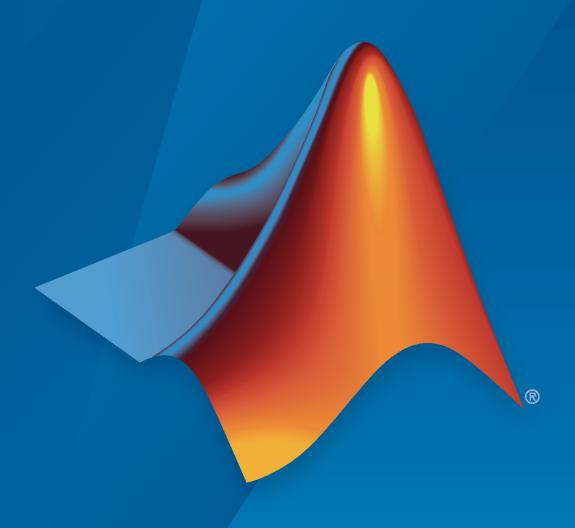Vehicle Network Toolbox™ Release Notes

MATLAB®&SIMULINK®

MathWorks®

How to Contact MathWorks

Latest news: www.mathworks.com

Sales and services: www.mathworks.com/sales_and_services

User community: www.mathworks.com/matlabcentral

Technical support: www.mathworks.com/support/contact_us

Phone: 508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

Contents

R2020b

R2020a

R2019b

R2019a

R2018b

R2018a

R2022a

Version: 5.2

New Features

Bug Fixes

Write to MDF-Files: Create and modify MDF-files

Vehicle Network Toolbox now allows you to create or write to MDF-files. The following new functions support this capability:

- `mdfCreate` — Create MDF-file with metadata
- `mdfWrite` — Write timetable data to MDF-file
- `mdfAddChannelGroupMetadata` — Add channel group and channel metadata to timetable
- `mdfAddAttachment` — Attach file to MDF-file
- `mdfRemoveAttachment` — Remove attachment from MDF-file

In addition:

- The `mdfInfo` function has been updated to optionally run without input arguments, to return a default file metadata structure that you can customize and use with `mdfCreate`.
- The `read` function for MDF-files now accepts an `IncludeMetadata` name-value pair input argument. The default value is `false`; when specified as `true`, channel group metadata and channel metadata are included as custom properties of the output timetable. Using the `IncludeMetadata` option might impact function performance when reading data from a channel group with many channels.

The following examples show you how to write to MDF-files:

- "Get Started with Writing Timetable Data to MDF-Files"
- "Custom Set Metadata When Writing Timetable Data to MDF-Files"
- "Overwrite Channel Group Data in an MDF-File"
- "Write Channel Group Data from an Existing MDF-File to a New MDF-File"
- "Work with MDF Attachment Files"
- "Decode CAN Data from and Write Results to MDF-Files"

Enhanced MDF read Function: Read all channels in a channel group

You can now read all channels in an MDF-file channel group, without having to specify individual channel names. The function syntax to perform this is:

```
data = read(mdfObj,chanGroupIndex)
```

For more information, see `read`.

Enhanced channelList Output: Added table variables

The output table from the `channelList` function for MDF-files is enhanced with the addition of the following variables:

- `ChannelGroupSourceInfo`
- `ChannelExtendedNamePrefix`
- `ChannelType`
- `ChannelDataType`

- ChannelNumBits
- ChannelComponentType
- ChannelCompositionType
- ChannelConversionType
- ChannelSourceInfo

Enhanced mdf Object: Added SourceInfo field for ChannelGroup and Channel properties

The mdf object now includes SourceInfo fields in the struct values of the ChannelGroup and ChannelGroup.Channel properties.

R2021b

Version: 5.1

New Features

Bug Fixes

Version History

A2L Version 1.7.1: xcpA2L function creates A2L version 1.7.1 object

In R2021b, the `xcpA2L` function executes faster and produces an `xcp.A2L` object that complies with version 1.7.1 A2L. For more information, see the ASAM MCD-2 MC v1.7.1 specification.

The changes to the `xcp.A2L` object include changes to the xcp.A2L Properties. These properties are no longer read-only.

Version History

The change to the generated `xcp.A2L` object that affects compatibility is:

• If a property does not exist in the A2L file, it results in a 0-by-0 empty class or enums.

MATLAB XCP: Support over CAN FD

Vehicle Network Toolbox now supports XCP functionality for the CAN FD protocol in MATLAB® on Windows® systems. When creating an XCP channel with the `xcpChannel` function, you can specify `"CAN FD"` as the CAN protocol mode. For example:

```
xcpch = xcpChannel("XCPSIM.a2l","CAN FD","Vector","Virtual 1",1)
```

Note XCP over CAN FD is not supported for PEAK-System devices.

MDF-File Enhancements: Finalizing and sorting

A new function `mdfFinalize` allows you to sort and finalize an MDF-file in place or into a new copy of the file.

In addition to previous functionality, the `mdfSort` function now allows you to sort an MDF-file in place without creating a copy of the file.

For an example, see Work with Unfinalized and Unsorted MDF-Files.

Linux Platform Support: BLF-file access

You can now access BLF-files in MATLAB on Linux® systems. See Standard File Formats.

blfread Function: Enhanced performance reading BLF-files

The `blfread` function now provides improved performance when reading a large number of messages from a BLF-file.

The test code to measure the performance with database decoding is:

```
profile on
candb = canDatabase("DB-file.dbc");
data = blfread("BLF-file.blf","Database",candb);
profile viewer
```

The test code to measure the performance without database decoding is:

```
profile on
data = blfread("BLF-file.blf");
profile viewer
```

When the test script was timed on a Windows 10, Intel® Xeon® W-2133 CPU @ 3.60-GHz system, the approximate execution times and memory usage for a few example files were:

Data Size	R2021a Execution time (sec)	R2021b Execution time (sec)	R2021a Memory Usage (GB)	R2021b Memory Usage (GB)
5.4 million messages with database decoding	294.5	93.8	19.8	18.0
5.4 million messages without database decoding	165.5	59.9	10.5	4.8

Pack and Unpack J1939 parameter group signals

The CAN Pack and CAN Unpack blocks can be used to encode and decode signals of J1939 parameter groups that are up to 8 bytes.

