dSPACE Release

New Features and Migration

Release 4.2
MTest 1.3 / AutomationDesk 1.2.1

September 2005

How to Contact dSPACE

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

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

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

© Copyright 2005 by:
dSPACE GmbH
Technologiepark 25
33100 Paderborn
Germany

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

ConfigurationDesk is a registered trademark of dSPACE GmbH in the United States or other countries, or both. Other brand names or product names are trademarks or registered trademarks of their respective companies or organizations.

Contents

About This Document

This document provides you with a brief overview of the major new features of dSPACE Release 4.2 since dSPACE Release 4.1. dSPACE Release 4.2 supports two MATLAB releases. If you install the MATLAB R14SP1 support, you have additional new features and limitations.

About dSPACE Releases

For information on how dSPACE software products are released, refer to *About dSPACE Releases* on page 7.

New features and enhancements of dSPACE Release 4.2

For a description of the key features, and a summary of the major enhancements made since dSPACE Release 4.1, refer to *Key Features of dSPACE Release 4.2* on page 9.

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 4.2. Refer to *Migrating to dSPACE Release 4.2* on page 27.

MATLAB R14SP1 Support

For a description of the key features and a summary of the major enhancements made for the MATLAB R14SP1 support in dSPACE Release 4.2, refer to *Key Features of the MATLAB R14SP1 Support in dSPACE Release 4.2* on page 31.

For details on the limitations you have to take into account when migrating to the MATLAB R14SP1 support in dSPACE Release 4.2, refer to *Limitations of the MATLAB R14SP1 Support in dSPACE Release 4.2* on page 57.

AutomationDesk 1.2.1 improvements

For a description of improvements made for AutomationDesk 1.2, refer to *Improvements in AutomationDesk 1.2.1* on page 81.

New Features of MTest 1.3

For a description of the key features and a summary of the major enhancements made for MTest 1.3, refer to *New Features of MTest 1.3* on page 83.

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).

A third digit is appended for a Maintenance Release , for example, dSPACE Release 4.0.1.

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.

Maintenance Release

A Maintenance Release is typically installed as an add-on to a dSPACE Release. The latest patches, Product Releases, and MATLAB Compatibility Updates are assembled in a Maintenance Release.

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 4.2

dSPACE Release 4.2 comes with the following key features:

Supported MATLAB releases

The software of dSPACE Release 4.2 offers full compatibility with the following releases of MATLAB from The MathWorks:

- MATLAB Release 13.0.1
- MATLAB Release 13 with Service Pack 1
- MATLAB Release 13 with Service Pack 2
- MATLAB Release 14 with Service Pack 1

It does not support MATLAB Release 14, MATLAB Release 13 (initial R13 release of August 2002), MATLAB R12.1 or earlier.

If you select the MATLAB R14SP1 support during installation, your dSPACE installation is compatible with MATLAB Release 14 with Service Pack 1. In this case, see also

- *Key Features of the MATLAB R14SP1 Support in dSPACE Release 4.2* on page 31

- *Limitations of the MATLAB R14SP1 Support in dSPACE Release 4.2* on page 57.

Supported operating systems

The software of dSPACE Release 4.2 supports Windows NT 4.0, Windows 2000, and Windows XP.

ConfigurationDesk and RTI Bypass Blockset 2.0 do not work with Windows NT 4.0.

New supported hardware

The following new hardware is supported by the software of dSPACE Release 4.2:

- RapidPro System

Refer to *Software Support of New RapidPro System* on page 13.

Implementation: key feature summary

The new key features of the implementation software are:

- The implementation software of the DS2211 was improved. The main improvements are:
  - Support of two event capture windows
  - Switching of DS2210 models to DS2211 models and vice versa
  - Flexible handling of engine variants
- Release 4.2 comes with the new RTI Bypass Blockset 2.0 and with RTI Bypass Blockset 1:
  - The new RTI Bypass Blockset 2.0 is a Simulink blockset for dialog-based configuration of bypass applications. It adds a software environment for service-based bypassing under Simulink to dSPACE Prototyper.

    The RTI Bypass Blockset 2.0 is a further development of previous RTI Bypass Blockset versions 1.x. It is not compatible with RTI Bypass Blockset versions 1.x.
  - RTI Bypass Blockset 1 (= RTI Bypass Blockset version 1.1.3) allows you to continue working with Simulink models built with RTI Bypass Blockset versions up to 1.1.2. It is provided for compatibility reasons only.

Refer to *New Implementation Features* on page 16.

Experiment and test: key feature summary

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

- Unicode support
  The experiment and test software supports Unicode, for example, for Japanese fonts and file paths. You can name items such as file paths, object data like instrument captions, comments, and strings in Python scripts in your native language. The menus and dialogs of the software are still in English. RTI does not support Unicode text, so the names of simulation models and their variables must be in English.
- AutomationDesk comes with:
  - A modified concept for report generation according to the features coming with the new Report library
  - New libraries for remote calibration via COM/DCOM interface, and access to the Diagnostic Toolset Version 7 from Softing GmbH
  - New data objects and automation blocks for the Platform Access library and the Main Library
  - The new General Properties dialog for specifying AutomationDesk settings, for example, the display mode of the Data Object Editor
  - Support of the multi-selection method for copy and paste actions
- MotionDesk and the MotionDesk Blockset come with:
  - The multitrack feature, which allows you to compare a simulation with recorded tracks synchronously and in real-time.
- ControlDesk comes with:
  - The new SteeringController instrument, which supports ForceFeedback game controllers. The instrument allows you to control your experiment with a game controller device such as a steering wheel or joystick. The instrument cannot be used under Windows NT.
  - Enhanced DS1006 support, which speeds up the downloading of real-time applications and the initialization of ControlDesk. With ControlDesk 2.6, you can update the firmware for DS1006 boards via a slot CPU.

Refer to *New Experiment and Test Features* on page 20.

Documentation: key feature summary

The documentation on installing and configuring boards has been reorganized. Refer to *New Documentation Features* on page 25.

Software Support of New RapidPro System

dSPACE Release 4.2 supports the new dSPACE RapidPro System.

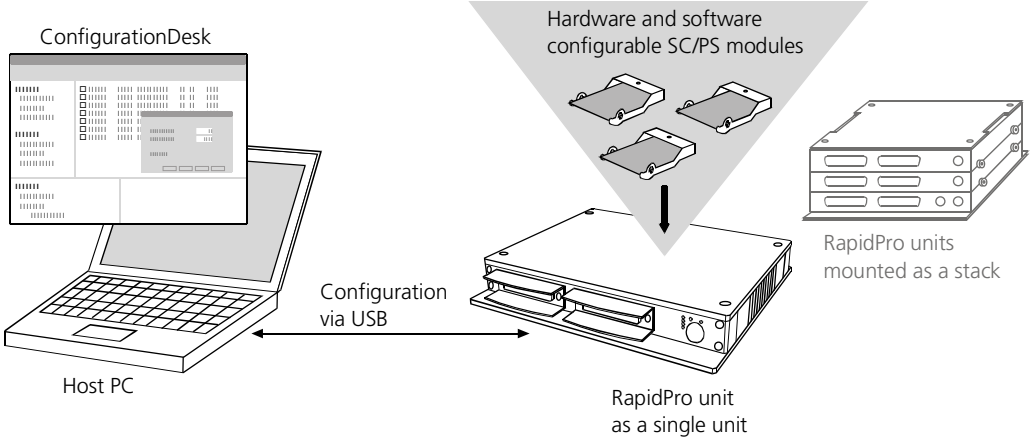
Use scenarios

The RapidPro System is modular and therefore very flexible. With different units and interchangeable modules, the system can be used in different applications:

- The RapidPro hardware adds solutions for signal conditioning and power stages to an existing dSPACE prototyping system.
- The RapidPro hardware is used as a slave I/O application of an existing rapid control prototyping (RCP) system in combination with signal conditioning and power stages.

System components

With three units and different interchangeable modules, the RapidPro System can be adapted to various requirements. You can easily configure various hardware parameters of the modules using the ConfigurationDesk software from dSPACE.



RapidPro units

The RapidPro hardware consists of three different unit types:

- RapidPro SC Unit (signal conditioning unit)
- RapidPro Power Unit (power stage unit)
- RapidPro Control Unit (intelligent I/O subsystem for dSPACE prototyping system)

With their compact and robust enclosures, the units are especially designed for in-vehicle use, but can also be used in laboratories and on test benches. The units are installed as single units (= separate units) or assembled as a stack of several units.

