Communications Blockset Release Notes

Contents

Summary by Version

This table provides quick access to what's new in each version. For clarification, see "About Release Notes" on page 2.

Version (Release)	New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Latest Version V3.6 (R2007b)	Yes Details	Summary	Bug Reports Includes fixes.	Printable Release Notes: PDF Current product documentation
V3.5 (R2007a)	Yes Details	Yes Summary	Bug Reports Includes fixes.	No
V3.4 (R2006b)	Yes Details	Yes Summary	Bug Reports Includes fixes.	No
V3.3 (R2006a)	Yes Details	Yes Summary	Bug Reports Includes fixes.	No
V3.2 (R14SP3)	Yes Details	No	Bug Reports Includes fixes.	No
V3.1 (R14SP2)	Yes Details	Yes Summary	Bug Reports Includes fixes.	No
V3.0.1 (R14SP1)	Yes Details	Yes Summary	Fixed bugs	No
V3.0 (R14)	Yes Details	Yes Summary	Fixed bugs	No
V2.5 (R13)	Yes Details	Yes Summary	Fixed bugs and known problems	No
V2.0.1 (R12.1)	Yes Details	Yes Summary	Fixed bugs	No
V2.0 (R12)	Yes Details	Yes Summary	Fixed bugs and known problems	No

About Release Notes

Use release notes when upgrading to a newer version to learn about new features and changes, and the potential impact on your existing files and practices. Release notes are also beneficial if you use or support multiple versions.

If you are not upgrading from the most recent previous version, review release notes for all interim versions, not just for the version you are installing. For example, when upgrading from V1.0 to V1.2, review the New Features and Changes, Version Compatibility Considerations, and Bug Reports for V1.1 and V1.2.

New Features and Changes

These include

- New functionality
- Changes to existing functionality
- Changes to system requirements (complete system requirements for the current version are at the MathWorks Web site)
- Any version compatibility considerations associated with each new feature or change

Version Compatibility Considerations

When a new feature or change introduces a reported incompatibility between versions, its description includes a **Compatibility Considerations** subsection that details the impact. For a list of all new features and changes that have reported compatibility impact, see the “Compatibility Summary for Communications Blockset” on page 77.

Compatibility issues that are reported after the product has been released are added to Bug Reports at the MathWorks Web site. Because bug fixes can sometimes result in incompatibilities, also review fixed bugs in Bug Reports for any compatibility impact.

Fixed Bugs and Known Problems

MathWorks Bug Reports is a user-searchable database of known problems, workarounds, and fixes. The MathWorks updates the Bug Reports database as new problems and resolutions become known, so check it as needed for the latest information.

Access Bug Reports at the MathWorks Web site using your MathWorks Account. If you are not logged in to your MathWorks Account when you link to Bug Reports, you are prompted to log in or create an account. You then can view bug fixes and known problems for R14SP2 and more recent releases.

The Bug Reports database was introduced for R14SP2 and does not include information for prior releases. You can access a list of bug fixes made in prior versions via the links in the summary table.

Related Documentation at Web Site

Printable Release Notes (PDF). You can print release notes from the PDF version, located at the MathWorks Web site. The PDF version does not support links to other documents or to the Web site, such as to Bug Reports. Use the browser-based version of release notes for access to all information.

Product Documentation. At the MathWorks Web site, you can access complete product documentation for the current version and some previous versions, as noted in the summary table.

Version 3.6 (R2007b) Communications Blockset

This table summarizes what's new in Version 3.6 (R2007b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug Reports Includes fixes.	Printable Release Notes: PDF Current product documentation

New features and changes introduced in this version are

BPSK, MPSK, and QPSK Demodulator Blocks Enhanced

The BPSK Demodulator Baseband and QPSK Demodulator Baseband blocks are enhanced with fixed-point capabilities and run significantly faster in many hard-decision use cases. The M-PSK Demodulator Baseband block is enhanced with fixed-point capabilities for M=8. See individual reference pages for floating-point and fixed-point signal-flow diagrams.

Reed Solomon and BCH Blocks Enhanced with Punctures and Erasures

The Integer-Input RS Encoder, Binary-Input RS Encoder, and BCH Encoder blocks are enhanced with punctures.

The Integer-Output RS Decoder, Binary-Output RS Decoder, and BCH Decoder blocks are enhanced with punctures and erasures.

New Demos

`commrseras` illustrates Reed-Solomon coding with erasures, punctures, and shortening.

`commdvbc` models part of the ETSI (European Telecommunications Standards Institute) EN 300 429 standard for cable system transmission of digital television signals.

`commncopnseqdither` illustrates the use of pseudorandom noise sequences and multi-stage phase quantization in the design of numerically controlled oscillators. Simulink Fixed Point is needed to run this demo.

`commwman80216d_stbc` represents an end-to-end baseband model of the physical layer of a wireless metropolitan area network (WMAN) according to the IEEE 802.16-2004 standard.

These can be accessed through the demos pane of the Help browser, or by typing the model name (e.g., `commncopnseqdither`) at the command line.

PN Sequence Generator Block Enhanced

PN Sequence Generator block is enhanced with bit-packing options and smallest integer output data type mode.

TCM Encoder Blocks Enhanced

The M-PSK TCM Encoder, Rectangular QAM TCM Encoder, and General TCM Encoder blocks are enhanced with options to terminate at the zero state.

Integer to Bit Converter and Bit to Integer Converter Blocks Enhanced

The Integer to Bit Converter and Bit to Integer Converter blocks can now handle **Number of bits per integer** values up to 32.

Find Delay Block Enhanced

The Find Delay block is enhanced to support unsigned integer inputs.

TCM Decoder Blocks Enhanced

The algorithm of the TCM decoder blocks (M-PSK TCM Decoder, Rectangular QAM TCM Decoder, and General TCM Decoder) is updated to be consistent with the Viterbi Decoder block's decoding algorithm.

Compatibility Considerations

Due to the stochastic nature of the output of these blocks, the exact results from these blocks may be different from those of previous releases.

Version 3.5 (R2007a) Communications Blockset

This table summarizes what's new in Version 3.5 (R2007a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug Reports Includes fixes.	No

New features and changes introduced in this version are

New LDPC Encoder and Decoder Blocks

The LDPC Encoder and LDPC Decoder blocks are added. These are based on the same algorithms used in the Communications Toolbox functions fec.ldpcenc and fec.ldpcdec.

Fixed-Point Capabilities Added to Rectangular QAM Demodulator Block

The Rectangular QAM Demodulator Baseband block now has fixed-point capabilities for hard-decision square QAM modes.

Bitwise Soft-Decision Outputs Enabled for General QAM Demodulator Block

Bitwise soft-decision outputs are enabled for the General QAM Demodulator Baseband block.

Various Blocks Enhanced to Accept Fixed-Point Data Types

The operational parameters of the following filters can now be adjusted for fixed-point data types:

- Raised Cosine Transmit Filter
- Raised Cosine Receive Filter
- Gaussian Filter
- Ideal Rectangular Pulse Filter
- Windowed Integrator

The Convolutional Encoder block now accepts and outputs `ufix1`.

The Error Rate Calculation is upgraded to accept inputs of fixed-point data types.

BCH Encoder and Decoder Blocks Allow User Specification of Primitive and Generator Polynomials

The BCH Encoder and BCH Decoder blocks now allow user specification of the primitive and generator polynomials.

They have also been enhanced to speed up initialization.

Default Output Data Type Settings Added to AM/PM Demodulator Blocks

The option `'Inherit via internal rule'` is added to the output data-type option, and is set to be the default setting in the following blocks:

- M-PAM Demodulator Baseband
- Rectangular QAM Demodulator Baseband
- General QAM Demodulator Baseband
- M-PSK Demodulator Baseband
- BPSK Demodulator Baseband
- QPSK Demodulator Baseband
- OQPSK Demodulator Baseband
- M-DPSK Demodulator Baseband
- DBPSK Demodulator Baseband
- DQPSK Demodulator Baseband

See individual block reference pages for details.

There are no compatibility issues, as only new instances of the block will be set to the default output, leaving the settings of currently used blocks unchanged.

New Default Output Data Type Option Added to Viterbi Decoder Block

The option `'Inherit via internal rule'` is added to the output data-type option in the Viterbi Decoder block, and is set to be the default setting. See block reference page for details.

There are no compatibility issues, as only new instances of the block will be set to the default output, leaving the settings of currently used blocks unchanged.

M-FSK Modulator Block Errors When Used with Sample-Based Signals and Multitasking Solver

This change was made to ensure that the M-FSK Modulator block produces correct output.

Compatibility Considerations

With this change, a model with the M-FSK Modulator block will not run if it is used for sampled-based signals with **Tasking mode for periodic sample times** set to `Auto` or `MultiTasking`. For sample-based input signals, **Tasking mode for periodic sample times** (in **Simulation > Configuration Parameters > Solver**) should be set to `SingleTasking`.

PN Sequence Generator Block Allows Mask Bits from Input Port

An option is added to the PN Sequence Generator block that gives the choice of providing output mask information through either the block dialog or an input port.

Binary-Output RS Decoder and Binary-Input RS Encoder Blocks Enhanced

The Binary-Output RS Decoder and Binary-Input RS Encoder blocks now have the option of inheriting their output data type from their input.