Vehicle CAN Bus Monitor removed

The Vehicle CAN Bus Monitor (CAN Tool) and its start function `canTool` have been removed.

Version History

The Vehicle CAN Bus Monitor is no longer available. Use the **CAN Explorer** app or **CAN FD Explorer** app instead.

New Examples

This release includes the following new examples using Vehicle Network Toolbox:

- CAN, CAN FD, and J1939 Communication in Simulink

  - Get Started with CAN Communication in Simulink
  - CAN Message Reception Behavior in Simulink
  - Replay J1939 Logged Field Data to a Simulation
- XCP Communication in MATLAB

  - Get Started with A2L-Files
  - Calibrate XCP Characteristics
  - Read XCP Measurements with Dynamic DAQ Lists
  - Read XCP Measurements with Direct Acquisition
- Standard File Formats in MATLAB

- Work with Unfinalized and Unsorted MDF-Files
- Analyze Data Using MDF Datastore and Tall Arrays
- Decode J1939 Data from MDF-Files
- Decode J1939 Data from BLF-Files

Functionality being removed or changed

Functionality	Result	Use Instead	Compatibility Considerations
Vehicle CAN Bus Monitor (`canTool`)	Error	**CAN Explorer** or **CAN FD Explorer**	Replace usage of `canTool` with `canExplorer` or `canFDExplorer`.

R2021a

Version: 5.0

New Features

Bug Fixes

Version History

New Apps for CAN Protocol Acquisition and Visualization: CAN Explorer and CAN FD Explorer

Two new apps provide graphical workflows for acquiring and visualizing CAN and CAN FD data. With the **CAN Explorer** and **CAN FD Explorer** you can apply CAN or CAN FD databases, configure a device and channel, select specific messages and signals, acquire and visualize data, and export data to the MATLAB workspace.

Use the **CAN Explorer** for networks and data configured for the CAN protocol. Use the **CAN FD Explorer** for networks and data configured for the CAN FD protocol.

For more information, see the following examples:

- Using CAN Explorer for Data Reception and Visualization
- Using CAN FD Explorer for Data Reception and Visualization

Version History

The Vehicle CAN Bus Monitor app will be removed in a future release. Use the **CAN Explorer** or **CAN FD Explorer** instead.

MDF-File Data Types: Choose applied conversion rules when reading MDF data into MATLAB

You can now choose the conversion options to apply when reading data from an MDF-file. You specify the option in the following ways:

- For `mdf` objects, set the `Conversion` property of the object, or specify a `"Conversion"` argument pair when calling the `read` function.
- For `mdfDatastore` objects, set the `Conversion` property of the object.

Allowed property values and argument options are:

- `'Numeric'` (default) — Apply only numeric conversion rules (CC_Type 1-6). Data with non-numeric conversion rules is imported as raw, unconverted values.
- `'None'` — Do not apply any conversion rules. All data is imported as raw data.
- `'All'` — Apply all numeric and text conversion rules (CC_Type 1-10).

For an example of reading MDF data with applied conversion rules, see Reading Data from an MDF-File with Applied Conversion Rules.

Version History

The default `Conversion` value is `'Numeric'`, which is the closest to the behavior of past releases. Certain numeric conversion rules might result in integer or single data types when the data is read into MATLAB by code that in past releases would return double.

MDF-File Text: Improved support for international characters

The MDF functions are enhanced to provide better support for international characters. Note that `mdfVisualize` and `mdfSort` do not offer support for the full range of characters supported in the other MDF functions.

J1939 Timetable Support

The J1939 `receive` and `j1939ParameterGroupImport` functions now return timetables of parameter groups.

The new function `j1939SignalTimetable` creates a signal timetable extracted from a J1939 parameter group timetable.

The new function `j1939ParameterGroupTimetable` allows you to:

- Convert an array of J1939 parameter group objects into a J1939 parameter group timetable.
- Convert raw CAN frames from a J1939 system stored in BLF or MDF files, into a J1939 parameter group timetable.

Version History

In previous releases, the `receive` and `j1939ParameterGroupImport` functions returned arrays of parameter group objects. Now they return timetables.

You can upgrade existing code by using `j1939ParameterGroupTimetable` to convert arrays of J1939 parameter group objects into J1939 parameter group timetables.

Linux Platform Support: J1939 in MATLAB

You can now perform J1939 communications in MATLAB on Linux systems.

Linux Platform Support: mdfVisualize function

The `mdfVisualize` function is now supported on Linux systems.

Parallel reading of MDF datastore files

You can now read multiple files of an MDF datastore in parallel for improved performance. The expanded form of the `readall` function syntax is:

```
data = readall(mdfds,"UseParallel",true)
```

For more information, see `readall`.

CDFX-file matrix orientation updated

To effect compliance with ASAM AE Calibration Data Format V2.1.0, the matrix orientation of CDFX-file reads are updated in this release:

- Values in an `SW-ARRAYSIZE` field of a CDFX-file are now read as columns-by-rows-by...
- Values in a `V` or a `VG` field are now read as row-major.

Version History

In past releases, `SW-ARRAYSIZE` values were read as rows-by-columns-by...; and `V` and `VG` values were read as column-major.

Any code that depended on this orientation must be updated for this release.

New Examples

This release includes the following new examples using Vehicle Network Toolbox:

- Reading Data from an MDF-File with Applied Conversion Rules
- Using CAN Explorer for Data Reception and Visualization
- Using CAN FD Explorer for Data Reception and Visualization

R2020b

Version: 4.5

New Features

Bug Fixes

Version History

Enhanced Simulink Support for XCP: New XCP UDP Bypass block

The Vehicle Network Toolbox XCP UDP Communications block library now includes the XCP UDP Bypass block, which connects the function-call outport to a function-call subsystem. For more information, see XCP UDP Bypass.