For more details on the unit types, refer to *Hardware Overview* in the *RapidPro System – Installation and Configuration Reference*.

SC/PS modules

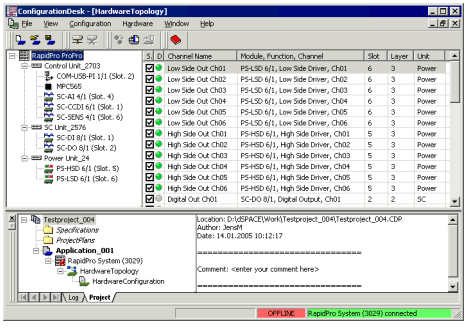
Hardware- and software-configurable signal conditioning (SC) and power stage (PS) modules can be mounted in the RapidPro units to set up individual systems. You can adapt hardware parameters to a wide range of applications with a minimum of effort.

For more details on the module types, refer to *Hardware Overview* in the *RapidPro System – Installation and Configuration Reference*.

ConfigurationDesk

ConfigurationDesk is a stand-alone Windows application that allows intuitive and efficient configuration of the RapidPro hardware.

It also provides detailed information on the hardware components installed in a RapidPro System (like modules used, hardware versions, serial numbers, and pinout information of the I/O connectors).



For further information, refer to *Introduction to ConfigurationDesk* in the *ConfigurationDesk – Configuration Guide*.

Implementation Software

For running models on the prototyping system, dSPACE provides Real-Time Interface (RTI) as a link between the hardware and the Simulink® development software from the MathWorks®. The following RTI blocksets are available for this:

- The RTI RapidPro Control Unit Blockset in combination with the Control Unit and a prototyping system (MicroAutoBox or DS1005-based) allows you to perform extensive standard I/O functionalities and special functionalities for engine, chassis and drives control.
  For further information, refer to *Introduction to Rapid Control Prototyping with the RapidPro System* in the *RapidPro Control Unit – RTI/RTLib Implementation Features*.
- The RTI RapidPro Diagnostics Blockset in combination with MicroAutoBox and a RapidPro Power Unit allows you to perform interactive diagnostics with feedback to the model.
  For more details on this specific blockset, refer to *RapidPro Diagnosis Blockset* in the *MicroAutoBox – RTI Reference*.

New Implementation Features

dSPACE Release 4.2 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 Implementation Features for Modular Hardware* on page 16
- *New Implementation Features for RTI CAN MultiMessage Blockset* on page 18
- *New RTI Bypass Blockset 2.0* on page 18

General Implementation Features, Enhancements and Changes

RTLib, RTI and RTI-MP of dSPACE Release 4.2 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 4.2 offers full compatibility with the following releases of MATLAB from The MathWorks:

- MATLAB Release 13.0.1
- MATLAB Release 13 with Service Pack 1
- MATLAB Release 13 with Service Pack 2
- MATLAB Release 14 with Service Pack 1

It does not support MATLAB Release 14, MATLAB Release 13 (initial R13 release of August 2002), MATLAB R12.1 or earlier.

New Implementation Features for Modular Hardware

dSPACE Release 4.2 provides the following new implementation features and enhancements for modular hardware:

DS1006

The load procedure was improved. A real-time application is loaded faster.

DS2211

The DS2211 RTI blockset and the DS2211 RTLib functions were improved:

Two event windows You can define one or two event capture windows for injection and ignition signal capturing.

Switching DS2210 <=> DS2211 You can convert DS2210 RTI blocks of a Simulink model to DS2211 RTI blocks and vice versa. Refer to *Converting Models for DS2210 to DS2211 and Vice Versa* in the *DS2211 Features* document.

Larger frequency range The upper frequency limit for the PWM and square-wave signal generation channels has increased to 100 kHz.

Flexible handling of engine variants Using new DS2211 RTI blocks, you can change the engine variant without the need to rebuild the real-time application. The relevant parameter can be changed via ControlDesk. Refer to *Angular Processing Unit - Variant* in the *DS2211 Features* document.

Complex comparator settings The parameter of the complex comparator is modifiable via model signals.

Default values for capture channel You can define default values for the capture channels.

Maximal pulse number increased The maximum number of pulses which can be captured has increased to 64.

DS4002

Two features of the DS4002 board are now supported by the DS4002 RTI blockset:

- You can generate monoflop signals using the *DS4002MONO_Bx_Cy* RTI block.
- You can measure the phase shift of a signal pair using the *DS4002PHASE_Bx_Cry_Cmz* RTI block.

Refer to *Timing I/O Unit* in the *DS4002 RTI Reference*.

New Implementation Features for RTI CAN MultiMessage Blockset

dSPACE Release 4.2 provides the following new implementation features and enhancements for the RTI CAN MultiMessage Blockset:

Free raw message You can process free raw messages independently of the DBC file. You can completely modify them, including their identifiers and message lengths.

Capture Message You can capture messages on the CAN bus whose identifiers match a customizable filter.

Manipulation of ID online You can modify the ID of a message either via TRC file or via inport. Modifying the IDs during run time is especially important when working with free raw messages.

Manipulation of MessageLength online You can modify the length of a TX message either via TRC file or via inport. Modifying the message lengths during run time is especially important when working with free raw messages.

Selection of raw data input or signal input online You can manipulate TX messages with raw data either via TRC file or via inport. You can switch between raw data and signals during run time, and also specify a default.

Online manipulation of GatewayBlock exclude list You can add or remove messages during run time.

Save settings for individual dialog pages You can save the settings of individual dialog pages to get granular backup of your work.

Customizable dialog tree You can remove unused dialog pages and structure pages according to your needs.

New RTI Bypass Blockset 2.0

Main features

dSPACE Release 4.2 provides the new RTI Bypass Blockset 2.0 with the following implementation features:

- Simulink blockset for dialog-based configuration of bypass applications

- Support of service-based bypassing based on the dSPACE Calibration and Bypassing Service and the dSPACE XCP Service
- Support of bypassing with service routines via DPMEM (dual-port memory) PODs and via XCP on CAN
- Easy selection of ECU variables via A2L file browser with sorting methods and search options
- Access of ECU variables by their names (automatic mapping of variables and their names to the corresponding addresses in the ECU memory, and handling of formulas for computing converted (physical) values to source (bin, dec, hex) values and vice versa)
- Bypass interface configuration via A2L (AML) file
- Specification of several database files (A2L files) for one bypass interface

Supported hardware

RTI Bypass Blockset 2.0 supports the following dSPACE hardware:

- MicroAutoBox
- Modular systems based on the DS1005 or DS1006 with DS4121 ECU Interface Board or DS4302 CAN Interface Board

- The new RTI Bypass Blockset 2.0 is a further development of the former RTI Bypass Blockset versions 1.x. It is not compatible with RTI Bypass Blockset versions 1.x. Thus, Simulink models which are built with an RTI Bypass Blockset 1.x version cannot be used with RTI Bypass Blockset 2.0.
  To continue working with Simulink models built with former RTI Bypass Blockset versions, you can work with RTI Bypass Blockset 1 (= RTI Bypass Blockset version 1.1.3), which is also provided by dSPACE Release 4.2.
- RTI Bypass Blockset 2.0 does not support Windows NT 4.0.

New Experiment and Test Features

dSPACE Release 4.2 has many new features and enhancements for dSPACE's experiment and test software.

New Features for ControlDesk

ControlDesk on dSPACE Release 4.2 comes with the following new features:

SteeringController instrument

With the SteeringController instrument, you can:

- Identify any type of game controller device attached to the system.
- Capture values from the device and transfer them to the connected platform.
- Read values from the platform and direct them to the game controller device to control the force feedback effects.

Refer to *SteeringController* in the *ControlDesk Instrument Reference*.

SelectionBox instrument

The SelectionBox instrument was improved:

- The new Incremental Search feature offers new ways of navigating through the text-value entries.
- You can now disable the display of the values of the text-value entries.

Refer to *SelectionBox* in the *ControlDesk Instrument Reference*.

Failure Simulation

The Failure Simulation component was improved:

- It supports the DS793 Sensor FIU. The DS793 is a failure insertion unit optional for dSPACE Simulator Mid-Size based on DS2211. It simulates electrical failures in the wiring harness on ECU inputs.
- You can activate combined FIU channels on several Mid-Size FIU boards simultaneously. To do this, you must modify the signal file. Refer to *Structure of Signal Files* in the *ControlDesk Experiment Guide*.

Unicode support

With ControlDesk 2.6 it is possible to work in non-Arabic operating system environments, for example, Japanese. You can name items such as file paths, object data like instrument captions, comments, and strings in Python scripts in your native language. The menus and dialogs of ControlDesk are still in English. RTI does not support Unicode text, so the names of simulation models and their variables must be in English.