Version 3.4 (R2006b) Communications Blockset

This table summarizes what's new in Version 3.4 (R2006b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug Reports Includes fixes.	No

New features and changes introduced in this version are

Fixed Point Capability Added to Various Blocks

Fixed point capability is added to the following blocks:

- M-PAM Modulator Baseband
- M-PAM Demodulator Baseband
- Rectangular QAM Modulator Baseband

- General QAM Modulator Baseband
- M-PSK Modulator Baseband
- BPSK Modulator Baseband
- QPSK Modulator Baseband
- OQPSK Modulator Baseband
- Viterbi Decoder

Bitwise Soft-Decision Outputs for the PSK and Rectangular QAM Demodulator

Bitwise soft-decision outputs are enabled for the M-PSK Demodulator Baseband, B-PSK Demodulator Baseband, QPSK Demodulator Baseband, and Rectangular QAM Demodulator Baseband blocks.

BCH Encoder and Decoder Run Faster and Operate on Shortened BCH Codes

BCH Encoder and Decoder blocks run faster and are enhanced to operate on shortened BCH codes.

Fixed-Point MSK Demo

A demo that illustrates the baseband implementation of a fixed-point MSK modulator and demodulator is added. This can be accessed through the demos pane of the Help browser, or by typing `commmsk` at the command line.

Binary Error Pattern Generator Block Is Obsoleted

The `Binary Error Pattern Generator` block is obsoleted.

Compatibility Considerations

It now resides in the library `cbobsv3`, which is obsolete and may be removed in the future.

Version 1.5 Blocks Removed

All the libraries and blocks associated with Communications Blockset version 1.5 have been removed from the product. These libraries are as follows.

- `commanabbnd`
- `commanapbnd`
- `commblkcod`
- `commchan`
- `commcnvcod`
- `commdigbbnd`
- `commdigpbnd`
- `commsink`
- `commsource`
- `commsrccod`
- `commsync`
- `commutil`

Compatibility Considerations

These blocks cannot be used with the current version of the Communications Blockset. The current version of the blockset provides some of the functionality in upgraded blocks.

Obsolete Blocks

The blocks in the following table have been obsoleted since version 3.0.

To access each replacement block, type the library name listed in the **Replacement Block Library** column at the MATLAB® command line.

Obsolete Block	Obsolete Block Library	Replacement Block	Replacement Block Library
Binary Error Pattern Generator	cbobsv3	None	N/A
DSB AM Demodulator Passband	commanapbnd2	DSB AM Demodulator Passband	commanapbnd3
DSB AM Modulator Passband	commanapbnd2	DSB AM Modulator Passband	commanapbnd3
DSBSC AM Demodulator Passband	commanapbnd2	DSBSC AM Demodulator Passband	commanapbnd3
DSBSC AM Modulator Passband	commanapbnd2	DSBSC AM Modulator Passband	commanapbnd3
FM Demodulator Passband	commanapbnd2	FM Demodulator Passband	commanapbnd3
FM Modulator Passband	commanapbnd2	FM Modulator Passband	commanapbnd3
PM Demodulator Passband	commanapbnd2	PM Demodulator Passband	commanapbnd3
PM Modulator Passband	commanapbnd2	PM Modulator Passband	commanapbnd3
SSB AM Demodulator Passband	commanapbnd2	SSB AM Demodulator Passband	commanapbnd3
SSB AM Modulator Passband	commanapbnd2	SSB AM Modulator Passband	commanapbnd3
Rayleigh Fading Channel	commchan2	Multipath Rayleigh Fading Channel	commchan3
Rician Fading Channel	commchan2	Multipath Rician Fading Channel	commchan3
M-PAM Demodulator Baseband	commdigbbndam2	M-PAM Demodulator Baseband	commdigbbndam3

Obsolete Block	Obsolete Block Library	Replacement Block	Replacement Block Library
M-PAM Modulator Baseband	commdigbbndam2	M-PAM Modulator Baseband	commdigbbndam3
Rectangular QAM Demodulator Baseband	commdigbbndam2	Rectangular QAM Demodulator Baseband	commdigbbndam3
Rectangular QAM Modulator Baseband	commdigbbndam2	Rectangular QAM Modulator Baseband	commdigbbndam3
General QAM Demodulator Baseband	commdigbbndam2	General QAM Demodulator Baseband	commdigbbndam3
General QAM Modulator Baseband	commdigbbndam2	General QAM Modulator Baseband	commdigbbndam3
BPSK Demodulator Baseband	commdigbbndpm2	BPSK Demodulator Baseband	commdigbbndpm3
BPSK Modulator Baseband	commdigbbndpm2	BPSK Modulator Baseband	commdigbbndpm3
DBPSK Demodulator Baseband	commdigbbndpm2	DBPSK Demodulator Baseband	commdigbbndpm3
DBPSK Modulator Baseband	commdigbbndpm2	DBPSK Modulator Baseband	commdigbbndpm3
DQPSK Demodulator Baseband	commdigbbndpm2	DQPSK Demodulator Baseband	commdigbbndpm3
DQPSK Modulator Baseband	commdigbbndpm2	DQPSK Modulator Baseband	commdigbbndpm3
M-DPSK Demodulator Baseband	commdigbbndpm2	M-DPSK Demodulator Baseband	commdigbbndpm3
M-DPSK Modulator Baseband	commdigbbndpm2	M-DPSK Modulator Baseband	commdigbbndpm3
M-PSK Demodulator Baseband	commdigbbndpm2	M-PSK Demodulator Baseband	commdigbbndpm3

Obsolete Block	Obsolete Block Library	Replacement Block	Replacement Block Library
M-PSK Modulator Baseband	commdigbbndpm2	M-PSK Modulator Baseband	commdigbbndpm3
OQPSK Demodulator Baseband	commdigbbndpm2	OQPSK Demodulator Baseband	commdigbbndpm3
OQPSK Demodulator Baseband	commdigbbndpm2	OQPSK Demodulator Baseband	commdigbbndpm3
QPSK Modulator Baseband	commdigbbndpm2	QPSK Modulator Baseband	commdigbbndpm3
QPSK Modulator Baseband	commdigbbndpm2	QPSK Modulator Baseband	commdigbbndpm3

Compatibility Considerations

Although they are currently still shipped with the product, they may be removed in future versions of the Communications Blockset. We recommend that you use the replacement blocks listed in the third column.

Version 3.3 (R2006a) Communications Blockset

This table summarizes what's new in Version 3.3 (R2006a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug Reports Includes fixes.	No

New features and changes introduced in this version are

Rician Channel Block Enhanced

The Multipath Rician Fading Channel block is updated with a new algorithm that is more accurate. The block can now simulate a line-of-sight Doppler component that is independent from the Doppler of the diffuse components. Sample time is now inherited.

Channel Visualization Added to Multipath Rician Fading Channel Block

Added an option to the Multipath Rician Fading Channel block that allows for use of the channel visualization tool.

Viterbi Decoder Block Updated with Puncturing and Erasing

The `Viterbi Decoder` block now decodes codewords with punctures and/or erasures.

Convolutional Encoder Block Updated

The `Convolutional Encoder` block now outputs punctured codewords.

It is also updated to allow trellis termination by appending tail bits.

M-PSK and Rectangular QAM Blocks Enhanced

The `M-PSK Modulator Baseband`, `M-PSK Demodulator Baseband`, `Rectangular QAM Modulator Baseband`, and `Rectangular QAM Demodulator Baseband` (only for square QAM) blocks now have an option for user-defined constellation mapping.

Demodulator Blocks Enhanced

The `M-PSK Demodulator Baseband`, `Rectangular QAM Demodulator Baseband` (for square QAM only), `BPSK Demodulator Baseband`, `QPSK Demodulator Baseband`, and `M-PAM Demodulator Baseband` blocks are enhanced to run significantly faster.

Additional C Data Type Support

Many blocks are updated to support C data types. See “Data Type Support” and individual block reference pages for details.

Phase/Frequency Offset Block Enhanced

The Phase/Frequency Offset block now accepts the frequency offset information through an input port.

The new block operates without any delay (the previous version had a sample delay of 1), and now accepts real inputs in addition to complex inputs. It accepts inputs of data type `double` or `single`, and its output data type matches that of its input.

Reed Solomon Decoder Block Enhanced

The Reed Solomon Decoder block is enhanced to run significantly faster.

DVBS2 Demo

The demo `dvbs2` is added, showcasing the state-of-the-art channel coding scheme used in the second generation Digital Video Broadcasting standard (DVB-S.2).

Obsolete Block Warning

Added a feature that warns you of the existence of obsolete Communications blocks when a model is opened.

Rician Fading Channel Block Is Obsoleted

The `Rician Fading Channel` block has been obsoleted. Its replacement, the Multipath Rician Fading Channel block, is found in `commchan3`.

Compatibility Considerations

It now resides in the library `commchan2`, which is obsolete and may be removed in the future.

SSB AM Passband Block Output

The SSB AM Modulator Passband and SSB AM Demodulator Passband blocks now output signals whose dimensions match those of their inputs.

Compatibility Considerations

Previously, inputs with dimensions `[1]` and `[1x1]` would result in outputs with dimension `[1]`. For such inputs, the blocks will now output signals with dimensions `[1]` and `[1x1]`, respectively.

saveas_commblks Obsoleted

The Communications Blockset has a `saveas_commblks` utility function to migrate models to previous releases. This is now obsoleted.

Compatibility Considerations

Simulink® provides a similar functionality through its **Save As...** option in the **File** menu of its models, as well as its function `save_system`.

Although `saveas_commblks` will continue to work in this release, we recommend that the Simulink functionality be used, as `saveas_commblks` will not be updated in the future.