MDF read Function: Enhanced performance reading MDF-files

The read function for MDF-files now provides improved performance when reading numeric data and byte array data from channel groups that contain a large number of channels.

The test code to measure the performance is:

```
m = mdf(mdfFile);
tic
    d = read(m,1,m.ChannelNames{1});
toc
```

When the test script was timed on a Windows 10, Intel Xeon CPU E5-2650 v2 @ 2.60-GHz system, the approximate execution times for a few example files were:

Data Type	Number of Channels	Samples per Channel	R2020a Execution Time (sec)	R2020b Execution Time (sec)	Speedup Factor
Real Little-Endian	3000	10000	2425	14.3	170x
Real Little-Endian	10000	3000	8134	48.6	167x
Byte Array	1000	5000	1333	33.9	39x
Byte Array	5000	1000	6181	56.4	109x

When the test script was timed on a Debian® 10, Intel Xeon CPU E5-1650 v3 @ 3.50-GHz system, the approximate execution times for a few example files were:

Data Type	Number of Channels	Samples per Channel	R2020a Execution Time (sec)	R2020b Execution Time (sec)	Speedup Factor
Real Little-Endian	3000	10000	1789	14.7	121x
Real Little-Endian	10000	3000	6040	70.8	85x
Byte Array	1000	5000	859	23.6	37x
Byte Array	5000	1000	4138	57.5	72x

Enhanced MDF Reading: Return physical data when possible

Reading data from MDF-files now returns physical values for any data samples that can be converted, and raw values only for those samples that cannot be converted.

Version History

In previous releases, if any data samples could not be converted from raw to physical values, all data samples requested from that channel were returned as raw.

New Examples

This release includes several new examples using Vehicle Network Toolbox.

- Event-Based CAN Transmission in Simulink shows how to use events to transmit CAN messages in Simulink® only when a change in data from one time step to the next is detected.

The following examples show how to import and decode CAN data from standard format files.

- Decoding CAN Data from MDF-Files
- Decoding CAN Data from BLF-Files

R2020a

Version: 4.4

New Features

Bug Fixes

Linux Platform Support: Access MDF-Files

You can now access MDF-files on Linux systems. For more information on platform support for standard file types, see Standard File Formats.

Simulink Blocks: CAN transmission on data change

The CAN Transmit and CAN FD Transmit blocks now support event-based transmission.

You can enable event-based transmission in the block parameters dialog by selecting **On data change** in the Transmit Options section.

When event-based transmission is enabled, messages are transmitted only at those time steps when a change in message data is detected. When the input data matches the most recent transmission for a given message ID, the message is not re-transmitted.

Event and periodic transmission can both be enabled to work together simultaneously. If neither is selected, the default behavior is to transmit the current input at each time step.

Programmatically Access CAN FD Block Parameters

The following Simulink blocks now support programmatic access to their parameters with the `set_param` and `get_param` functions:

```
CAN FD Configuration
CAN FD Transmit
CAN FD Receive
```

New Examples

This release includes the following new example showing how to validate the output of a cruise control algorithm using Vehicle Network Toolbox and MATLAB Class-Based Unit Testing Framework.

• Class-based Unit Testing Of Automotive Algorithms Via CAN

For more information, see also Class-Based Unit Tests (MATLAB).

R2019b

Version: 4.3

New Features

Bug Fixes

Linux Platform Support: Connect to Kvaser devices in MATLAB and Simulink

This release supports connecting to Kvaser devices and virtual channels on Linux platform systems in both MATLAB and Simulink. This support includes code generation in both environments, and rapid accelerator mode in Simulink.

Programmatically Access CAN/CAN FD Block Parameters

The following Simulink blocks now support programmatic access to their parameters with the `set_param` and `get_param` functions:

```
CAN FD Pack
CAN FD Unpack
CAN Pack
CAN Unpack
CAN Configuration
CAN Transmit
CAN Receive
```

For an example, see Programmatically Building Models for CAN Communication.

MDF-File Information and Content Sorting

Two new functions allow you to access MDF-file metadata and to sort the data contents of an MDF-file.

File Information — The `mdfInfo` function returns a struct that contains information from the file header metadata within an MDF-file. For example:

```
fileInfo = mdfInfo('MDFFile.mf4');
```

Data Sorting — The `mdfSort` function allows you to create a copy of an MDF-file with its data sorted for reading. For example:

```
sortedPath = mdfSort('UnsortedMDFFile.mf4','SortedMDFFile.mf4');
```

If you get an error when trying to read from an unsorted file, create a sorted file and read from that instead. The optional output argument, `sortedPath`, indicates the full path name to the sorted file.

Improved MDF-File Import Performance

This release provides significantly improved performance when opening and reading MDF-files. On a test machine with Windows 10, Intel Xeon W-2133 CPU @ 3.60 GHz:

- The `mdf` function shows a 28% speedup in R2019b when opening files containing a large number of channel groups and channels.
- The `read` function shows a 43% speedup in R2019b when reading data from files containing channels with a large amount of data.

Simulink Code Generation with Vector Devices: Support execution on different devices of the same hardware model

Code generated from Simulink models that use CAN blocks with Vector devices now supports execution with any device of the same Vector hardware model. Code generated in previous releases executed only on the Vector device with the same serial number.

New Examples

This release includes the following new example showing a Vehicle Network Toolbox application in Simulink.

- Programmatically Building Models for CAN Communication

R2019a

Version: 4.2

New Features

Bug Fixes

Version History

ASAM CDFX-File Support

This release supports ASAM calibration data format CDFX-files. You can import, modify, and export data to existing instances in CDFX-files. Adding or removing instances is not supported.

The following new functions provide support for this feature.

```
cdfx
instanceList
systemList
getValue
setValue
write
```

For more information, see the following examples:

*   Working With ASAM CDFX Files in MATLAB
*   Using ASAM CDFX Data With Simulink
*   Using ASAM CDFX Data With Simulink Data Dictionary

See also File Format Limitations.

Vector BLF-File Support

Vehicle Network Toolbox now supports reading and writing Vector binary logging format BLF-files for CAN and CAN FD. For more information, see the reference pages for the following new functions.