DS1006 support

The DS1006 support was enhanced:

- Downloading real-time applications and initializating ControlDesk are faster now.
- With ControlDesk 2.6, you can update firmware for DS1006 boards via a slot CPU.

New Features for AutomationDesk

AutomationDesk on dSPACE Release 4.2 comes with the following new features:

Improved report generation

The report generation concept has been modified with regard to the features of the new Report library. The report content and layout was specified by the report settings, which are part of the general AutomationDesk settings. A report is now not only a formatted copy of the result but an individualized presentation of the executed automation task.

Data Container

The new **Data Container** can be used for grouping data objects. If your automation task requires a lot of data objects, for example, capture variables, a data container can help to organize the Project Manager more clearly. Transactions such as copy and move are now more convenient.

For further information, refer to *DataContainer* in the *AutomationDesk Library Reference*.

New libraries

AutomationDesk has new libraries:

- Report
  With the data objects and automation blocks of this library, you can add reporting blocks to your automation task. You can insert texts, images, 2-D plots, tables, URLs and data objects in your report.
  For further information, refer to *Report* in the *AutomationDesk Library Reference*.
- Remote Calibration (COM)
  With the data objects and automation blocks of this library, you can integrate a calibration tool, for example, CalDesk, into your automation task via the COM/DCOM interface. The measurement and calibration functions are based on the ASAM-MCD3 standard.
  For further information, refer to *RemoteCalibration (COM)* in the *AutomationDesk Library Reference*.
- DTS7
  With the data objects and automation blocks of this library, you can integrate the Diagnostic Toolset Version 7 from Softing GmbH into your automation task. The diagnostic functions are based on the ASAM-MCD3 standard.
  For further information, refer to *DTS7* in the *AutomationDesk Library Reference*.

Platform Access library

New automation blocks in the Platform Access library

- The automation blocks of the **Extended Capturing** folder allow a more flexible handling of capture tasks.
- The **AddCaptureToReport** block can be used for inserting a capture result in the report.

For further information, refer to *Platform Access* in the *AutomationDesk Library Reference*.

Further improvements

- Improved recognizability of data objects and automation blocks via significant symbols.
- Multiselection method now available for copy and paste actions.

New Features for MotionDesk

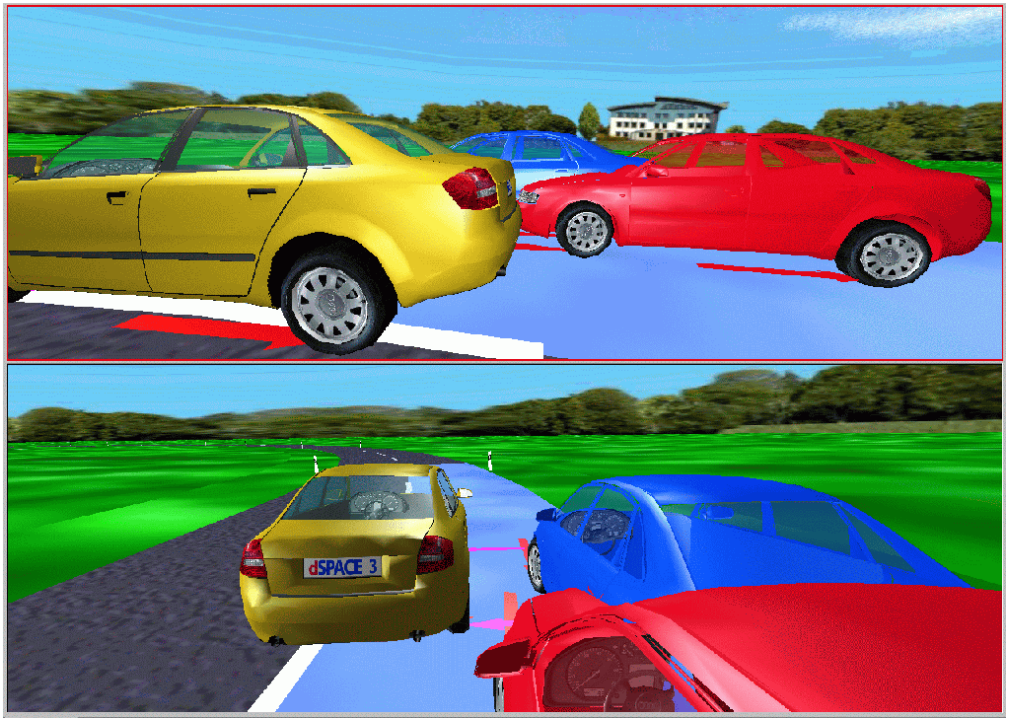
MotionDesk on dSPACE Release 4.2 comes with the following new features:

Multitrack mode

The multitrack feature lets you compare a simulation with recorded tracks and provides synchronous, real-time animation. The MotionDesk Blockset supports this feature. When working with the multitrack feature, you can:

- Configure the track properties, for example, to add a time offset for a recorded track.
- Assign different colors to movable objects of different tracks to distinguish them from each other.
- Hide movable objects, for example, to improve animation performance.

- Use the new lossless data acquisition to reach high quality, fine resolution, and lossless motion data.



For details, refer to *Comparing a Simulation With Recorded Tracks* in the *MotionDesk 3-D Online Visualization Guide*.

Slow/fast motion

In the Motion Player, you can now replay an animation up to 32 times slower or faster than normal.

New Documentation Features

With dSPACE Release 4.2, dSPACE reorganized the Installation and Configuration documentation for dSPACE systems (hardware and software). The "Configuring the System" and "Getting Started" chapters of all *Installation and Configuration Guides* are collected in a new document. The *dSPACE System First Working Steps* document guides you through your first working steps with a dSPACE system based on a DS1103, DS1104, DS1005, DS1006, or MicroAutoBox.

Migrating to dSPACE Release 4.2

After you install dSPACE Release 4.2, it may be necessary to carry out additional steps.

Migrating from dSPACE Release 4.1

Refer to *How to Migrate from dSPACE Release 4.1* on page 28.

Migrating from dSPACE Release 4.0 or earlier

Refer to *How to Migrate from dSPACE Release 4.0 or Earlier* on page 29.

How to Migrate from dSPACE Release 4.1

Note the following points when migrating from dSPACE Release 4.1 to dSPACE Release 4.2:

Implementation Software

Migrating to RTI Bypass Blockset 1

The former RTI Bypass Blockset is renamed RTI Bypass Blockset 1. Simulink models built with RTI Bypass Blockset version 1.1.2 or earlier can be reused with RTI Bypass Blockset 1.

Incompatibility of RTI Bypass Blockset and RTI Bypass Blockset 1

dSPACE Release 4.2 comes with the new RTI Bypass Blockset (version 2.0). This blockset is not compatible with RTI Bypass Blockset 1. It is not possible to migrate Simulink with RTI Bypass Blockset up to version 1.1.2 and models built with RTI Bypass Blockset 1 to the new RTI Bypass Blockset.

Experiment Software

CLIB

Since dSPACE Release 4.2, CLIB supports the Microsoft Visual C++ 6.0 and .NET compilers only. The support of earlier compilers was dropped.

How to Migrate from dSPACE Release 4.0 or Earlier

To migrate from dSPACE Release 4.0 or earlier to dSPACE Release 4.2, you have to migrate step by step via the intervening dSPACE Releases.

For example, if you want to migrate from dSPACE Release 3.5 to dSPACE Release 4.2, you have to follow the migration steps given in:

1. New Features and Migration of dSPACE Release 4.0
2. New Features and Migration of dSPACE Release 4.1
3. Finally, the migration steps given in *How to Migrate from dSPACE Release 4.1* on page 28.

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 `%dSPACE_ROOT\Doc\Print` folder). The PDF files are named `NewFeaturesAndMigrationxx.pdf`, where xx stands for the version or release number.

Key Features of the MATLAB R14SP1 Support in dSPACE Release 4.2

The MATLAB R14SP1 support in dSPACE Release 4.2 enables you to use dSPACE Release 4.2 together with MATLAB, Simulink, Real-Time Workshop and further products of The MathWorks R14 with Service Pack 1. For simplicity, the term "MATLAB R14SP1 support" is used throughout the rest of this document. For details on MATLAB R14SP1, refer to the MATLAB R14SP1 documentation.

Note that RTI does not support all new MATLAB R14SP1 features. For details on the limitations that apply to working with the MATLAB R14SP1 support, refer to *Limitations of the MATLAB R14SP1 Support in dSPACE Release 4.2* on page 57.