Version 3.2 (R14SP3) Communications Blockset

This table summarizes what's new in Version 3.2 (R14SP3):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	No	Bug Reports Includes fixes.	No

New features and changes introduced in this version are

Analog Modulation Library Enhancement

The blocks in the Analog Modulation library now implement the formulas used in the Communications Toolbox to produce more consistent results.

Many Blocks Updated to Generate Embeddable Real-Time Workshop C-Code

Many blocks are updated to produce optimized embeddable Real-Time Workshop® C-code. See "Communications Block Properties" for details.

Many Blocks Updated to Work Within Triggered Subsystems

Many blocks are updated to work within triggered subsystems. See "Communications Block Properties" for details.

Additional C Data Type Support

Many blocks are updated to support C data types. See "Data Type Support" and individual block reference pages for details.

Two Parameters Now Tunable for Error Rate Calculation Block

The two parameters **Target number of errors** and **Maximum number of symbols** are now tunable during simulations and for RSIM executables.

New Demo of Timing Recovery Using Fixed-Rate Resampling

A new demo, `timrec_resample`, illustrates symbol timing adjustments using interpolation and numerically-controlled oscillator (NCO) based control as part of clock recovery in a digital modem.

Version 3.1 (R14SP2) Communications Blockset

This table summarizes what's new in Version 3.1 (R14SP2):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Bug Reports Includes fixes.	No

New features and changes introduced in this version are

Channel Visualization Added to Multipath Rayleigh Fading Channel Block

Added a channel visualization option to the Multipath Rayleigh Fading Channel block, which allows for use of the new channel visualization tool.

Multipath Rayleigh Fading Channel Block Performance Improvement

Increased the signal processing speed for the Multipath Rayleigh Fading Channel block by a factor of 2 to 10.

26 Blocks Generate Embeddable Real-Time Workshop C-code

In this release, the following blocks are now inlined, and will produce optimized embeddable Real-Time Workshop C-code. See “Communications Block Properties” for details.

- AWGN Channel and Multipath Rayleigh Fading Channel blocks in the Channels library
- Integrate and Dump block in the Communications Filters library
- Error Rate Calculation block in the Communications Sinks library
- Poisson Integer Generator block in the Random Data Sources sublibrary
- Kasami Sequence Generator and PN Sequence Generator blocks in the Sequence Generators sublibrary
- Convolutional Encoder and Viterbi Decoder blocks in the Convolutional Coding sublibrary
- Binary-Input RS Encoder, Integer-Input RS Encoder, Binary-Output RS Decoder, and Integer-Output RS Decoder blocks in the Block Coding sublibrary
- All blocks in the Block Interleaving sublibrary
- All blocks in the Convolutional Interleaving sublibrary
- All blocks in the Digital Baseband Amplitude Modulation sublibrary
- All blocks in the Digital Baseband Frequency Modulation sublibrary

- All blocks in the Digital Baseband Phase Modulation sublibrary
- Interlacer, Deinterlacer, Puncture, Insert Zero, and Derepeat blocks in the Sequence Operations library
- Bit-to-Integer Converter and Integer-to-Bit Converter blocks in the Utilities library

C Data Type Support for 26 Blocks

Added Native C data type support for the same 26 S-functions listed above. Updated six complete demos to include this new support.

Some Blocks Now Work in a Triggered Subsystem

Updated the following blocks so that they will now work in a triggered subsystem:

- Integrate and Dump block in the Communications Filters library
- Error Rate Calculation block in the Communications Sinks library
- All blocks in the Random Data Sources sublibrary
- All blocks in the Noise Generators sublibrary
- All blocks in the Sequence Generators sublibrary
- All blocks in the Block Coding sublibrary
- Convolutional Encoder and Viterbi Decoder blocks in the Convolutional Coding sublibrary
- Helical Interleaver and Helical Deinterleaver blocks in the Convolutional Interleaving sublibrary
- All blocks in the Digital Baseband Amplitude Modulation sublibrary
- All blocks in the Digital Baseband Phase Modulation sublibrary except for the OQPSK Modulator Baseband and OQPSK Demodulator Baseband blocks
- Interlacer, Deinterlacer, and Derepeat blocks in the Sequence Operations library

Note that triggered subsystems do not support multirate operation, so any mode of the above blocks that requires multirate operation will not work.

New Gardner Symbol Timing Recovery Demo

A new demo, `gardner_intdelay`, illustrates Gardner symbol timing recovery for a symbol frequency offset.

commdigbbndam2 and commdigbbndpm2 Libraries Updated

Compatibility Considerations

The libraries `commdigbbndam2` and `commdigbbndpm2` have been updated to `commdigbbndam3` and `commdigbbndpm3`, respectively. The modulators and demodulators in these new libraries do not have the **Samples per symbol** parameter.

Improvements and Changes to the Multipath Rayleigh Fading Channel Block

The Multipath Rayleigh Fading Channel block is now two to five times faster.

It now has options to output complex path gains and channel filter delay data.

Compatibility Considerations

It now accepts arbitrarily small Doppler (but not zero) and inherits sample time instead of setting it.

It now only accepts frame-based input, and thus does not support sample-based input. To work around this, use the frame conversion block of the Signal Processing blockset to reformat the signal. Note that the `Rician Fading Channel` block remains the same.

It now outputs the same waveform as the toolbox Rayleigh fading channel, and will therefore be a different waveform when compared to the same block in the previous version. Note that the statistical characteristics of the waveform should be identical to those of the previous version's block output.

Discrete-Time VCO Block

Compatibility Considerations

Changes are made to the Discrete-Time VCO block so that the phase accumulator wraps around and the block does not use a clock. The block does not allow continuous-time inputs.

Voltage Controlled Oscillator Block Renamed to Continuous-Time VCO

Compatibility Considerations

The Voltage Controlled Oscillator block was renamed to be consistent with the Discrete-Time VCO block.

CPM Modulator Baseband and GMSK Modulator Baseband Blocks

Compatibility Considerations

For the CPM Modulator Baseband block, the number of input symbols must be a factor of the length of the symbol prehistory parameter. This is true for cases when the input frame has fewer symbols than the length of the **Symbol prehistory** parameter. The block will now error out if this condition is not met. This also applies to the GMSK Modulator Baseband block, because it depends on the CPM Modulator Baseband block.

Version 3.0.1 (R14SP1) Communications Blockset

This table summarizes what's new in Version 3.0.1 (R14SP1):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Fixed bugs	No

New features and changes introduced in this version are

Source Block Dialog Boxes and the Model Explorer

In this release, the following blocks have been affected by changes in the behavior of source block dialog boxes and the Model Explorer.

- Binary Error Pattern Generator
- Gaussian Noise Generator
- Rayleigh Noise Generator
- Rician Noise Generator
- Uniform Noise Generator
- Bernoulli Binary Generator
- Poisson Integer Generator
- Random Integer Generator
- Barker Code Generator
- Gold Sequence Generator
- Hadamard Code Generator
- Kasami Sequence Generator
- OVSF Code Generator

- PN Sequence Generator
- Walsh Code Generator

Compatibility Considerations

See the section of the Simulink release notes corresponding to R14SP1.

Version 3.0 (R14) Communications Blockset

This table summarizes what's new in Version 3.0 (R14):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Fixed bugs	No

New features and changes introduced in this version are

Timing Phase Recovery

The blocks in the table below perform timing phase recovery, determining the best instant within a symbol period to sample a signal at the receiver. Sampling at the best instant improves the receiver's performance on a noisy signal. All blocks listed in the table are in the Timing Recovery sublibrary of the Synchronization library.

Block	Purpose
Early-Late Gate Timing Recovery	Recover the symbol timing phase using the early-late gate method
Gardner Timing Recovery	Recover the symbol timing phase using Gardner's method
MSK-Type Signal Timing Recovery	Recover the symbol timing phase using a fourth-order nonlinearity method
Mueller-Muller Timing Recovery	Recover the symbol timing phase using the Mueller-Muller method
Squaring Timing Recovery	Recover the symbol timing phase using a squaring method

For more information and an example, see "Timing Phase Recovery" in the Using the Communications Blockset documentation. For demos, enter `gardner_vfracdelay` or `msk_sync` in the MATLAB Command Window.

Carrier Phase Recovery

The blocks in the table below perform carrier phase recovery. They are in the Carrier Recovery sublibrary of the Synchronization library.

Block	Purpose
M-PSK Phase Recovery	Recover the carrier phase using the M-Power method
CPM Phase Recovery	Recover the carrier phase using the 2P-Power method

For more information and an example, see “Carrier Phase Recovery” in the Using the Communications Blockset documentation. For a demo, enter `msk_sync` in the MATLAB Command Window.

Equalizers

The blocks in the table below enable you to equalize a signal using a linear equalizer, a decision feedback equalizer, or a maximum-likelihood sequence estimation equalizer based on the Viterbi algorithm. All blocks listed in the table are in the Equalizers library.