*   `blfinfo`
*   `blfread`
*   `blfwrite`

Convert ASAM MDF Format Timetables to Vehicle Network Toolbox Timetables

The `canMessageTimetable` and `canFDMessageTimetable` functions are enhanced to accept a timetable of ASAM MDF standard logging format data from CAN or CAN FD messages, and return Vehicle Network Toolbox timetables of CAN or CAN FD messages. For example, to convert messages from ASAM MDF CAN format to a timetable:

```
m = mdf('myfile.mf4');
mdfData = read(m); % Creates ASAM format timetables
db = canDatabase('myDatabase.dbc')
msgTimetable = canMessageTimetable(mdfData{1},db);
```

Code Generation Support for CAN FD Blocks: Deploy models using CAN FD Replay and CAN FD Log blocks as stand-alone executables

The following blocks now support code generation for CAN FD:

*   CAN FD Replay
*   CAN FD Log

XCP Protocol: MAX_DLC_REQUIRED recognized for XCP over CAN in MATLAB

An XCP master now uses the `MAX_DLC_REQUIRED` setting from the A2L file to send fully padded, fixed-length XCP messages from MATLAB, if required.

Open MDF-File in Simulation Data Inspector

This release provides a new function to open MDF-files for viewing and analyzing in the Simulation Data Inspector. For more information, see the `mdfVisualize` reference page.

Enhanced Simulink Support for XCP

The Vehicle Network Toolbox XCP Communication block library includes several new blocks which now distinguish XCP communication over CAN or over UDP.

With these enhancements, you can now have XCP communication over CAN and over UDP simultaneously in the same model. A model can also support multiple XCP slaves in UDP. The following table maps the old blocks to the new.

Old Block	New Replacement
XCP Configuration	XCP CAN Configuration or XCP UDP Configuration
XCP Data Acquisition	XCP CAN Data Acquisition or XCP UDP Data Acquisition
XCP Data Stimulation	XCP CAN Data Stimulation or XCP UDP Data Stimulation
XCP UDP Transport Layer	Functionality incorporated into XCP UDP Configuration

Version History

With this enhancement, some blocks in past releases that supported XCP over CAN or UDP are replaced with new blocks to distinguish the protocol. The XCP UDP Transport Layer block is no longer necessary. When you open existing models, the old blocks are automatically replaced by the new blocks.

Timestamp Support in XCP Blocks

The following blocks now support accompanying data with timestamps. When the Timestamp is enabled, the block reads or sends the timestamps between Simulink and the XCP slave. The **Enable Timestamp** check box appears in the block parameters dialog box when the parameter is supported in the specified A2L file.

- XCP CAN Data Acquisition
- XCP CAN Data Stimulation
- XCP UDP Data Acquisition

- XCP UDP Data Stimulation

Improved Consistency for Signal Value Packing

When encoding or packing signal values, the following Simulink blocks now round rather than truncate the least significant bits: CAN Pack, CAN FD Pack, and J1939 Transmit

As a result, these blocks now generate values consistent with the MATLAB operations that encode and decode signal values.

Version History

Because of this change, the signal values returned by these blocks might be slightly different from past releases.

Improved DBC Attribute Presentation

DBC attribute values are now handled in a manner more consistent with their data types defined in the DBC file.

Version History

Some DBC attribute value data types are changed in this release. For example, DBC values of native hexadecimal type are now displayed in hex, rather than decimal as in the past. Enumeration attribute values display as strings rather than numeric values.

New Examples

This release includes the following new example showing Vehicle Network Toolbox applications in MATLAB and Simulink.

- Mapping MDF Channels to Simulink Model Input Ports
- Working With ASAM CDFX Files in MATLAB
- Using ASAM CDFX Data With Simulink
- Using ASAM CDFX Data With Simulink Data Dictionary
- Developing an App Designer App for a Simulink Model using CAN

R2018b

Version: 4.1

New Features

Bug Fixes

CAN FD Message Logging and Replay Blocks: Log CAN FD messages from vehicle test runs to a file, and replay logged data as input to a Simulink model

The following new blocks support CAN FD messages. For more information, see the reference page for each:

- CAN FD Log
- CAN FD Replay

For more information on an application using these blocks, see the example Log and Replay CAN FD Messages.

Code Generation Support for CAN FD Blocks: Deploy models using CAN FD blocks as stand-alone executables, and use CAN FD Pack and CAN FD Unpack blocks with embedded targets

The following blocks now support code generation for CAN FD:

- CAN FD Configuration
- CAN FD Pack
- CAN FD Receive
- CAN FD Transmit
- CAN FD Unpack

Note The new CAN FD Log and CAN FD Replay blocks do not support code generation.

Code Generation: Improved code quality

Improved code generation from the following blocks results in code of higher quality and efficiency:

- CAN Pack
- CAN Unpack
- J1939 Transmit
- J1939 Receive

New CAN FD Functions

The following new functions allow you to directly use CAN FD without having to specify the protocol mode. For more information, see the reference page for each function.

- canFDChannelList
- canFDChannel
- canFDMessage
- canFDMessageTimetable

- `canFDMessageReplayBlockStruct`

MDF channelList Function: Search an MDF-file to find specific channels

The `channelList` function returns a table of information about channels and groups in an MDF-file. The output allows you to identify and quickly access specific channels. For more information, see the `channelList` reference page.

MDF-File Import Performance: Open MDF-files significantly faster than in prior releases

This release offers significantly better performance when accessing MDF-files.

XCP Protocol: DAQ/STIM computation method scaling and vector measurements

New functions allow you to access scaled DAQ/STIM data and support vector-based measurements. For more information, see the reference pages for the following new functions:

- `readDAQ`
- `writeSTIM`

XCP Protocol: Support DAQ list priority in Simulink DAQ block

You can now specify the numeric priority of a DAQ list to control DAQ list output from an XCP slave.

XCP Protocol: MAX_DLC_REQUIRED recognized for XCP over CAN in Simulink

An XCP master now uses the `MAX_DLC_REQUIRED` setting from the A2L file to send fully padded, fixed-length XCP messages, if required.