- Before you start working with the MATLAB R14SP1 support, you should install patches for MATLAB R14SP1. You can find information on the recommended patches on http://www.dspace.de/goto?Release42_R14SP1_Support.
- When you work with the MATLAB R14SP1 support in dSPACE Release 4.2, 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 support has several new implementation features, enhancements, and changes. Refer to *General Features, Enhancements and Changes* on page 33.

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 35.

Handling tasks

When you work with the MATLAB R14SP1 support, you should note some changes in task handling. Refer to *Changes in Task Handling* on page 43.

Building models

The MATLAB R14SP1 support allows you to work with some of the new features of MATLAB R14SP1 for building models. Refer to *New Features for Building Models* on page 45.

Simulating models

You should note some changes in MATLAB R14SP1 regarding the simulation of models. Refer to *New Simulation Features* on page 52.

General Features, Enhancements and Changes

The MATLAB R14SP1 support provides the following general enhancements and changes.

dSPACE Release 4.2 support in MATLAB R14SP1

Your dSPACE CD provides a version of dSPACE Release 4.2 which supports MATLAB R14SP1. This installation allows you to work with all RTI products, ControlDesk, and MLIB. All functions of the RTI products are available unless otherwise stated. For details on limitations, refer to *Limitations of the MATLAB R14SP1 Support in dSPACE Release 4.2* on page 57.

- If you install the MATLAB R14SP1 support in dSPACE Release 4.2, the installation does not support MATLAB R13.x (R13.0.1, R13SP1, R13SP1+, and R13SP2) and MATLAB R14. If you 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 support. It is disabled during the installation of the MATLAB R14SP1 support.

Compatibility of RTI models

RTI models created with the MATLAB R13.x support in dSPACE Release 4.2 or earlier provide full functionality when you work with the MATLAB R14SP1 support. Nevertheless, you should be aware of the limitations described in *Limitations of the MATLAB R14SP1 Support in dSPACE Release 4.2* on page 57.

If you save an RTI model with the MATLAB R14SP1 support in dSPACE Release 4.2, you can no longer use it with earlier dSPACE releases or with the MATLAB R13.x support in dSPACE Release 4.2.

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 support allows you to work with the RTI FlexRay Blockset.

Support of the RTI FlexRay Blockset not only depends on MATLAB, but also requires the DECOMSYS tools. Contact DECOMSYS GmbH for appropriate MATLAB R14SP1-compatible versions of the DECOMSYS tools.

New Features for Creating Models and Configuring RTI

This MATLAB R14SP1 support allows you to work with 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 port
- Environment Controller block

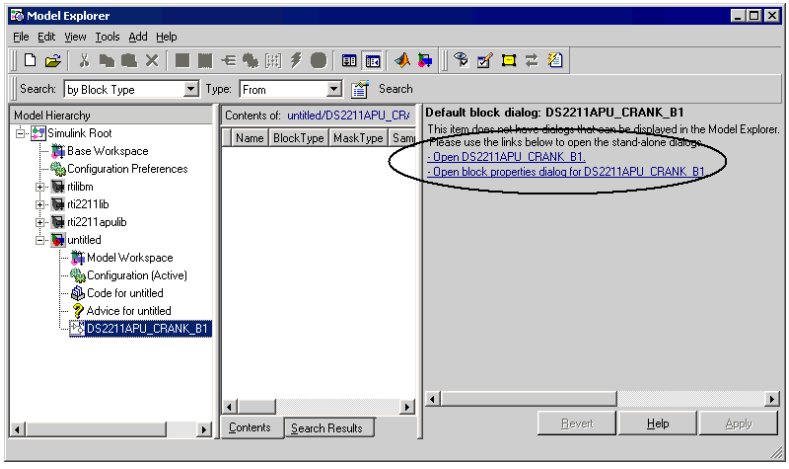
For detailed information on features which are not supported by this MATLAB support, refer to *Limitations of the MATLAB R14SP1 Support in dSPACE Release 4.2* on page 57.

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

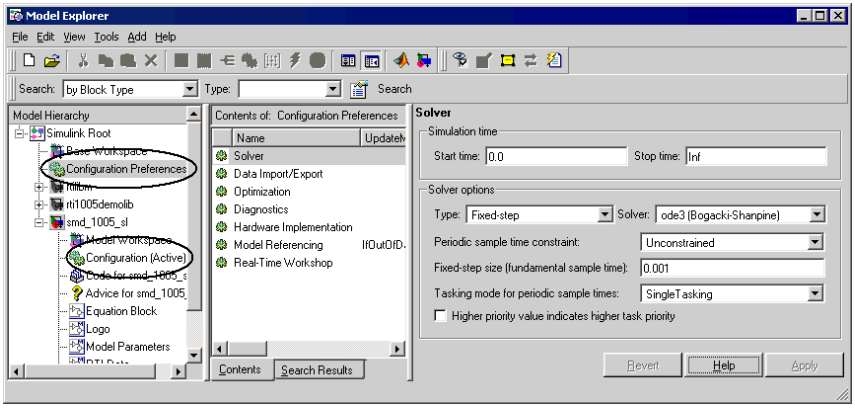
When the model is separated, RTI-MP ensures that the configuration of the main model is transferred to all submodels properly.

Model Referencing options are not supported. Refer to *Model referencing is not supported* on page 59.

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:

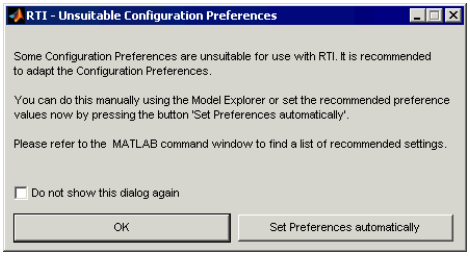
- The Configuration Preferences set
- Model-specific configuration sets



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.

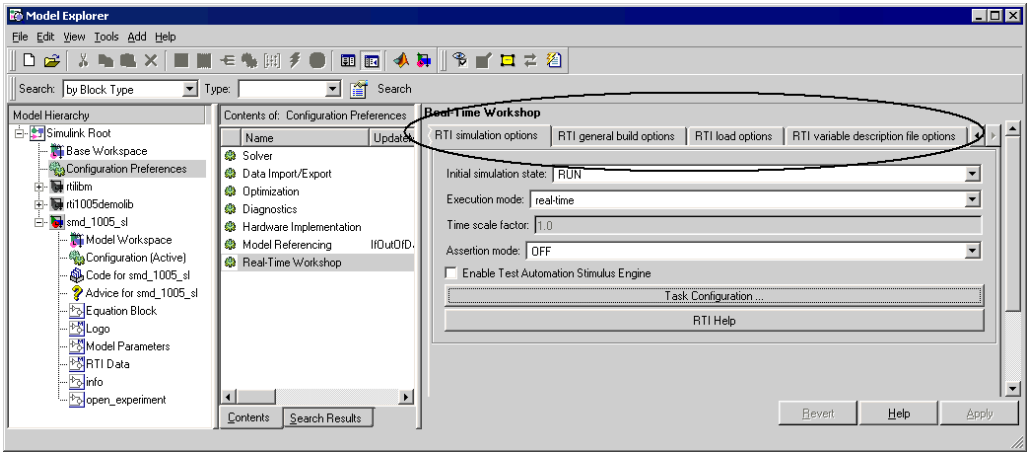
For RTI single-processor models, you can configure RTI options in active and inactive configuration sets for which you have specified an RTI<xxxx> target. Of course, you can also perform all standard Simulink operations on configuration sets, for example:

- 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

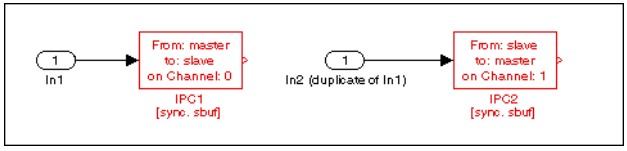
RTI fully supports the Signal & Scope Manager, introduced with MATLAB R14. It helps you manage signal generators and viewers.

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.



