerned. This limitation results from a problem in Simulink. It applies to MLIB or Python accordingly.

This problem results from a bug in MATLAB R14 which is now fixed in MATLAB R14SP1. This limitation therefore applies only if you performed:

- Migration from the MATLAB R14 Compatibility Update and
- Experiment changes (due to the MATLAB R14 bug) during migration from MATLAB R13.x to MATLAB R14.

Vectorial block parameters

Due to an intentional change in Real-Time Workshop, vectorial block parameters with n elements (row or column vectors) are no longer interpreted as a matrix, but as a vector of length n. As a result, all dimension-specific data in the variable description file (TRC file) generated with MATLAB R14SP1 differs from the data generated with MATLAB R13.

For example, if you want to access the mth element of the myVec vector, you have to use myVec[m] instead of myVec[m][1].

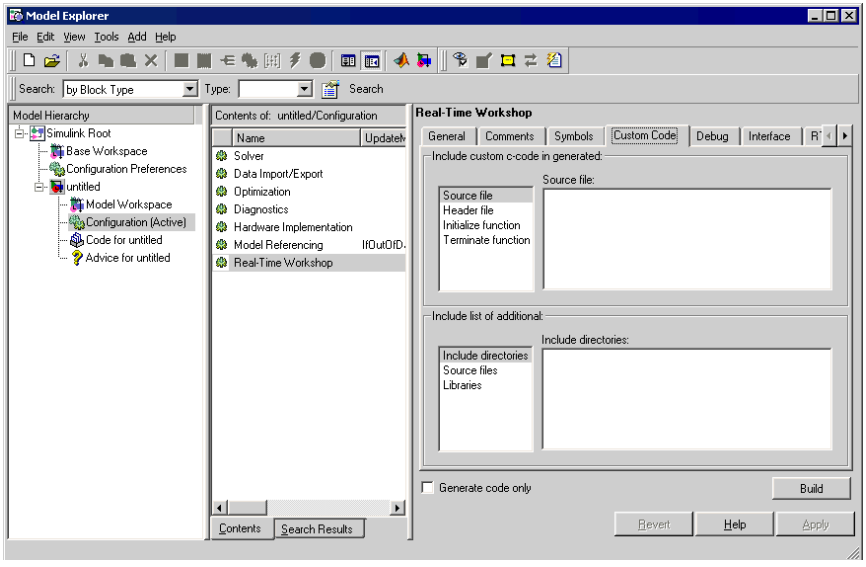
You may have to recreate all connections between vector elements to ControlDesk instruments in existing ControlDesk experiments. In a similar way, you may have to adapt MLIB and Python scripts.

This limitation applies only to parameters, not to signals.

Real-Time Workshop/ Stateflow Custom Code

Due to changes in MATLAB R14SP1, the handling of custom code in Stateflow has changed.

With R13 and earlier versions of MATLAB, you can add custom code to a Stateflow chart directly in the Real-Time Workshop target of the Stateflow chart. With MATLAB R14SP1, you have to specify custom code using the Custom Code settings of Real-Time Workshop:



If you open an existing model created with a Simulink version earlier than Simulink 6.0 (R14), Simulink automatically transfers the settings from the Stateflow chart to the Custom Code settings of Real-Time Workshop.

For RTI-MP, you must now use the **Use Real-Time Workshop Custom Code settings** checkbox on the CPU pages of the Multiprocessor Setup dialog to configure the CPUs for which the Real-Time Workshop custom code settings are used.

SimMechanics Blockset is not supported for DS1006

The generated SimMechanics code uses functions from `stdio.h`, for example, `printf()`. The DS1006 compiler does not support these functions.

This limitation does not apply to PowerPC systems using the Microtec compiler (DS1005, DS1103, DS1104, DS1401).

Limitations in External Simulation

The following limitations apply to external simulation.

Restricted support of external simulation

Due to problems in Simulink and Real-Time Workshop, external simulation is subject to the following limitations:

- Suppose you want to perform the following steps for a new model or a model saved with an older MATLAB version:

  1. Start the build process.
  2. Save the model.
  3. Close the model.
  4. Open the model.
  5. Start an external simulation.

  If you do, the following error message is displayed:

  ```
  Checksum mismatch. Target code needs to be rebuilt.
  ```

  To prevent the error message from being displayed, it is recommended to perform the following steps **before** the first build process:

  1. Save the model.
  2. Close the model.
  3. Open the model.

  If the above error message is displayed, you must restart the build process and save the model again.

  Check The MathWorks web site or http://www.dspace.de/goto?MLCU_R14SP1 to see whether there is a patch available for this kind of problems.

- External simulation can be used with RTI-MP only if the `EnableRTIMPExtSimWorkaround` RTI option is enabled. You can enable this option by entering the following command in the MATLAB Command Window:

  ```
  rti_option ('EnableRTIMPExtSimWorkaround', '1')
  ```

With this option enabled, each submodel is closed and opened again during the RTI-MP build process. As a result, the build process takes longer than it usually does.

If you do not enable this RTI option, the following error message is displayed when you start an external simulation for an RTI-MP submodel:

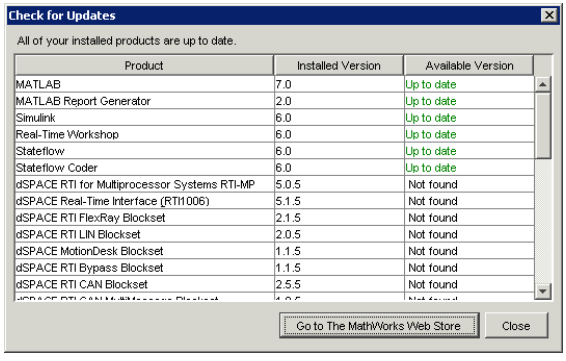
```
Checksum mismatch. Target code needs to be rebuilt.
```

Miscellaneous Limitations

In addition to the above limitations, there are miscellaneous limitations you have to consider when you migrate to the MATLAB R14SP1 Compatibility Update.

Check for Updates

MATLAB R14SP1 allows you to check if more recent versions of The MathWorks products are available. The check for product updates also includes dSPACE products, for example, RTI, RTI blocksets, MLIB, etc. which are not products of The MathWorks. The result of the check for dSPACE product updates is therefore "not found". This is not a malfunction of your installation.



Model Advisor messages

With MATLAB's Model Advisor, you can analyze your model to configure Simulink and Real-Time Workshop appropriately. Since the Model Advisor is a tool from The MathWorks, it cannot take into account RTI specifics. Therefore, the Model Advisor may issue tips which are unsuitable or inapplicable for RTI. Double-check against the RTI documentation in case of doubt.

Error messages concerning S-function upgrades

If you select "error" or "warning" from the **S-function upgrades needed** list on the Compatibility page of the Diagnostics dialog, an error message or warning may be displayed for RTI blocks. You can ignore these messages.

It is recommended to select “none” or “warning” from the **S-function upgrades needed** list.