Block	Purpose
CMA Equalizer	Equalize using the constant modulus algorithm
LMS Decision Feedback Equalizer	Equalize using a decision feedback equalizer that updates weights with the LMS algorithm
LMS Linear Equalizer	Equalize using a linear equalizer that updates weights with the LMS algorithm
MLSE Equalizer	Equalize using the Viterbi algorithm
Normalized LMS Decision Feedback Equalizer	Equalize using a decision feedback equalizer that updates weights with the normalized LMS algorithm
Normalized LMS Linear Equalizer	Equalize using a linear equalizer that updates weights with the normalized LMS algorithm

Block	Purpose
RLS Decision Feedback Equalizer	Equalize using a decision feedback equalizer that updates weights with the RLS algorithm
RLS Linear Equalizer	Equalize using a linear equalizer that updates weights with the RLS algorithm
Sign LMS Decision Feedback Equalizer	Equalize using a decision feedback equalizer that updates weights with the signed LMS algorithm
Sign LMS Linear Equalizer	Equalize using a linear equalizer that updates weights with the signed LMS algorithm
Variable Step LMS Decision Feedback Equalizer	Equalize using a decision feedback equalizer that updates weights with the variable-step-size LMS algorithm
Variable Step LMS Linear Equalizer	Equalize using a linear equalizer that updates weights with the variable-step-size LMS algorithm

For more information, see “Equalizers”. For an example, see the new Defense Communications: US MIL-STD-188-110B demo (`milstd_188110Bmodel`).

Filtering and Pulse Shaping

The blocks in the table below perform filtering and pulse shaping. All blocks listed in the table are in the Comm Filters library.

Block	Purpose
Gaussian Filter	Filter the input signal, possibly downsampling, using a Gaussian FIR filter
Ideal Rectangular Pulse Filter	Shape the input signal using ideal rectangular pulses

Block	Purpose
Raised Cosine Receive Filter	Filter the input signal, possibly downsampling, using a raised cosine FIR filter
Raised Cosine Transmit Filter	Upsample and filter the input signal using a raised cosine FIR filter

Trellis-Coded Modulation

The blocks in the table below perform trellis-coded modulation. All blocks listed in the table are in the TCM sublibrary of Digital Baseband Modulation, in the Modulation library.

Block	Purpose
General TCM Decoder	Decode trellis-coded modulation data, mapped using an arbitrary constellation
General TCM Encoder	Convolutionally encode binary data and map using an arbitrary constellation
M-PSK TCM Decoder	Decode trellis-coded modulation data, modulated using the PSK method
M-PSK TCM Encoder	Convolutionally encode binary data and modulate using the PSK method
Rectangular QAM TCM Decoder	Decode trellis-coded modulation data, modulated using the QAM method
Rectangular QAM TCM Encoder	Convolutionally encode binary data and modulate using the QAM method

Utility Blocks for Working with Delays

The blocks in the table below help you compute or manipulate the delay through one or more blocks in your model. This is especially useful when you are comparing two signals to compute error rates, or when you need to align boundaries of codewords or other groupings with Simulink frame boundaries. All blocks listed in the table are in the Utility Blocks library.

Block	Purpose
Align Signals	Align two signals by finding the delay between them
Find Delay	Find the delay between two signals

The reference pages for these blocks include examples of how to use them in a variety of situations.

Enhanced Source Coding Blocks

The new Quantizing Encoder and Quantizing Decoder blocks replace the older Sampled Quantizer Encode and Quantizer Decode blocks, which are obsolete. The new blocks perform scalar quantization encoding and decoding operations, respectively. The new blocks can process frame-based column vectors in addition to other kinds of vectors. The new Quantizing Encoder block does not require you to specify the vector length or sample time as parameters in the dialog box.

The older encoder produced a third output signal that represented the mean square distortion, while the new Quantizing Encoder block does not. The older encoder produced a discrete-time output signal even if its input was continuous-time, whereas the new Quantizing Encoder block preserves sample times.

AWGN Channel Enhancement for RSim Target

Selected parameters of the AWGN Channel block are now compatible with the Real-Time Workshop rapid simulation (RSim) target. This means that if you use Real-Time Workshop to build an RSim executable, then you can tune selected parameters without recompiling the model. This is useful for Monte Carlo simulations in which you run the simulation multiple times (perhaps

on multiple computers) with different amounts of noise. The table below indicates, for different modes of the block, which parameters are tunable.

Mode	Tunable Parameters
`Eb/No`	**Eb/No**, **Input signal power**
`Es/No`	**Es/No**, **Input signal power**
`SNR`	**SNR**, **Input signal power**
`Variance from mask`	**Variance**

For more information about the RSim target, see the Real-Time Workshop documentation set.

New Demos

New demos in Release 14 are listed in the table below. You can open the demos by finding them in the **Demos** pane of the MATLAB Help browser or by entering the corresponding model names in the MATLAB Command Window.

Title	Model Name
Convolutional Encoder with Uncoded Bits and Feedback	`conv_encoderdemo`
Soft-Decision GMSK Demodulator	`gmsk_softdecision`
Adjacent and Co-Channel Interference	`adjcochanint`
Adaptive Equalization Using Embedded MATLAB	`equalizer_eml`
Gardner Timing Phase Recovery	`gardner_vfracdelay`
MSK Signal Recovery	`msk_sync`
IEEE 802.11a WLAN Physical Layer	`wlan80211a`
Physical Layer Model of the cdma2000® Standard	`cdma2000_phlayer`
Defense Communications: US MIL-STD-188-110B	`milstd_188110Bmodel`

Demonstration models have also been reorganized into categories to make it easier for you to find relevant demos. You can view the categories using the **Demos** pane of the MATLAB Help browser.

Changes in BCH Encoder and BCH Decoder

The BCH Decoder block has been changed such that the second output port is optional and the error-correction capability is no longer a parameter. Also, this block and the BCH Encoder block no longer accept sample-based inputs.

Compatibility Considerations

If you built models with earlier versions of these two blocks, then you should

- Resave the models using Release 14, to avoid producing Simulink warnings.
- Revise the models so that the inputs to the BCH blocks are frame-based column vectors rather than sample-based vectors. To change the shape or frame status of a signal, you can use the Reshape block in Simulink, or the Frame Status Conversion block in the Signal Processing Blockset. Because the outputs from the BCH blocks are now frame-based column vectors, you might need to revise other parts of your model as well.

Changes in Fading Channel Blocks

Compatibility Considerations

The Multipath Rayleigh Fading Channel and Rician Fading Channel blocks are designed to process only sample-based scalars or frame-based column vectors. In Release 13, the blocks mistakenly accepted sample-based column vectors as input. In Release 14, the blocks correctly produce an error message if the input signal is a sample-based vector or a matrix.

Changes in Integrators

The new Communications Filters library contains a new Integrate and Dump block and a new Windowed Integrator block.

Compatibility Considerations

These blocks behave differently compared to the respective blocks of the same names in Release 13.

The new Integrate and Dump block

- Does not reduce the sum modulo a constant. The **Absolute value bound** parameter is not part of the new block.
- Does not require you to enter the sample time in the dialog box. The **Sample time** parameter is not part of the new block.
- Measures the **Integration period** parameter value in samples, not seconds.
- Can process sample-based scalars and frame-based matrices, but not sample-based vectors of length greater than 1. In a frame-based matrix, a given column is interpreted as a set of samples from a single channel.
- Can optionally discard a specified number of input samples at the beginning of the simulation. In frame-based mode, the number of samples to discard can be different for each channel (column) of the input matrix.
- Can optionally suppress the intermediate cumulative sums and output only the final sum.

The new Windowed Integrator block

- Does not require you to enter the sample time or vector size in the dialog box. The **Sample time** and **Input vector size** parameters are not part of the new block.
- Measures the integration period in samples, not seconds.
- Can process sample-based scalars and frame-based matrices, but not sample-based vectors of length greater than 1. In a frame-based matrix, a given column is interpreted as a set of samples from a single channel.
- Processes only discrete-time signals, not continuous-time signals.
- Uses cumulative sums as integrals and does not offer a choice of integration methods. The **Integration method** parameter is not part of the new block.

To learn more about the new blocks, see the Integrate and Dump and Windowed Integrator online reference pages, respectively.

Legacy Models Containing Integrator Blocks. If you built models with the older Integrate and Dump block or the older Windowed Integrator block, then the block is unchanged there. You can update the block manually by replacing it with the newer block from the Communications Filters library. You might need to change parameters or other parts of your model to make the new block fit into your model.

To find the older blocks in their default library setting, type `comminteg2` in the MATLAB Command Window.

Note The older Integrate and Dump block and the older Windowed Integrator blocks are obsolete and might be removed from a future release of the Communications Blockset.

Change in Error Rate Calculation Block

Compatibility Considerations

If you set **Output data** to `Workspace` in the Error Rate Calculation block, then the variable containing the output data resides in the base MATLAB workspace. In previous releases, the variable resided in the calling workspace.

This change is relevant if you invoke the simulation from a function. If you need to access the output data within the function, use `evalin`. For example, in a function, the command below accesses a variable called `ErrorVec` in the base MATLAB workspace and assigns its value to a variable by the same name in the function workspace.

```
ErrorVec = evalin('base','ErrorVec;');
```

If you invoke the simulation directly from the model window or by entering a `sim` command in the MATLAB Command Window, then the change in behavior of the Error Rate Calculation block does not affect you.

Version 1.3 Libraries Removed

Compatibility Considerations

The block libraries from the Communications Toolbox Version 1.3 (Release 10) are no longer installed as part of Release 14. The block libraries from the Communications Toolbox Version 1.5 (Release 11) might be removed from a future release.

Obsolete Blocks

Compatibility Considerations

The table below lists blocks from Release 13 that are obsolete as of Release 14. In particular, all digital passband modulation, digital passband demodulation, analog baseband modulation, and analog baseband demodulation blocks are obsolete. In place of digital passband blocks, use their digital baseband counterparts. In place of analog baseband blocks, use their analog passband counterparts.