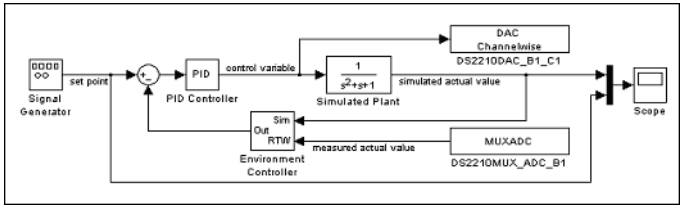
Environment Controller block

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

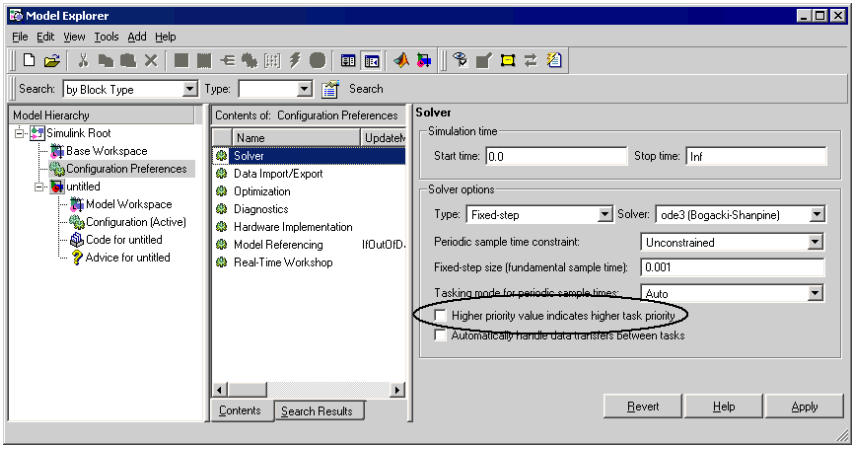
Stateflow test points are new with the MATLAB R14SP1 support. A Stateflow test point is a signal that is observable during simulation. The signal's observability is assured by Stateflow. RTI generates read-only entries in the appropriate Stateflow chart group in the variable description file (TRC file)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

Changes in Task Handling

When migrating to the MATLAB R14SP1 support, 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.

Enhanced time and elapsed time support

RTI lets you work with the enhanced time and elapsed time support in Real-Time Workshop. RTI supports Real-Time-Workshop time counter variables for all interrupt-driven tasks. RTI assigns values which are derived from the dSPACE Real-Time Kernel or the Time Stamping module to these variables. This ensures accurate time information with high resolution for interrupt-driven tasks.

- Suppose your model contains timer tasks and interrupt-driven tasks. Time t = 0 is defined as the point in time when the first timer task (Timer Task 1) is executed. Interrupt-driven tasks which were started by hardware interrupts before this point in time then also read time 0. This also applies to triggered subsystems which are connected to an RTI Background Code block.
- Numeric errors previously caused by interrupt-driven tasks reading the time variable value of Timer Task 1 no longer occur.
- The limitation 'Blocks depending on absolute time' stated in the *RTI and RTI-MP Implementation Guide* does not apply to the MATLAB R14SP1 support in dSPACE Release 4.2. For details on the limitation, refer to *General Limitations* in the *RTI and RTI-MP Implementation Guide*.

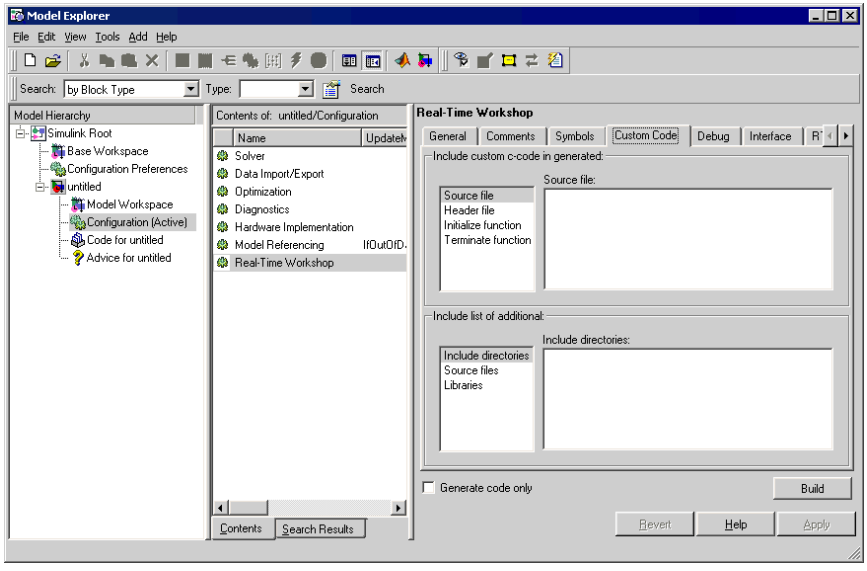
New Features for Building Models

The MATLAB R14SP1 support allows you to work with 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

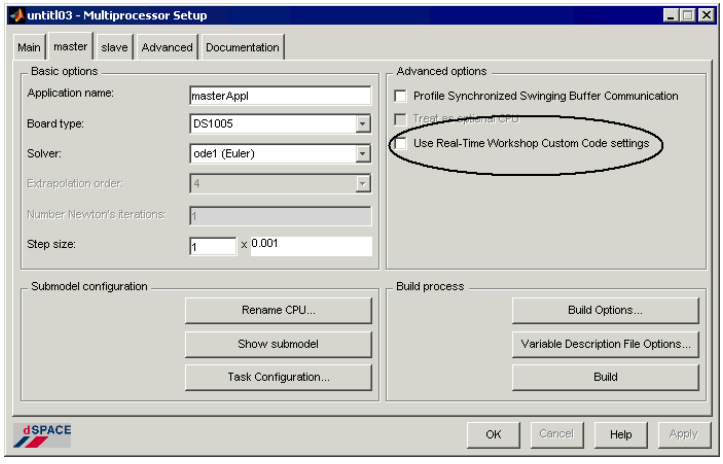
Custom Code page

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 use the RTI User-Code file (`USR.C`) to add hand-written C code to your real-time application. With the MATLAB R14SP1 support, 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 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 the MATLAB R14SP1 support, 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.

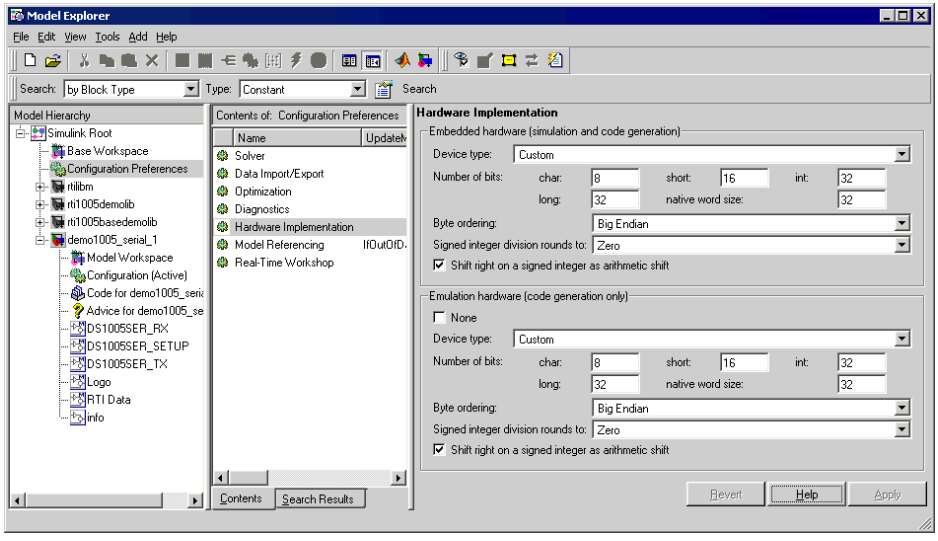


If you use Real-Time Workshop's Custom Code page, you cannot specify different custom code for each submodel of a multiprocessor system. If this is required, you must use the RTI User-Code files (`USR.C`) and RTI User Makefiles (`USR.MK`).

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 R14SP1 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.

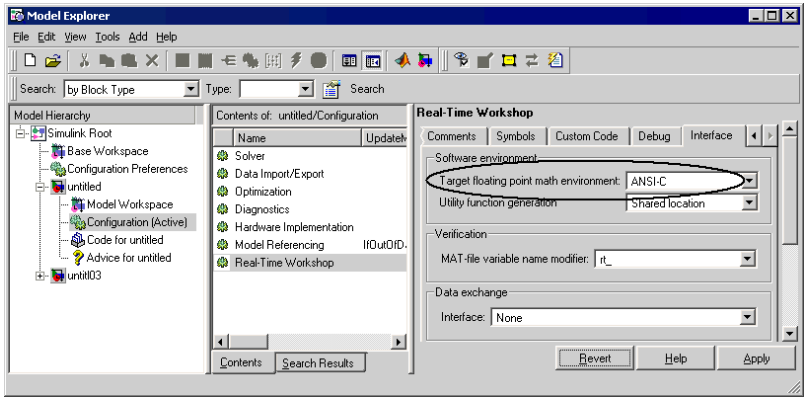
When you change the RTI system target file of your model to another RTI platform, for example, from rti1005.tlc to rti1103.tlc, all hardware characteristics are reset to the values specified in the above table.