Hardware Support: Connect to PEAK-System hardware on Linux platform and to Kvaser Hybrid hardware on Windows platform

This release of the toolbox supports PEAK-System devices on Linux.

This release of the toolbox supports Kvaser Hybrid devices on Windows.

For a complete list of supported CAN hardware, see https://www.mathworks.com/hardware-support/can-bus-software.html.

R2018a

Version: 4.0

New Features

Bug Fixes

Version History

CAN FD Protocol: Send and receive messages with Simulink using the CAN FD protocol

This release provides a new library of Simulink blocks to support CAN FD communication. The CAN FD Communication library includes the following blocks. For more information, see the reference page for each block.

- CAN FD Configuration
- CAN FD Transmit
- CAN FD Receive
- CAN FD Pack
- CAN FD Unpack

These blocks offer full support for the CAN FD protocol, while the workflow for using them is similar to that for the existing CAN blocks.

XCP Protocol: Communicate with ECUs from Simulink using XCP over UDP, or from MATLAB using XCP over UDP and TCP

Vehicle Network Toolbox functionality in MATLAB now supports XCP protocol communication over UDP or TCP. The `xcpChannel` function is enhanced to accept `'UDP'` or `'TCP'` as the transport layer argument, followed by arguments specifying the IP address and port values. For example:

```
xcpChan = xcpChannel(A2LfileObj,'TCP',IPaddr,portNmbr)
```

A new XCP UDP Transport Layer block accommodates Simulink XCP communication over UDP. This new block is in the XCP Communications library. For more information, see the XCP UDP Transport Layer block reference page.

XCP Protocol: Tune characteristics and read/write values with computation method scaling in MATLAB

This release supports scaling and direct memory reading and writing for XCP channel axes, characteristics, and measurements. For more information, see the reference pages for the following new functions:

```
readCharacteristic
writeCharacteristic
readMeasurement
writeMeasurement
readAxis
writeAxis
```

Version History

As part of this enhancement, the A2L parser returns an object from `xcpA2L` with more properties and of different types. In previous releases, many of the xcp.A2L object properties were structures; but now there are more properties, and those that used to be structures are now objects. Accessing the content of these properties using dot notation should work the same way; but other means of access might require modification.

MDF: Read entire file content in a single function call

The MDF `read` function has been enhanced to allow you to read the entire MDF-file contents with a single call. To read the whole file, call the `read` function with only the MDF-file object as an argument; for example:

```
data = read(mdfObj)
```

Linux Platform Support: Encode and decode CAN or CAN FD messages and communicate on virtual channels on Linux

This release supports encoding and decoding CAN and CAN FD messages on MathWorks virtual channels on Linux. This support does not include other vendors, hardware, or protocols.

Note You cannot run on Linux any Simulink models that contain blocks from the Vehicle Network Toolbox XCP Communication or J1939 Communication block libraries.

Hardware Support: Connect to additional Vector CAN interface devices

Vehicle Network Toolbox now dynamically supports additional Vector devices recognized by the driver. For more information on support for Vector devices, see https://www.mathworks.com/hardware-support/vector.html.

New Examples

This release includes new examples showing Vehicle Network Toolbox applications in MATLAB and Simulink.

MATLAB

- Executing a Forward Collision Warning Application with Live CAN FD & TCP/IP Data
- Data Analytics Application With Many MDF Files

Simulink

- CAN Connectivity in a Robotics Application
- CAN Connectivity in an Automotive Application

MDF InitialTimestamp: Improved resolution and more time zones

The MDF object returned from the `mdf` function now has an `InitialTimestamp` property value of a datetime type with nanosecond resolution.

The function now returns a UTC value for this property, or if that's not available, a local time. If the value is local time, you must account for its offset from other time zones. If no start time can be determined, it returns an empty datetime.

Version History

Previous versions of the toolbox returned an `InitialTimestamp` only for UTC, with a resolution of 1 second.

R2017b

Version: 3.4

New Features

Bug Fixes

Version History

CAN FD Protocol: Send and receive messages with MATLAB using the CAN FD protocol

Vehicle Network Toolbox now supports the ISO CAN FD protocol in MATLAB for all vendors, including MathWorks virtual channels, for raw messages and message definitions from DBC files.

The following functions are updated to accommodate the new `ProtocolMode` property with its `'CAN FD'` value, and the bus timing configuration necessary for CAN FD. See each function reference page for more information and examples.

```
canfdch = canChannel('Vector','Virtual 1',1,'ProtocolMode','CAN FD');
canfdmsg = canMessage(1000,false,64,'ProtocolMode','CAN FD')
configBusSpeed(canfdch,arbbusspeed,databusspeed)
```

Note The Vehicle CAN Bus Monitor does not support the CAN FD protocol.

MDF Datastore: Work with large MDF-file collections that might not fit in memory

Vehicle Network Toolbox allows you to create a datastore from multiple MDF-files. For more information, see Measurement Data Format (MDF).

Simulink Bus Objects: Simplify Simulink block diagrams by choosing Simulink Bus Signals when configuring CAN blocks

The blocks provided in the CAN Communication block library of Vehicle Network Toolbox now support the use of Simulink bus objects.

Among these blocks, the CAN Pack, CAN Receive, and CAN Replay block parameters have a new **Output as bus** option, to output a native Simulink bus signal.

For more information on Simulink bus objects, see Composite Signals.

Hardware Support: Connect to additional National Instruments USB CAN interface devices

This release offers support for the USB-8502 National Instruments™ USB CAN interface device.

For a list of other supported devices, see Supported Hardware.

canChannelList: List CAN devices and channels in MATLAB table

The new function `canChannelList` generates a MATLAB table that contains information about the available vendors, devices, and channels. For more information and examples, see the `canChannelList` reference page.

Version History

Because the new function provides similar information, the following functions will be removed in a future release.

```
canHWInfo
CAN.VendorInfo
CAN.ChannelInfo
```

MDF-File Access: Improved performance

This release offers improved speeds for MDF-file access with the MDF `read` function.