Note For backward compatibility, the obsolete blocks in the table below are still provided in Release 14 in the *`matlabroot`*`/commblks/commblksobsolete` directory tree. However, they might be removed in a future release and it is recommended that you avoid using these obsolete blocks in your models.

Where applicable, the second column lists blocks that provide similar functionality. In some cases, the similar block requires different parameter settings, data formats, or signal attributes compared to the original block. Therefore, you should read the documentation for the similar block before using it in your model.

Obsolete Block	Similar Block(s), if Any
Continuous-Time Eye and Scatter Diagrams	Discrete-Time Eye Diagram Scope, Discrete-Time Scatter Plot Scope, Discrete-Time Signal Trajectory Scope
CPFSK Demodulator Passband	CPFSK Demodulator Baseband
CPFSK Modulator Passband	CPFSK Modulator Baseband
CPM Demodulator Passband	CPM Demodulator Baseband
CPM Modulator Passband	CPM Modulator Baseband
Discrete Modulo Integrator	
DPCM Decoder	
DPCM Encoder	
DSB AM Demodulator Baseband	DSB AM Demodulator Passband
DSB AM Modulator Baseband	DSB AM Modulator Passband
DSBSC AM Demodulator Baseband	DSBSC AM Demodulator Passband
DSBSC AM Modulator Baseband	DSBSC AM Modulator Passband
Enabled Quantizer Encode	Quantizing Encoder
FM Demodulator Baseband	FM Demodulator Passband
FM Modulator Baseband	FM Modulator Passband
General QAM Demodulator Passband	General QAM Demodulator Baseband
General QAM Modulator Passband	General QAM Modulator Baseband
GMSK Demodulator Passband	GMSK Demodulator Baseband
GMSK Modulator Passband	GMSK Modulator Baseband
M-DPSK Demodulator Passband	M-DPSK Demodulator Baseband
M-DPSK Modulator Passband	M-DPSK Modulator Baseband
M-FSK Demodulator Passband	M-FSK Demodulator Baseband
M-FSK Modulator Passband	M-FSK Modulator Baseband

Obsolete Block	Similar Block(s), if Any
Modulo Integrator	
M-PAM Demodulator Passband	M-PAM Demodulator Baseband
M-PAM Modulator Passband	M-PAM Modulator Baseband
M-PSK Demodulator Passband	M-PSK Demodulator Baseband
M-PSK Modulator Passband	M-PSK Modulator Baseband
MSK Demodulator Passband	MSK Demodulator Baseband
MSK Modulator Passband	MSK Modulator Baseband
OQPSK Demodulator Passband	OQPSK Demodulator Baseband
OQPSK Modulator Passband	OQPSK Modulator Baseband
PM Demodulator Baseband	PM Demodulator Passband
PM Modulator Baseband	PM Modulator Passband
Quantizer Decode	Quantizing Decoder
Rectangular QAM Demodulator Passband	Rectangular QAM Demodulator Baseband
Rectangular QAM Modulator Passband	Rectangular QAM Modulator Baseband
Sampled Quantizer Encode	Quantizing Encoder
SSB AM Demodulator Baseband	SSB AM Demodulator Passband
SSB AM Modulator Baseband	SSB AM Modulator Passband
Triggered Read From File	From File (Simulink)
Triggered Write to File	To File (Simulink)

Blocks Now in Different Library Locations

Compatibility Considerations

The table below lists blocks that reside in different libraries in Release 14, compared to Release 13. If you used these blocks in models that you saved in Release 13, then the blocks will still work in Release 14. However, you should

be aware of the changed locations in case you look for these blocks in Release 14 in the library windows or the Simulink Library Browser.

Block	Release 13 Location	Release 14 Location
Baseband PLL	Synchronization	Components sublibrary of Synchronization
Binary Error Pattern Generator	Data Sources sublibrary of Comm Sources	Noise Generators sublibrary of Comm Sources
Charge Pump PLL	Synchronization	Components sublibrary of Synchronization
Complex Phase Difference	Sequence Operations sublibrary of Basic Comm Functions	Utility Blocks
Complex Phase Shift	Sequence Operations sublibrary of Basic Comm Functions	Utility Blocks
Discrete-Time VCO	Controlled Sources sublibrary of Comm Sources	Components sublibrary of Synchronization
Integrate and Dump	Integrators sublibrary of Basic Comm Functions	Communications Filters
Linearized Baseband PLL	Synchronization	Components sublibrary of Synchronization
Phase-Locked Loop	Synchronization	Components sublibrary of Synchronization
Voltage-Controlled Oscillator	Controlled Sources sublibrary of Comm Sources	Components sublibrary of Synchronization
Windowed Integrator	Integrators sublibrary of Basic Comm Functions	Communications Filters

Utility Functions Library Renamed. The Utility Functions library is now called Utility Blocks.

Contents of Basic Comm Function Library Moved. The Basic Comm Functions library, which consisted of the Integrators sublibrary and the Sequence Operations sublibrary, is no longer in the Communications Blockset. Sequence Operations has become a top-level library. The Integrate and Dump block and the Windowed Integrator block, formerly in the Integrators sublibrary, are now in the Communications Filters library. The Discrete Modulo Integrator and Modulo Integrator blocks are now obsolete.

Changes in Block Dialog Boxes

A few blocks have renamed some of their parameters or made other dialog box changes.

Compatibility Considerations

Legacy models might issue warnings when you first open them with Release 14. After you resave the models with Release 14, the warnings will not recur. Specific changes are listed below.

Block	Release 13 Characteristic	Change in Release 14
BCH Decoder	**Show number of errors** check box	**Output number of corrected errors** check box
Binary-Output RS Decoder	**Output port for number of corrected errors** check box	**Output number of corrected errors** check box
Discrete-Time Eye Diagram Scope	Dialog box uses check boxes to show or hide groups of parameters	Dialog box uses tabbed panels to organize parameters
Discrete-Time Scatter Plot Scope		
Discrete-Time Signal Trajectory Scope		

Block	Release 13 Characteristic	Change in Release 14
Discrete-Time VCO	**Oscillation frequency** parameter	Renamed as **Quiescent frequency** parameter
Voltage-Controlled Oscillator		

Changes in commstartup Function

The `commstartup` function, which changes the default Simulink model settings to values more appropriate for the simulation of communication systems, has changed some of its settings.

Compatibility Considerations

When you run `commstartup`, it

- Changes the default solver to a discrete solver.
- Changes the default value of a Simulink diagnostic setting so that Simulink does not issue a warning when a source block uses an inherited sample time. Some Communications Blockset blocks internally inherit sample times, which can be a useful and valid modeling technique.

Simulation Settings of Legacy Models

Compatibility Considerations

Your legacy models might issue warnings if they use settings other than the ones listed in “Changes in commstartup Function” on page 45 above. You can suppress the warnings by changing certain settings and resaving the model.

Discrete Solver. If you have legacy models that issue a warning like

```
Warning: The model 'untitled' does not have continuous states,
hence using the solver 'VariableStepDiscrete' instead of the
solver 'ode45' specified in the Configuration Parameters dialog.
```

when you start the simulation in R14, then consider changing the solver to a discrete solver and resaving the model. To change the solver, use the

Configuration Parameters option on the model window's **Simulation** menu.

Sample Time of Source Blocks. Some Communications Blockset blocks internally inherit sample times, which can be a useful and valid modeling technique. If you have legacy models that issue a warning like

```
Warning: Source 'untitled/DSP Constant' specifies that its sample
time (-1) should be back-inherited. You should explicitly specify
the sample time of sources.
```

when you start the simulation in R14, then consider changing the diagnostic setting manually and resaving the model. To change the setting manually, choose the **Configuration Parameters** option on the model window's **Simulation** menu, expand **Diagnostics** in the left pane, select **Sample Time** in the left pane, and then set **Source block specifies -1 sample time** to `none` in the right pane.

Version 2.5 (R13) Communications Blockset

This table summarizes what's new in Version 2.5 (R13):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Fixed bugs and known problems	No

New features and changes introduced in this version are

RF Impairments Library

The new RF Impairments library contains blocks to simulate radio frequency (RF) impairments at the receiver. The blocks in the library are listed in the following table.

Block Name	Purpose
Free Space Path Loss	Reduce the amplitude of the input signal by the amount specified
I/Q Imbalance	Create a complex baseband model of the signal impairments caused by imbalances between in-phase and quadrature receiver components
Memoryless Nonlinearity	Apply a memoryless nonlinearity to a complex baseband signal
Phase/Frequency Offset	Apply residual phase and frequency offsets to a complex baseband signal
Phase Noise	Apply receiver phase noise to a complex baseband signal
Receiver Thermal Noise	Apply receiver thermal noise to a complex baseband signal

Sequence Generators Library

The Comm Sources library is now divided into four sublibraries for Version 2.5. Three of these sublibraries contain the blocks from the Version 2.0.1 Comm Sources library:

- Data Sources
- Noise Sources
- Controlled Sources

The fourth, the Sequence Generators sublibrary, contains the PN Sequence Generator block and five new blocks for Version 2.5. You can use the blocks in the Sequence Generators sublibrary to generate sequences for spreading or synchronization in a communication system. The following table lists the blocks in the Sequence Generators sublibrary.