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 the MATLAB R14SP1 support does not allow you to work with 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 support. 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?Release42_R14SP1_Support if there is a patch available for this kind of problem.

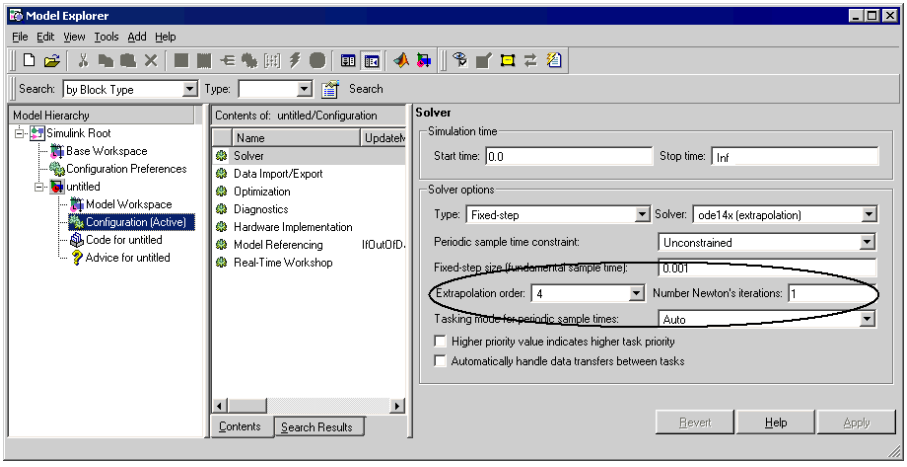
New Simulation Features

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

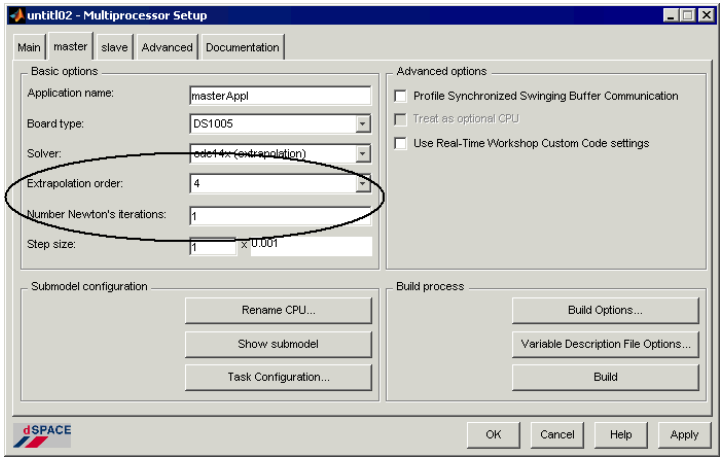
New ode14x solver

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.

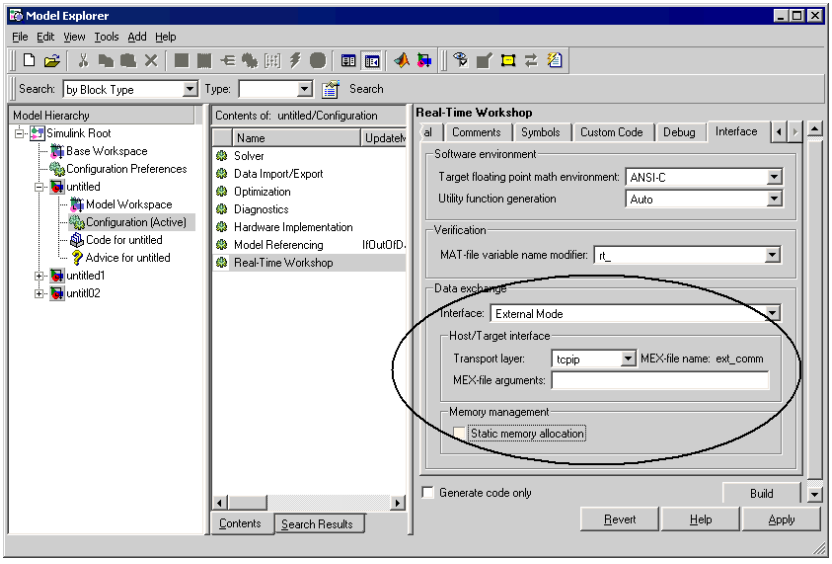


External simulation

To perform external simulation with the MATLAB R14SP1 support, 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	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 71.

Limitations of the MATLAB R14SP1 Support in dSPACE Release 4.2

You should note some limitations when you work with the MATLAB R14SP1 support in dSPACE Release 4.2.

Unsupported features of MATLAB R14SP1

There are some new features introduced with MATLAB R14 which are not supported by the MATLAB R14SP1 support. Refer to *Unsupported Features of MATLAB R14SP1* on page 59.

Software environment and compatibility

When you migrate to the MATLAB R14SP1 support, there are some limitations in the software environment and compatibility. Refer to *Limitations in Software Environment and Compatibility* on page 61.

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 62.

Limitations of the MATLAB R14SP1 Support in dSPACE Release 4.2

Handling tasks

Due to changes in MATLAB R14SP1, there are some limitations in task handling. Refer to *Limitations in Task Handling* on page 64.

Building models

There are some limitations when you build and download your model. Refer to *Limitations in Building Models* on page 68.

Simulating models

When you use the MATLAB R14SP1 support, working with external simulation is restricted. Refer to *Limitations in External Simulation* on page 71.

Miscellaneous limitations

In addition to the above limitations, there are miscellaneous limitations you have to consider when you migrate to the MATLAB R14SP1 support. Refer to *Miscellaneous Limitations* on page 73.

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 support 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 the following settings of the **Periodic sample time constraint** option in 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

If you install the MATLAB R14SP1 support in dSPACE Release 4.2, the installation does not support MATLAB R13.x (R13.0.1, R13SP1, R13SP1+, and R13SP2) and MATLAB R14. If you 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 support. It is disabled during the installation of the MATLAB R14SP1 support.

Limitations in Creating and Configuring Models

The following limitations apply to modeling and configuration.

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?Release42_R14SP1_Support if 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 4.2 (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 support. You must reconfigure the RTI options manually.

Limitations in Task Handling

The following limitations apply to task handling.