A2L File Access: Improved performance

This release offers improved speeds for A2L file access. For information on XCP communication with A2L files, see A2L File Management.

Hardware Support: Kvaser Smartcraft Diagnostic Interface

This release offers support for the Kvaser Smartcraft Diagnostic Interface.

Functionality being removed or changed

Functionality	Result	Use Instead	Compatibility Considerations
`canHWInfo`	Still works	`canChannelList`	Replace instances of `canHWInfo` with `canChannelList`.
`CAN.VendorInfo`	Still works	`canChannelList`	Replace instances of `CAN.VendorInfo` with `canChannelList`, and index into the resulting table.
`CAN.ChannelInfo`	Still works	`canChannelList`	Replace instances of `CAN.ChannelInfo` with `canChannelList`, and index into the resulting table.

R2017a

Version: 3.3

New Features

Bug Fixes

Version History

Virtual Channels: Emulate CAN bus communication without hardware

This release of Vehicle Network Toolbox includes MATLAB and Simulink support for a MathWorks® virtual CAN device with two channels. You can view the device and channels in MATLAB with the `canHWInfo` function.

```
info = canHWInfo

info =

CAN Devices Detected

    Vendor    |   Device   | Channel | Serial Number |             Constructor
  ----------- | ---------- | ------- | ------------- | -------------------------------------
  MathWorks   | Virtual 1  | 1       | 0             | canChannel('MathWorks','Virtual 1',1)
  MathWorks   | Virtual 1  | 2       | 0             | canChannel('MathWorks','Virtual 1',2)
```

Many examples throughout the documentation show how to use these virtual channels, so that you can run them on your own system. For example, see Transmit and Receive CAN Messages.

Timetable Formatted CAN Messages and Signals: Receive and import data in timetables, and convert existing message objects to timetables

When dealing with CAN messages and signals, the use of timetables instead of arrays of objects can offer significant performance and display improvements. For general information on timetables, see Timetables.

The following functions are enhanced with the option to return timetables of CAN messages. See their reference pages for more information and examples.

Function	Purpose
`receive`	Receive messages from CAN bus
`canMessageImport`	Import CAN message log file from third-party tool

The following new functions create CAN message and signal timetables from existing messages.

Function	Purpose
`canMessageTimetable`	Convert CAN messages into timetable
`canSignalTimetable`	Create CAN signal timetables from CAN message timetable

Other functions support CAN messages as timetables without any modification to their syntax. These include `transmit`, `replay`, `save`, and `load`.

J1939 Parameter Group and CAN Signal Import Functions: Import third-party log files directly into J1939 parameter groups and CAN signal timetables

The following new functions support the ability to import log files into J1939 parameter groups and CAN signal timetables. See their reference pages for more information and examples.

Function	Purpose
canSignalImport	Import CAN log file into decoded signal timetables
j1939ParameterGroupImport	Import J1939 log file

canMessageReplayBlockStruct Function: Convert CAN messages for CAN Replay block

The new function canMessageReplayBlockStruct formats CAN messages for use with the CAN Replay block. For more information, see the canMessageReplayBlockStruct reference page.

J1939 and CAN Signal Values No Longer Clipped by Database Limits

When encoding or decoding signals to or from messages on the CAN bus, MATLAB and Simulink no longer clip the signal values at the minimum and maximum defined in the database file. This updated behavior is now consistent with the J1939 protocol.

Version History

In past releases, the signal values were clipped to fall within the range defined by the database file minimum and maximum. Signals are no longer automatically clipped in this way.

R2016b

Version: 3.2

New Features

Bug Fixes

Measurement Data Format (MDF) File Support: Import data and attachments stored in MDF-files

Vehicle Network Toolbox supports reading data and saving attachments from Measurement Data Format (MDF) files. The following new functions provide these capabilities.

```
mdf
read
saveAttachment
```

For more information and examples, see Measurement Data Format (MDF).

Vehicle CAN Bus Monitor: Monitor CAN bus traffic with improved app performance

The Vehicle CAN Bus Monitor performance is improved in this release. Open the app with the `canTool` function.

R2016a

Version: 3.1

New Features

Bug Fixes

Version History

64-bit Hardware Support: Access and analyze data acquired from National Instruments NI-XNET hardware using 64-bit MATLAB

Vehicle Network Toolbox now supports analysis of CAN data from National Instruments NI-XNET hardware.

InitialTimestamp Property: Easily compute the absolute time of your CAN messages with the new property available on a CAN channel object

The new `InitialTimestamp` property of the CAN channel object enables you to compute the absolute time of CAN message data and signal values.

Synchronization and Triggering: Start National Instruments NI-XNET CAN channels via external triggers, and synchronize with National Instruments DAQ devices

Vehicle Network Toolbox now supports hardware triggering from National Instruments NI-XNET devices and National Instruments DAQ devices.

You can now trigger the start of an NI-XNET CAN device channel, and synchronize the start with an NI DAQ device.

You can configure synchronization and triggering with the `StartTriggerTerminal` property on the NI-XNET CAN channel object. This property string value configures a synchronization-trigger connection to start the NI-XNET CAN channel on the connected source terminal.

New properties of National Instruments XI-NET CAN channel objects

The following properties of NI-XNET CAN channel objects are now supported:

- `BusStatus` — Protocol level status of the network indicating potential error or bus off states
- `ReceiveErrorCount` — Number of receive error events detected by the device at the protocol level
- `TransmitErrorCount` — Number of transmit error events detected by the device at the protocol level
- `OnboardTermination` — Configure the on-board termination of the NI-XNET device that controls bus termination

Discontinued support for NI-CAN devices

Beginning in this release, Vehicle Network Toolbox no longer supports National Instruments NI-CAN devices.

Version History

This release does not support NI-CAN devices. To continue using NI-CAN devices, use an earlier version of Vehicle Network Toolbox. Alternatively, consider using a device supported in this release. For more information, see CAN Bus Support from Vehicle Network Toolbox.