Block Name	Purpose
Barker Code Generator	Generate a Barker Code
Gold Sequence Generator	Generate a Gold sequence from a set of sequences
Kasami Sequence Generator	Generate a Kasami sequence from the set of Kasami sequences
Hadamard Code Generator	Generate a Hadamard code from an orthogonal set of codes
OVSF Code Generator	Generate an orthogonal variable spreading factor (OVSF) code from a set of orthogonal codes
PN Sequence Generator	Generate a pseudonoise sequence
Walsh Code Generator	Generate a Walsh code from an orthogonal set of codes

Eye Diagram, ScatterPlot, and Signal Trajectory Scopes

The Version 2.0.1 Discrete-Time Eye and Scatter Diagram block, in the Comm Sinks library, has been replaced by three new blocks for Version 2.5, as described in the following table.

Block Name	Purpose
Discrete-Time Eye Diagram Scope	Display multiple traces of a modulated signal
Discrete-Time Scatter Plot Scope	Display a modulated signal in its signal space by plotting its in-phase component against its quadrature component
Discrete-Time Signal Trajectory Scope	Display a modulated signal in its signal space by plotting its in-phase component versus its quadrature component

These blocks greatly enhance the features of the Discrete-Time Eye and Scatter Diagram.

CRC Library

The Channel Coding library has been renamed the Error Correction and Detection library, and a new sublibrary, CRC, has been added to the Error Detection and Correction library. The CRC library contains new blocks for appending cylic redundancy check (CRC) bits to data and for detecting errors in transmission.

The following table lists the blocks in the CRC library.

Block Name	Purpose
CRC-N Generator	Generate CRC bits according to the selected CRC method and append them to input data
CRC-N Syndrome Detector	Detect errors in the input data according to the specified CRC method
General CRC Generator	Generate CRC bits according to the generator polynomial and append them to input data
General CRC Syndrome Detector	Detect errors in the input data according to the generator polynomial

Enhancements to Reed-Solomon Blocks

The following four blocks, in the Block sublibrary of the Error Detection and Correction Library, have new features:

- Binary-Input RS Endoder
- Binary-Input RS Decoder
- Integer-Input RS Encoder
- Integer-Input RS Decoder

You can now specify the primitive polynomial and generator polynomial, which are used to generate the codes. This enables you to use a much wider range of Reed-Solomon codes. There is also a new option to output the number of corrected errors from the Binary-Input RS Decoder and Integer-Input RS Decoder blocks.

New Demos

The Communications Blockset contains eleven new demos for Version 2.5. These include a large-scale demo model of a commercial application of a third generation (3G) wireless system using wide-band code division multiple access (WCDMA). The demo presents an end-to-end transmission between a base station and a mobile station, as specified by the Third Generation Partnership Project (3GPP).

The new demos are as follows:

- WCDMA End-to-End Physical Layer Demo
- WCDMA Coding and Multiplexing Demo
- WCDMA Spreading and Modulation Demo
- RF Satellite Link Demo
- HiperLAN/2 Demo
- Bluetooth Voice Transmission Demo
- Adaptive Equalization Demo
- CPM Phase Tree Demo
- GMSK vs. MSK Demo
- Filtered QPSK vs. MSK Demo
- Raleigh Fading Channel Demo

Enhancements to CPM Modulator Block

The CPM modulator block now enables you to specify both the entire pulse length and the pulse main lobe length when simulating an LSRC frequency pulse length. This feature enables you to simulate a modulation such as 3SRC6.

Fixed Bugs

Fading channels

The accuracy of Doppler spread of the Rayleigh and Rician Fading Channel Blocks has been improved. The blocks now give better results for high sampling rates and small Doppler frequencies, as specified in communication standards such as WCDMA.

List of demos for which code can now be generated using Real Time Workshop

The following demos now generate code using Real Time Workshop:

- `dmt_sim`
- `dvbt_sim`
- `tstgraycod`
- `phasenoise_sim`
- `dmt_alt_sim`
- `tstconvcod`

Passband FSK and CPM modulators and demodulators are more accurate

The passband FSK and CPM modulators have been modified to generate more accurate waveforms by performing FIR interpolation when upsampling. The FIR filters significantly reduce the levels of any spectral copies revealed by upsampling. They also introduce delay into the modulators.

Reed-Solomon blocks now encode and decode correctly

The Reed-Solomon blocks now encode and decode signals correctly.

Known Problems

Several Communications Blockset blocks are incompatible with Real-Time Workshop

Several Communications Blockset blocks are incompatible with Real-Time Workshop. As a result, Real-Time Workshop cannot generate code for models that include these blocks:

- Continuous-Time Eye and Scatter Diagrams
- Triggered Read from File
- Triggered Write to File
- Integer-Input RS Encoder
- Integer-Output RS Decoder
- Binary-Input RS Encoder
- Binary-Output RS Decoder
- Blocks in the CPM sublibrary of the Digital Baseband sublibrary of the Modulation library

Several Communications Blockset demos are incompatible with Real-Time Workshop

Several Communications Blockset demos are incompatible with Real-Time Workshop. As a result, Real-Time Workshop cannot generate code for these demos:

- 256 Channel ADSL
- Adaptive Equalization
- Bluetooth Voice Transmission
- CPM Phase Tree Example
- Digital Video Broadcasting Model
- Discrete Multitone Signaling
- Filtered QPSK vs. MSK

- GMSK vs. MSK
- WCDMA Coding and Multiplexing Example
- WCDMA End-to-End Physical Layer
- WCDMA Spreading and Modulation Example

Old Models Using the Baseband or Passband SSB Modulators Must Be Resaved

Compatibility Considerations

The baseband and passband SSB modulators have been updated for Release 13 to include a pop-up menu enabling you to choose between upper and lower sideband modulation. You should resave any models using the old SSB modulators before running them in Release 13, to avoid producing Simulink warnings.

Change the Boolean Logic Signals Parameter to Off

The Communications Blockset does not support signals with `boolean` data type.

Compatibility Considerations

In Release 13, the Simulink **Boolean logic signals** parameter is now set to `On` by default. If you use Simulink blocks such as the Logical Operator block together with Communications Blockset blocks in a model, you must change the default setting of the **Boolean logic signals** parameter setting to `Off`. To do so, enter

```
commstartup
```

at the beginning of each MATLAB session, before you create a model. This sets the **Boolean logic signals** parameter to `Off` for every model you create during the current MATLAB session.

To manually change the **Boolean logic signals** parameter in a model to `Off`, do the following steps:

1. Select **Simulation parameters** from the model window's **Simulation** menu.
2. Click the **Advanced** tab in the **Simulation Parameters** dialog box.
3. Select **Boolean logic signals** in the **Optimizations** field.
4. Under **Action**, select the **Off** check box.
5. Click **OK**.

Note that this changes the **Boolean logic signals** parameter to `Off` only for the current model.

Because the default setting of the **Boolean logic signals** parameter prior to Release 13 was `Off`, it is not necessary to make changes to models that you created prior to Release 13.

Version 2.0.1 (R12.1) Communications Blockset

This table summarizes what's new in Version 2.0.1 (R12.1):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Fixed bugs	No

New features and changes introduced in this version are

Setting Simulink Preferences Automatically

The new `commstartup.m` script sets certain Simulink preferences to values that are most appropriate for the simulation of communication systems. To use this script, type the command `commstartup` in your `startup.m` file or in the MATLAB Command Window.

Converting Between Bipolar and Unipolar Signals

The Utility Functions library contains new blocks that convert between bipolar and unipolar signals. The blocks are Bipolar to Unipolar Converter and Unipolar to Bipolar Converter.

Choosing Seeds for Random-Output Blocks

The `randseed` function is a new function that generates prime numbers for use as **Initial seed** parameters in blocks that produce random output. Compared to composite seeds, prime seeds yield output that has better statistical properties.

Using Error Counts to Control Simulation Duration

You can now configure the Error Rate Calculation block so that it automatically stops the simulation upon detecting a specified number of

errors. You do not need to know in advance how long it will take to accumulate that many errors.

Choosing the Algorithm for Integrator Blocks

The Discrete Modulo Integrator block now allows you to choose the integration method using a mask parameter. The corresponding mask parameter in the Windowed Integrator block has changed its name from **Method** to **Integration method** for consistency with other integration blocks.

Fixed Bugs

- The M-FSK Baseband Modulator, M-FSK Baseband Demodulator, M-FSK Passband Modulator, and M-FSK Passband Demodulator blocks now use the correct tone spacing.
- The PN Sequence Generator block now generates only binary values, and the numbers in the sequence do not depend on the frame status or size.

Binary Symmetric Channel Block

Compatibility Considerations

The Binary Symmetric Channel block dialog box now omits the **Input vector length** and **Sample time** parameters because the block now determines these quantities automatically. However, if you open a model in Release 12.1 that contains the Release 12.0 Binary Symmetric Channel block, then the Command Window might display warnings about block parameters. To suppress these warnings in the future, simply save the model from Release 12.1.

Digital Passband Modulation Blocks

Compatibility Considerations

Any model that includes a digital passband modulator block or a digital passband demodulator block must use a variable-step solver rather than a fixed-step solver. To configure a model so that it uses a variable-step solver,

select **Simulation parameters** from the model window's **Simulation** menu and then set the **Type** parameter on the **Solver** panel to `Variable-step`.