Subsystems triggered by several interrupts

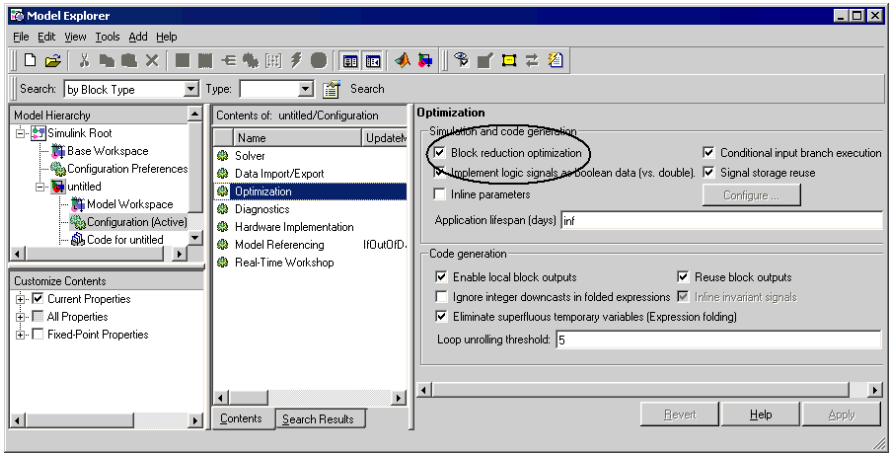
If a subsystem is triggered by several RTI interrupts (hardware, software, and/or timer interrupts), Simulink issues the following warning:

```
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 59. 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 59). 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:

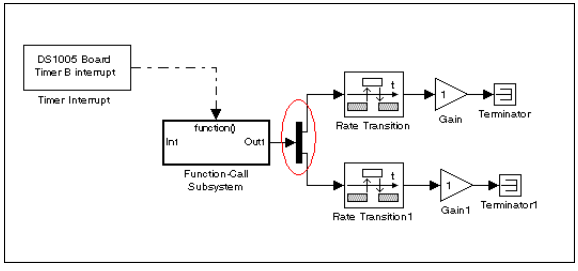
  The Simulink Rate Transition block switches to the **NoOp** mode.

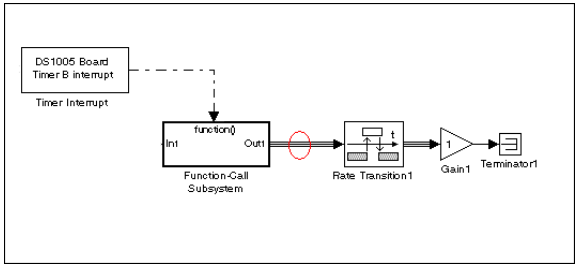
  Note that RTI does not support the Block reduction option. This mode can therefore be used only for Simulink simulations.

- 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:

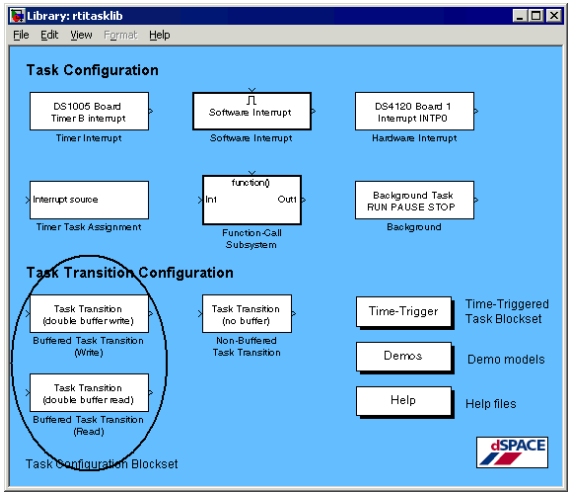
```
Warning: Inconsistent sample times. Sample time of signal
([-1, -1]) driving input port 1 of
'untitled/RateTransition1' differs from the expected sample
time at this input port ([-1, -2]).
```

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 occurring 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 might 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 and also affects MLIB and Python.

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.

Subsystem outport names containing a "/"

Existing ControlDesk experiments may access subsystem output signals at outports whose names contain a "/". When you migrate from MATLAB R13.x or earlier to MATLAB R14SP1, you might have to recreate the connections to these signals. This limitation results from a problem in Simulink and also affects MLIB and Python.

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

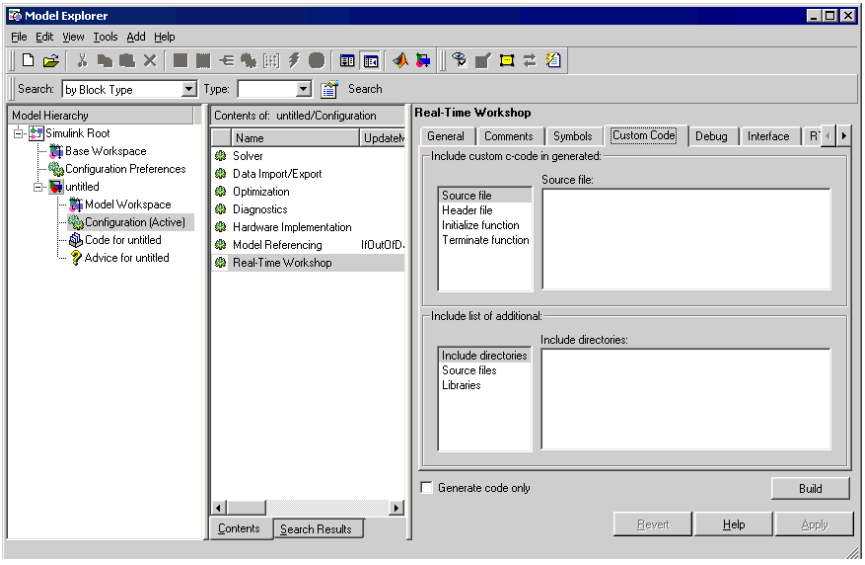
You may have to recreate all connections between vector elements to ControlDesk instruments in existing ControlDesk experiments. In a similar way, you may have to adapt MLIB and Python scripts.

This limitation applies only to parameters, not to signals.

Real-Time Workshop/ Stateflow Custom Code

Due to changes in MATLAB 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.

- 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:

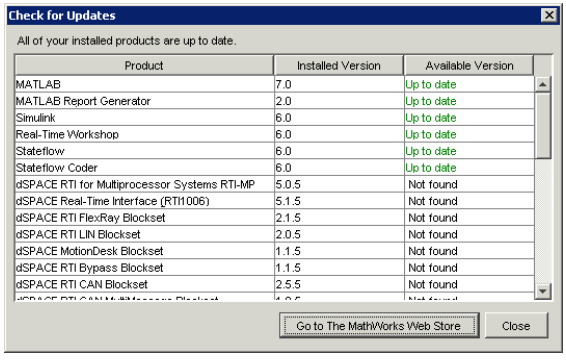
```
Checksum mismatch. Target code needs to be rebuilt.
```

Miscellaneous Limitations

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

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.

Last-Minute Information on dSPACE Release 4.2

This chapter provides information on changes and enhancements that were made after the dSPACE Release 4.2 documentation was completed.

MATLAB R14SP2 support

In addition to the support of several MATLAB R13.x versions, dSPACE Release 4.2 contains a separate set of dSPACE product versions that support MATLAB R14 with Service Pack 1 (MATLAB R14SP1) and MATLAB R14 with Service Pack 2 (MATLAB R14SP2). To install dSPACE Release for MATLAB R14SP1 or R14SP2, select the **MATLAB R14SP1/R14SP2** option during installation. The setup program then lets you specify a MATLAB path where either R14SP1 or R14SP2 is installed.

Except for some minor differences, the information provided on what is called the 'MATLAB R14SP1 support in dSPACE Release 4.2' throughout this document also applies when you use the dSPACE Release 4.2 software together with MATLAB R14SP2. Refer to the following topics:

- *Key Features of the MATLAB R14SP1 Support in dSPACE Release 4.2* on page 31
- *Limitations of the MATLAB R14SP1 Support in dSPACE Release 4.2* on page 57.

Additional features and limitations that apply if you use dSPACE Release 4.2 together with MATLAB R14SP2 are described below.

Supported features for MATLAB R14SP2

For information on the dSPACE support of MATLAB R14SP2-specific features, refer to *Supported Features of MATLAB R14SP2* on page 77.

Limitations for MATLAB R14SP2

For information on MATLAB R14SP2-specific limitations that apply when you use dSPACE Release 4.2, refer to *MATLAB R14SP2-Specific Limitations in dSPACE Release 4.2* on page 79.

Supported Features of MATLAB R14SP2

In general, the information provided in *Key Features of the MATLAB R14SP1 Support in dSPACE Release 4.2* on page 31 also applies to dSPACE Release 4.2 with MATLAB R14SP2, with the following features and differences:

Automatically handle data transfers between tasks

RTI supports the R14SP2 enhancements of the Simulink Solver option **Automatically handle data transfers between tasks**. For information on the settings for the Rate Transition blocks inserted automatically by Simulink and on the code generated for them by Real-Time Workshop, see the relevant MATLAB R14SP2 documentation from The MathWorks.

Using the Simulink Rate Transition block instead of RTI Task Transition blocks

R14SP2 includes enhancements for Simulink's Rate Transition block to now also ensure data integrity when the block is used for asynchronous task transitions and when the task priorities are not specified to Simulink (as is the case with RTI). See Real-Time Workshop 6.2 Release Notes for details. This enhancement allows you to use the Simulink Rate Transition block instead of the Task Transition blocks in the RTI TaskLib.

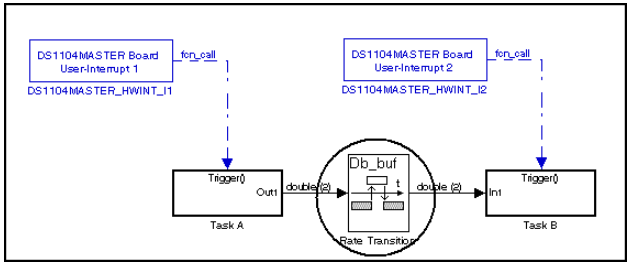
The above feature makes the following limitation obsolete when you use dSPACE Release 4.2 with MATLAB R14SP2:

- *Using the Simulink Rate Transition block* on page 65

See also *Limitations that no longer apply with MATLAB R14SP2* on page 80.

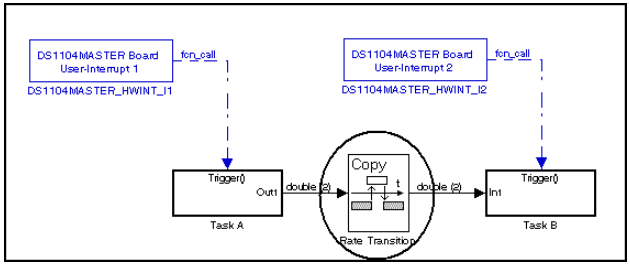
When the Rate Transition block display shows 'Db_buf' (**Ensure data integrity during data transfer** option is selected), the behavior of the Simulink Rate Transition blocks is comparable to that of a Buffered Task Transition Block pair (Write/Read) from RTI TaskLib.

The following illustration shows the Simulink Rate Transition block when protected data transfer is implemented between two interrupt-driven tasks:



When the Rate Transition block display shows 'Copy' (**Ensure data integrity during data transfer** option is not selected), the behavior of the Simulink Rate Transition block is comparable to that of a Non-Buffered Task Transition Block from RTI TaskLib.

The following illustration shows the Simulink Rate Transition block when unprotected data transfer is implemented between two interrupt-driven tasks:



MATLAB R14SP2-Specific Limitations in dSPACE Release 4.2

In general, the information provided in *Limitations of the MATLAB R14SP1 Support in dSPACE Release 4.2* on page 57 also applies to dSPACE Release 4.2 with MATLAB R14SP2, with the following limitations and differences:

MATLAB installation folders with spaces are not supported

Since MATLAB R14SP2, MATLAB Installer supports installation in folders whose paths contain spaces and, depending on your localization, may suggest the **Program Files** folder as the default. However, dSPACE software does **not** support MATLAB installation folder names with spaces in the path. The dSPACE Release Setup issues a warning if you try to install the dSPACE Release for a MATLAB installation that has spaces in its installation path.

Warning about structure assignment issued during RTI build process

In rare cases, the following warning might be issued during RTI build processes when MATLAB R14SP2 is used:

```
Because of a bug, previous releases of MATLAB have allowed
structure assignment to a nonempty nonstructure to overwrite
the previous value. MATLAB continues to allow this in this
release, but generates a warning message. In a future release,
attempting this type of assignment will generate an error.
```

This warning is harmless and can be ignored.

Real-Time Workshop C++ Language Option is not supported

In MATLAB R14SP2, Real-Time Workshop has a new option for switching the target language between C and C++. For RTI, the target language must be set to C.

Incorrect code for subsystems containing RTI Interrupt blocks

Depending on the Real-Time Workshop system code setting for atomic subsystems that include an RTI Interrupt block, incorrect code may be generated. This is a bug in Real-Time Workshop 6.3 and also may affect RTI models. For details, refer to http://www.mathworks.com/access/helpdesk/bug_reports/openbugs_rtw_6-2.html.

Limitations that no longer apply with MATLAB R14SP2

The following limitations described in *Limitations of the MATLAB R14SP1 Support in dSPACE Release 4.2* do not apply when you use dSPACE Release 4.2 with MATLAB R14SP2:

- *Edit field changes can be lost* on page 62
- *Using the Simulink Rate Transition block* on page 65 (refer also to *Using the Simulink Rate Transition block instead of RTI Task Transition blocks* on page 77)
- *Code generation problem for interrupt-driven subsystems* on page 68
- *Restricted support of external simulation* on page 71

Improvements in AutomationDesk 1.2.1

AutomationDesk 1.2.1 comes with several improvements in relationship to AutomationDesk 1.2.

Improved performance

- The time needed to save an AutomationDesk project has been reduced.
- The time needed to open a sequence in the Sequence Builder has been reduced.

Improved localization

- Unicode text is now correctly displayed in generated PDF reports.
- Problems with operating systems used with the Japanese language version have been solved.

Improved AddURL block

The AddURL automation block of the Report library provides an option to copy the specified file to the report resources. This feature now works correctly.

Improved parallel execution

The timing problems for IfThenElse, Repeat, and While automation blocks used in a Parallel automation block have been solved.

Updating projects

If you open an AutomationDesk project that you saved with an earlier AutomationDesk version, the project is first updated. For large projects, it is recommended to use the dSPACE Project Update tool. You can start the tool via **Start - dSPACE Tools - AutomationDesk - Update AutomationDesk Project**. For further information, refer to the Project Update documentation, which you can find at the same location.

New Features of MTest 1.3

MTest versions

MTest 1.3 is based on AutomationDesk 1.2.1. The following products will be installed depending on your licenses:

- MTest 1.3 and a basic version of AutomationDesk 1.2.1

  This installation contains only AutomationDesk components that are required for working with MTest. You can work with MTest projects.

- MTest 1.3 and a full version of AutomationDesk 1.2.1

  This installation contains all AutomationDesk components, and MTest as an add-on to AutomationDesk. You can work with AutomationDesk standard projects and MTest projects.

Unicode support

MTest 1.3 can work with operating system environments that use a non-Latin script, such as Japanese. You can name items such as file paths, MTest element names, and descriptions in a local language. The menus and dialogs of MTest are in English.

- The names of simulation models and their variables must be non-unicode.
- All names that result in C identifiers, like variables, types, and functions, must be specified in ASCII characters, since this is required by the C language specification.

General MTest settings

You can now specify some general settings for use by all MTest projects in MTest-specific pages of the AutomationDesk Properties dialog:

- Regenerate TargetLink code
  If you start a test execution in software-in-the-loop (SIL) or processor-in-the-loop (PIL) mode, the code to be tested should be updated with current modifications. This setting specifies whether to regenerate the code on every test execution, or only if there are modifications that require a code update, or never.
- Init file execution
  Test execution uses initialization files that you specify. This setting specifies whether the initialization files are executed on every test execution, or only if a different initialization file was executed previously.
- CTE step transition
  If you specify the test data using Razorcat's Classification Tree Editor (CTE), the steps are generated at either the end or the start of a step transition. In earlier MTest versions, the steps were always generated at the end of a step transition.
- Preferences
  The MTest Preferences dialog for specifying various user-specific paths was formerly located in **Start - dSPACE Tools - AutomationDesk**. It is now integrated in the AutomationDesk Properties dialog.
- Report settings
  MTest 1.3 uses the new report concept introduced with AutomationDesk 1.2. This requires some settings, for example, the output format must be specified. The report settings made for AutomationDesk projects are ignored in MTest.

New report concept

- MTest's new report concept includes additional information on project elements, like the description that you entered in an element's Properties dialog.
- The report element is now included in the result element and is accessible only via the Result Browser.
- Report generation is more flexible than before. You can activate report generation in the Execution Configuration dialog, when you start a test, or later on via the result element's context menu. In the Result Browser, you can start report generation on different levels of the result. You can generate a report for a result subset, for example, for the results of a test in SIL mode.
- The display of signal plots has been improved.
- Image Information included in the report, for example, the test frame, is now stored in PNG format.

New MTest element for additional documents

For large descriptions, additional data information or other relevant sources for a test, you can now add a **Document** element to your MTest project and manage it via MTest.

Faster access to initialization files

All Properties dialogs in which you can specify an initialization file now have a button to open the initialization file in an editor.

Updating a test frame

If you have customized the generated test frame, for example, by adding blocks or subsystems to the test frame, or modifying the effective test interface, you can now update the containing test object - with the same interface - without creating a new test frame and recustomizing the interface.

Fixed-point support

MTest now supports fixed-point data types. MTest recognizes fixed-point data types used in the interface of the test object. You can also specify additional signals in fixed-point data type notation to be used for the test interface.

Configuring evaluation methods for MTest test elements

Formerly, you specified an evaluation configuration for each test execution environment type (Simulink, MIL, SIL, PIL). You can now specify an evaluation configuration for the MTest test element. Each child element then uses this evaluation method. You can still specify an individual evaluation method for each test execution environment type.

Testing of TargetLink modules

TargetLink modules are now selectable as test objects in MTest. If a TargetLink module contains a TargetLink Function block and its interface has TargetLink inports and outports, the test results for the model and the generated code will be comparable. If the interface of the TargetLink module does not have TargetLink inports and outports, MTest automatically inserts the required TargetLink ports in the test frame. To achieve comparable results in SIL and PIL mode as well, you must configure the ports before starting a test.

The dSPACE tool for updating AutomationDesk projects (**Start - dSPACE Tools - AutomationDesk - Update AutomationDesk Project)** cannot be used for MTest projects.