R2015b

Version: 3.0

New Features

Bug Fixes

Support for J1939 protocol

This release of Vehicle Network Toolbox supports the SAE J1939 vehicle bus. This support includes:

- A command-line set of operations for access to J1939 channels and parameter groups. For more information on these, use the following MATLAB command:

  ```
  help vnt
  ```
- A J1939 Communication block library, available in the Simulink Library Browser, under the Vehicle Network Toolbox node.

Support for reading additional information from CANdb database files

For a full listing, see the `CAN database operation` category in the output from the MATLAB command:

```
help vnt
```

Support for additional Kvaser hardware

This release offers expanded support for Kvaser devices for the Memorator, PCIEcan, Leaf, USBcan, and BlackBird families; and added support for Ethercan devices.

R2015a

Version: 2.4

New Features

Bug Fixes

Significant decrease in memory usage for CAN messages

CAN messages in the MATLAB workspace use significantly less memory.

CAN message import speed doubled

You can now import CAN messages faster.

R2014b

Version: 2.3

New Features

Bug Fixes

Four times faster execution time for canMessageImport function

You can import messages into Vehicle Network Toolbox four times faster than before using the `canMessageImport`.

Optimized code generation for CAN blocks

CAN Simulink blocks are now optimized for code generation. Improvements include:

- Generated code is optimized and no longer contains mismatched data type comparisons.
- Improved MISRA compliance.
- Optimized constraint checks to avoid unnecessary saturation of code.
- Easier integration of custom transmit and receive blocks into existing applications.

R2014a

Version: 2.2

New Features

Bug Fixes

Support for Kvaser, National Instruments, PEAK-System, and Vector CAN interface devices via Support Package Installer

You can now install these supported CAN devices via the Support Package Installer

- Kvaser
- NI-CAN
- NI-XNET
- PEAK-System
- Vector

Support for Additional Vector Devices

You can now use Vehicle Network Toolbox with Vector VN8900 interface family devices.

R2013b

Version: 2.1

New Features

Support for additional National Instruments CAN devices that use NI-XNET driver

You can now use Vehicle Network Toolbox with NI CAN devices that use the NI-XNET driver including NI PCI-8511, NI PCI-8512, NI PCI-8513, NI PXI-8511, NI PXI-8512, and NI PXI-8513.

Support for PEAK-System CAN interface devices

You can now use Vehicle Network Toolbox with PEAK-System devices including PCAN-USB, PCAN-USB PRO, PCAN-ExpressCard, PCAN-PCI, PCAN-PCI Express, and PCAN-cPCI families.

Support for additional Kvaser devices

You can now use Vehicle Network Toolbox with Kvaser Blackbird V2, Kvaser Leaf Light V2, and Kvaser Mini PCI Express devices.

Support for additional Vector devices

You can now use Vehicle Network Toolbox Vector VN7600 device.

R2013a

Version: 2.0

New Features

Bug Fixes

XCP on CAN protocol support

You can now use XCP over CAN to communicate with your vehicle network.

MATLAB functions for transmitting and receiving XCP messages

You can now use new MATLAB functions to transmit and receive messages on your vehicle network using XCP.

Simulink XCP blocks for connecting a model to a CAN bus or ECU

You can now use new Simulink XCP blocks to build a model that communicates with a CAN bus or an ECU.

XCP MATLAB functions and Simulink blocks for reading and writing dynamic DAQ and STIM lists

You can now use new MATLAB functions and Simulink XCP blocks to read data acquisition lists and write data stimulation lists to an XCP channel.

XCP seed and key support for controlling access to a secured ECU

You can now use seed and key security to control access to your ECU using XCP over CAN.

A2L description file support

You can parse an A2L file and also use it to create XCP channels connected to a CAN device.

R2012b

Version: 1.7

Bug Fixes

Version History

Old Message Filtering Functions Removed

The following filtering functions have been removed:

- `filterAcceptRange`
- `filterBlockRange`
- `filterReset`
- `filterSet`

Use these new message filtering functions instead:

- `filterAllowAll`
- `filterAllowOnly`
- `filterBlockAll`

Compatibility Considerations

You cannot access any of the old message filtering functions. Replace all instances of the old functions with the new functions.

Support for Vector VN1600 series CAN interface devices

You can now use the VN1600 series devices for CAN communication using the Vehicle Network Toolbox.

R2012a

Version: 1.6

New Features

Bug Fixes

Version History

New UserData Property

You can now attach custom information to a CAN channel, message, or database using the new `UserData` property.

New discard Function

You can now use the `discard` function to clear all available messages on a CAN channel without restarting the channel.

Enhanced Sampling Time Field in Vehicle Network Toolbox Simulink Blocks

You can now use MATLAB variables to represent sampling time in Vehicle Network Toolbox Simulink blocks.

receiveRaw Function No Longer Available

You can no longer access `receiveRaw` and `canMessageCompatibilityMode` functions in the MATLAB workspace.

Version History

You cannot access direct signal properties as of R2012a of Vehicle Network Toolbox. Update your code to access physical signal values through the `Signals` property of message objects.

New Hardware Support for Kvaser Devices

New hardware support was added for Kvaser Memorator R SemiPro, USBcan R, and Eagle devices.

R2011b

Version: 1.5

New Features

Bug Fixes

Version History

New CAN Message Replay and Logging Simulink Blocks

You can now log and replay CAN messages from your Simulink Model using the CAN Log and CAN Replay blocks.

New CAN Message Filtering Functions

You can use these functions to filter standard and extended CAN messages:

- `filterAllowAll`
- `filterAllowOnly`
- `filterBlockAll`

Version History

The following message filtering functions will be removed in a future release:

- `filterAcceptRange`
- `filterBlockRange`
- `filterReset`
- `filterSet`

You will see a warning when you use these functions. Use these new filtering functions on page 22-2 instead.

New Platform Support for 64-bit Windows for Vector and Kvaser Devices

You can now use Vector and Kvaser CAN devices with Vehicle Network Toolbox on a 64-bit Windows system.

Support Added for Custom Receive and Transmit Blocks

You can now create your own Receive and Transmit blocks that communicate with unsupported hardware. You can use these custom blocks with the available Vehicle Network Toolbox Simulink blocks.