Version 2.0 (R12) Communications Blockset

This table summarizes what's new in Version 2.0 (R12):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as **Compatibility Considerations**, below. See also Summary.	Fixed bugs and known problems	No

New features and changes introduced in this version are

Digital Modulation Libraries

The digital modulation libraries have been replaced with new ones. The new libraries contain baseband and passband sublibraries for

- Amplitude modulation (PAM, QAM)
- Phase modulation (PSK, DPSK)
- Frequency modulation (FSK)
- Continuous phase modulation (CPM), including MSK and GMSK

Interleaving Libraries

A new Interleaving library contains sublibraries for block interleaving and convolutional interleaving. These sublibraries support general block interleavers and general multiplexed interleavers, as well as several special cases of these.

Fading Channels

The new Multipath Rayleigh Fading Channel and Rician Fading Channel blocks implement baseband simulations of fading propagation channels. These blocks model real-world mobile communication effects and are useful for modeling mobile wireless communication systems.

Enhanced Support for Convolutional Coding

The new APP Decoder block implements *a posteriori probability* decoding. The enhanced Convolutional Encoder and Viterbi Decoder blocks now support a more general class of convolutional codes by accepting a trellis parameter in their dialog boxes. The new `poly2trellis` function in the Communications Toolbox supports this enhancement, by converting a polynomial description of an encoder into a corresponding trellis description.

Sequence Operations

These new blocks in the Sequence Operations library manipulate data sequences in various ways:

- Bit to Integer Converter and Integer to Bit Converter convert between integers and their binary representations.
- Complex Phase Shift and Complex Phase Difference manipulate or analyze the phase of a complex signal.
- Derepeat is an inverse of the Signal Processing Blockset's Repeat block.
- Interlacer and Deinterlacer can be useful for combining or separating in-phase and quadrature components of a signal.
- Puncture and Insert Zero are useful for processing punctured codes.

Fixed Bugs

Some blocks do not support Real-Time Workshop code generation

You can now generate code with all blocks using Real-Time Workshop, except:

- Eye and scatter diagrams
- Continuous-time voltage-controlled oscillator
- Passband modulators and demodulators
- CPM modulators and demodulators
- Multipath Rayleigh Fading Channel
- Rician Fading Channel

Some blocks are not compatible with the Simulink Accelerator

All blocks, except the analog passband modulators and demodulators, are now compatible with the Simulink Accelerator.

The digital modulators produce continuous-time outputs and process datain continuous time

All of the digital modulators and demodulators have been rewritten and they now process data in discrete time and produce discrete-time outputs.

The M-FSK demodulators use incorrect correlation

The M-FSK demodulators now do a complex correlation of the input.

The complex Rayleigh fading channels have incorrect characteristics

The noise produced by the Rayleigh fading channels did not match with what is specified in the mask dialog. The two complex Rayleigh fading channels have now been replaced by the Multipath Rayleigh Fading Channel, which models real-world mobile communications effects.

Probabilities of ones in output from the Binary Vector Noise Generatorare incorrect

The probabilities of ones in the output of the Binary Vector Noise Generator now match what is specified in the mask.

Gaussian Noise Generator uses wrong sample time

The Gaussian Noise Generator now uses the correct sample time as what is specified in the mask.

PN Sequence Generator repeats its output

The PN Sequence Generator has been rewritten and now gives the correct results.

Eye and scatter plot blocks produce an error if you close the figure windowand run the simulation again

You can now run the simulation again after you close the figure window.

Scrambler and Descrambler give incorrect results for non-base-2 calculations

Scrambler and Descrambler now perform properly for any integer calculation base greater than 1.

Scrambler and Descrambler ignore all elements other than the first in acalculation base vector

Scrambler and Descrambler now accept only a scalar calculation base.

Scrambler and Descrambler negate the results when the initial states containnegative numbers

Scrambler and Descrambler now accept only nonnegative numbers as initial states.

The AWGN Channel produces all-zero outputs when given continuous-time inputsin both of the Signal to noise ratio (Es/No and SNR) modes

The AWGN Channel now does not accept continuous-time input for both of the Signal to noise ratio (Es/No and SNR) modes.

Segmentation violation when blocks are given invalid or empty parameters

All blocks now produce an error when given invalid or empty parameters.

Known Problems

Code Generation Limitations

Several blocks are incompatible with Real-Time Workshop. As a result, Real-Time Workshop cannot generate code for models that include these blocks:

- Discrete-Time Eye and Scatter Diagrams block
- Continuous-Time Eye and Scatter Diagrams block
- Voltage-Controlled Oscillator block
- Multipath Rayleigh Fading Channel block
- Rician Fading Channel block
- Blocks in the Analog Passband and Digital Passband sublibraries of the Modulation library
- Blocks in the CPM sublibrary of the Digital Baseband sublibrary of the Modulation library

Furthermore, blocks in the Analog Passband sublibrary of the Modulation library are not compatible with the Simulink Accelerator.

Limited Frame and Matrix Support

The Communications Blockset provides limited support for matrix and frame-based signals. In a future release, more blocks will support multichannel behavior, and more blocks will be optimized for faster frame-based processing. Release 12 strives to be forward-compatible in the sense that future signal support modes should not invalidate current modes and should minimize the difficulty of upgrading from Release 12 to a future release.

As a consequence of this forward-looking view, some blocks now use strict guidelines to determine the kinds of signals that they accept. One consideration is that if a block will ultimately support frame-based multichannel signals, then a sample-based vector input might potentially represent either a frame of data from a single channel or a set of samples from multiple channels. Therefore, even if such a block does not currently provide such comprehensive signal support, it accepts only frame-based vectors, whose interpretation is unambiguous.

New Block Libraries

The Communications Blockset uses a new set of block libraries, although it also includes the previous set of block libraries for backward compatibility.

Compatibility Considerations

The new set of libraries is what appears in the Simulink Browser (on PC) and what opens if you enter `commlib` at the MATLAB prompt. You should build new models using this new set.

Your previous models link to the previous set of libraries unless you choose to replace individual blocks manually. You can access the previous set of libraries by entering `commlib 1.5` at the MATLAB prompt.

Reorganization of Utility Functions in New Set of Libraries. The Utility Functions library has been reorganized. The table below lists blocks in Release 12 that were in the Release 11 Utility Functions library.

Block	New Location
Data Mapper	Utility Functions
Derepeat	Sequence Operations sublibrary
Descrambler	Sequence Operations sublibrary
Differential Decoder	Source Coding
Differential Encoder	Source Coding
Discrete Modulo Integrator (formerly called Discrete Time Modulo Integrator)	Integrators sublibrary
Discrete-Time VCO	Comm Sources
Windowed Integrator	Integrators sublibrary
Modulo Integrator	Integrators sublibrary
Integrate and Dump (formerly called Scheduled Reset Integrator)	Integrators sublibrary
Scrambler	Sequence Operations sublibrary
Voltage-Controlled Oscillator	Comm Sources

The Sequence Operations and Integrators sublibraries are in the Basic Comm Functions library.

New Signal Support

As of Release 12, Simulink supports matrix signals in addition to one-dimensional arrays, and frame-based signals in addition to sample-based signals.

Compatibility Considerations

The Communications Blockset processes certain kinds of matrix and frame-based signals.

Because a future release is planned to include more comprehensive matrix and frame support, some Release 12 blocks avoid conflict with future features by using strict guidelines to determine the kinds of signals that they now accept. As a consequence, if you used vector signals in a model before Release 12, then you might need to use a particular kind of vector signal in Release 12 (such as a frame-based column vector, a frame-based row vector, or a sample-based vector of a particular shape or dimension).

As another consequence of frame support, the AWGN Channel and Derepeat blocks no longer have the **Frame-based inputs** check box and the **Number of channels** parameter as in the Communications Toolbox 1.5. Instead, these blocks inherit the frame status and number of channels from their inputs.

Functionality Changes in Specific Blocks

Compatibility Considerations

- The Continuous-Time Eye and Scatter Diagrams and Discrete-Time Eye and Scatter Diagrams blocks process *complex* signals, whereas their counterparts before Release 12 (called Eye-Diagram Scatter Plot and Sample-Time Eye-Diagram Scatter) processed real vectors that listed in-phase and quadrature components separately.
- The blocks for Reed-Solomon and BCH coding no longer have a second input port for an enabler signal. The change affects the Binary-Input RS Encoder, Binary-Output RS Decoder, Integer-Input RS Encoder, Integer-Output RS Decoder, and BCH Decoder blocks.

- The Scrambler, Descrambler, and PN Sequence Generator blocks no longer have a trigger input. The Scrambler and Descrambler blocks no longer have a state output. The PN Sequence Generator block produces output from the last register in the generator, not the first.
- The Convolutional Encoder and Viterbi Decoder blocks have new interfaces because they can now accept a more general trellis description of a convolutional encoder.
- The Version 1.4 Error Rate Calculation block considers a vector input to be a sample, whereas the current block considers a vector input to be a frame of multiple samples. For vector inputs of length `n`, a **Receive delay** parameter value of `k` in the Version 1.4 block is equivalent to a **Receive delay** of `k*n` in the current block.
- The Voltage-Controlled Oscillator block now uses the cosine, not sine, function to produce its waveform. This change affects the phase of the output signal.
- The blocks in the Synchronization library no longer use a **Gain at the output** parameter. The remaining parameters that define characteristics of the voltage-controlled oscillator have changed slightly. Also, the Baseband PLL and Linearized Baseband PLL blocks now include three output ports instead of one, to match the Phase-Locked Loop and Charge Pump PLL blocks.