Enhanced canMessageImport Function

The `canMessageImport` function now import messages faster from Vector and Kvaser log files.

R2011a

Version: 1.4

New Features

Bug Fixes

Version History

New Hardware Support for National Instruments CAN Devices

Vehicle Network Toolbox supports NI CAN devices. You can use these devices for CAN communication in the toolbox and with CAN blocks in Simulink.

Note For a complete list of supported hardware, visit the Vehicle Network Toolbox product page at the MathWorks Web site `Supported Hardware` page.

Message Filtering Option in the CAN Tool

You now have the option to filter incoming messages in the CAN Tool and display only the messages you want to see, using message IDs. See Monitoring CAN Message Traffic for more information.

Deployable Code Generation in the CAN Blocks

You can now generate code from Simulink blocks in Vehicle Network Toolbox. Models containing the CAN Receive, CAN Transmit, and CAN Configuration blocks run successfully in Accelerator, Rapid Accelerator, External, and Deployed Modes. These blocks also now support using the Simulink Coder™ `packNGo` function.

receiveRaw function Removed

The `receiveRaw` function no longer works in the current release of Vehicle Network Toolbox.

Version History

You cannot access direct signal properties or use the `canMessageCompatibilityMode` function to restore access to direct signal properties starting with the current release of Vehicle Network Toolbox. Update your code to access physical signal values through the `Signals` property of message objects.

R2010b

Version: 1.3

New Features

Bug Fixes

New canMessageImport Function

The new `canMessageImport` function allows you to import CAN messages from Vector or Kvaser CAN log files.

Enhanced CAN Tool

The CAN Tool includes many enhancements such as:

- A redesigned, flexible interface
- Ability to decode messages attached to CAN database files

Enhanced CAN Channel for Event-Based Message Transmission

Use the `transmitEvent` function in MATLAB to enable messages for event-based transmission.

Enhanced CAN Channel for Periodic Message Transmission

Use the `transmitPeriodic` function in MATLAB to enable messages for periodic transmit. You can also configure the CAN Transmit block to send messages periodically.

New Hardware Support for Kvaser Devices

Vehicle Network Toolbox supports the following additional Kvaser CAN devices:

- PCIEcan HS/HS
- PCIEcan HS

R2010a

Version: 1.2

New Features

Bug Fixes

Version History

New Hardware Support for Kvaser Devices

Vehicle Network Toolbox supports Kvaser CAN devices, including these products:

BlackBird	PC104
LAPcan	PC104+
LAPcan II	PCcan
Leaf	PCIcan
Leaf Light	PCIcanx
Leaf Light Rugged	PCIcan II
Leaf Professional	PCIcanx II
Leaf SemiPro	USBcan
Memorator	USBcan II
Memorator II	USBcan Rugged
Memorator Professional	USBcan Professional

You can use these devices with the Vehicle Network Toolboxtm_vehiclenetworktoolbox; and the CAN block library.

For a complete list of devices, see the Supported Hardware page at the MathWorks Web site.

Enhanced Toolbox and Block Library Performance

The toolbox now transmits, receives, and processes messages at a faster rate. The Transmit and Receive blocks transmit and receive messages at a faster rate.

Removal of the receiveRaw Function

The `receiveRaw` function will be removed in a future release. In earlier software versions, the `receiveRaw` function expedited receive capability with minimal processing time and maximized message throughput. With enhanced toolbox performance this capability is no longer required. Performance of the standard `receive` function is significantly increased and you can use it to receive all messages.

Version History

The `receiveRaw` function will be removed in a future release. You will see a warning when you use this function. Use the `receive` function instead.

New Signals Property for CAN Message Functions

The `Signals` property displays each physical signal as defined for a CAN message in the CAN database. The `Signals` property is not applicable if you are not using a CAN database.

Signals displayed by the `Signals` property replaces access to direct signal property display of CAN messages. It has also increased the toolbox performance and the throughput of the message receive and transmit functions.

Version History

Direct signal properties will no longer appear in a message output. Use the new `Signals` property to access physical signal values. Update your code if you are using a CAN database and direct signal properties. Refer to the `Signals` property for more information. It is recommended that you update your code and use the new `Signals` property because all access to direct signal properties will be removed in a future release. If you cannot update your code at this time, you can use the `canMessageCompatibilityMode` function to restore access to direct signal properties.

New BusLoad Property

The `BusLoad` property displays the load on the CAN bus associated with the CAN message traffic on Kvaser devices.

Enhanced filterSet Function

The `filterSet` function now automatically calculates and applies the best case filter settings from an array of CAN IDs.

Enhanced replay Function

The `replay` function now replays messages from MATLAB to Simulink with their original timestamps without blocking Simulink operation.

R2009b

Version: 1.1

New Features

Bug Fixes

New Hardware Support for the Vector Informatik Device

Additional CAN hardware support is added for the CANcardXLe device.

R2009a

Version: 1.0

New Features

Bug Fixes

CAN Connectivity from MATLAB

You connect to a CAN network from the MATLAB interface with R2009a.

Vector CAN Database Support

Version 1.0 of the Vehicle Network Toolbox supports the Vector CAN database.

CAN Message Encoding and Decoding

You can encode and decode CAN traffic messages with Version 1.0 of the Vehicle Network Toolbox.

CAN Message Filtering

Version 1.0 of Vehicle Network Toolbox allows you to filter CAN messages by message ID.

CAN Message Logging and Replay

With Version 1.0 of Vehicle Network Toolbox you can log and replay CAN messages.

CAN Block Library

Version 1.0 of Vehicle Network Toolbox comes with a Simulink CAN block library that you can use to simulate CAN network models.

CAN Message Traffic GUI

Version 1.0 of Vehicle Network Toolbox provides a simple GUI that allows you to monitor CAN message traffic.

MATLAB Compiler Support

Version 1.0 of Vehicle Network Toolbox supports the MATLAB Compiler™. With this capability, you can use the MATLAB Compiler to take MATLAB files as input and generate redistributed, standalone applications that include Vehicle Network Toolbox functionality.