Block Name Changes

Compatibility Considerations

The table below lists the old and new names of blocks that were part of the Communications Toolbox before Release 12 and that have changed their names. The old names are from the last printed version of the Communications Toolbox User's Guide. Because the libraries have been reorganized since that document was printed, the third column of the table lists the current library name for each block.

Names of Blocks in Version 1.x and Version 2, Where Different

Old Block Name (Version 1.x)	New Block Name (Version 2)	Library Location
ADM with Carrier	DSB AM Demodulator Passband	Analog Passband
ADM with Carrier CE	DSB AM Demodulator Baseband	Analog Baseband
AM with Carrier	DSB AM Modulator Passband	Analog Passband
AM with Carrier CE	DSB AM Modulator Baseband	Analog Baseband
BCH Decode Vector In/Out	BCH Decoder	Block Codes
BCH Encode Vector In/Out	BCH Encoder	Block Codes
Baseband Model PLL	Baseband PLL	Synchronization
Bernoulli Random Binary Noise Generator	Bernoulli Binary Generator	Comm Sources
Binary Error Channel	Binary Symmetric Channel	Channels
Cyclic Decode Vector In/Out	Binary Cyclic Decoder	Block Codes
Cyclic Encode Vector In/Out	Binary Cyclic Encoder	Block Codes
DPCM Decode	DPCM Decoder	Source Coding
DPCM Encode	DPCM Encoder	Source Coding
DSB-SC ADM	DSBSC AM Demodulator Passband	Analog Passband
DSB ADM CE	DSBSC AM Demodulator Baseband	Analog Baseband

Names of Blocks in Version 1.x and Version 2, Where Different (Continued)

Old Block Name (Version 1.x)	New Block Name (Version 2)	Library Location
DSB-SC AM	DSBSC AM Modulator Passband	Analog Passband
DSB AM CE	DSBSC AM Modulator Baseband	Analog Baseband
Discrete Time VCO	Discrete-Time VCO	Comm Sources
Discrete Time Modulo Integrator	Discrete Modulo Integrator	Integrators
Eye-Pattern & Scatter Plot	Continuous-Time Eye and Scatter Diagrams	Comm Sinks
FDM	FM Demodulator Passband	Analog Passband
FDM CE	FM Demodulator Baseband	Analog Baseband
FM	FM Modulator Passband	Analog Passband
FM CE	FM Modulator Baseband	Analog Baseband
Gaussian Random Noise Generator	Gaussian Noise Generator	Comm Sources
Hamming Decode Vector In/Out	Hamming Decoder	Block Codes
Hamming Encode Vector In/Out	Hamming Encoder	Block Codes
Linear Block Decode Vector In/Out	Binary Linear Decoder	Block Codes
Linear Block Encode Vector In/Out	Binary Linear Encoder	Block Codes

Names of Blocks in Version 1.x and Version 2, Where Different (Continued)

Old Block Name (Version 1.x)	New Block Name (Version 2)	Library Location
Linearized Baseband Model PLL	Linearized Baseband PLL	Synchronization
μ-Law Compressor	Mu-Law Compressor	Source Coding
μ-Law Expander	Mu-Law Expander	Source Coding
PDM	PM Demodulator Passband	Analog Passband
PDM CE	PM Demodulator Baseband	Analog Baseband
PLL	Phase-Locked Loop	Synchronization
PM	PM Modulator Passband	Analog Passband
PM CE	PM Modulator Baseband	Analog Baseband
Poisson Random Integer Generator	Poisson Integer Generator	Comm Sources
Quantization Decode	Quantizer Decode	Source Coding
Reed-Solomon Decode Binary Vector In/Out	Binary-Output RS Decoder	Block Codes
Reed-Solomon Decode Integer Vector In/Out	Integer-Output RS Decoder	Block Codes
Reed-Solomon Encode Binary Vector In/Out	Binary-Input RS Encoder	Block Codes
Reed-Solomon Encode Integer Vector In/Out	Integer-Input RS Encoder	Block Codes
Rician Random Noise Generator	Rician Noise Generator	Comm Sources

Names of Blocks in Version 1.x and Version 2, Where Different (Continued)

Old Block Name (Version 1.x)	New Block Name (Version 2)	Library Location
SSB ADM	SSB AM Demodulator Passband	Analog Passband
SSB ADM CE	SSB AM Demodulator Baseband	Analog Baseband
SSB-AM	SSB AM Modulator Passband	Analog Passband
SSB-AM CE	SSB AM Modulator Baseband	Analog Baseband
Sample Time Eye-Pattern Diagram & Scatter Plot	Discrete-Time Eye and Scatter Diagrams	Comm Sinks
Scheduled Reset Integrator	Integrate and Dump	Integrators
Signal Quantizer	Sampled Quantizer Encode	Source Coding
Triggered Signal Quantizer	Enabled Quantizer Encode	Source Coding
Uniform Random Noise Generator	Uniform Noise Generator	Comm Sources

Names of Blocks in Version 1.x and Version 2, Where Different (Continued)

Old Block Name (Version 1.x)	New Block Name (Version 2)	Library Location
Uniform Random Integer Generator	Random Integer Generator	Comm Sources
VCO	Voltage-Controlled Oscillator	Comm Sources

Obsolete Blocks

Compatibility Considerations

The table below lists blocks that appear in the previous version of the Communications Toolbox User's Guide but that are not included in the Release 12 Communications Blockset. Where applicable, the second column lists blocks that provide similar functionality. In some cases, the similar block requires different parameter settings, data formats, or signal attributes compared to the original block. Therefore, you should read the documentation for the similar block before using it in your model.

Blocks Not in v2, and Similar v2 Blocks

Obsolete Block	Similar Block(s), if Any
Array Function	See Math library in Simulink.
BCH Code View Table	Use `bchpoly` in Communications Toolbox.
BCH Decode Sequence In/Out	BCH Decoder.
BCH Encode Sequence In/Out	BCH Encoder.
Coherent MFSK Corr Demod	
Coherent MFSK Demod	
Coherent MFSK Demod CE	

Blocks Not in v2, and Similar v2 Blocks (Continued)

Obsolete Block	Similar Block(s), if Any
Complex Filter	See Filtering library in DSP Blockset.
Convolutional Decode Sequence In/Out	Viterbi Decoder.
Convolutional Decode Vector In/Out	Viterbi Decoder
Convolutional Encode Sequence In/Out	Convolutional Encoder.
Convolutional Encode Vector In/Out	Convolutional Encoder
Cyclic Decode Sequence In/Out	Binary Cyclic Decoder.
Cyclic Encode Sequence In/Out	Binary Cyclic Encoder.
DPSK Demod	M-DPSK Demodulator Passband
DPSK Mod	M-DPSK Modulator Passband
D-TDMA Demux	
D-TDMA Mux	
Edge Detector	Edge Detector in DSP Blockset
Envelope Detector	Maximum, Minimum in DSP Blockset
Error Counter	Counter, in DSP Blockset
Error Rate Meter	Error Rate Calculation
Hamming Decode Sequence In/Out	Hamming Decoder.
Hamming Encode Sequence In/Out	Hamming Encoder.
Hilbert Filter	Remez FIR Filter Design in DSP Blockset
Integer Scalar to Vector	Integer to Bit Converter
Integer Vector to Scalar	Bit to Integer Converter
Interleave	Matrix Interleaver

Blocks Not in v2, and Similar v2 Blocks (Continued)

Obsolete Block	Similar Block(s), if Any
K-Step Delay	Integer Delay in DSP Blockset
Limited Binary Error Channel	Binary Vector Noise Generator
Linear Block Decode Sequence In/Out	Binary Linear Decoder.
Linear Block Encode Sequence In/Out	Binary Linear Encoder.
MASK Demap	
MASK Demod	M-PAM Demodulator Passband
MASK Demod CE	M-PAM Demodulator Baseband
MASK Map	
MASK Mod	M-PAM Modulator Passband
MASK Mod CE	M-PAM Modulator Baseband
Mean and Variance	Mean, Variance in DSP Blockset
Mean and Std	Mean, Standard Deviation in DSP Blockset
MFSK Map	
MFSK Mod	M-FSK Modulator Passband
MFSK Mod CE	M-FSK Modulator Baseband
Min/Max Demap	
Min/Max Index	Maximum, Minimum in DSP Blockset
Modulo	Math Function in Simulink
MPSK Correlation Demodulation	
MPSK Demod	M-PSK Demodulator Passband
MPSK Demod CE	M-PSK Demodulator Baseband
MPSK Map	

Blocks Not in v2, and Similar v2 Blocks (Continued)

Obsolete Block	Similar Block(s), if Any
MPSK Mod	M-PSK Modulator Passband
MPSK Mod CE	M-PSK Modulator Baseband
MSK Demod	MSK Demodulator Passband
MSK Mod	MSK Modulator Passband
Noncoherent MFSK Corr Demod	
Noncoherent MFSK Demod	M-FSK Demodulator Passband
Noncoherent MFSK Demod CE	M-FSK Demodulator Baseband
Number Counter	Counter, in DSP Blockset
OQPSK Demod	OQPSK Demodulator Passband
OQPSK Mod	OQPSK Modulator Passband
QADM	General QAM Demodulator Passband
QADM CE	General QAM Demodulator Baseband
QAM	General QAM Modulator Passband
QAM CE	General QAM Modulator Baseband
QASK Demap Arbitrary Constellation	
QASK Demap Circle Constellation	
QASK Demap Square Constellation	
QASK Demod Arbitrary Constellation	General QAM Demodulator Passband
QASK Demod CE Arbitrary Constellation	General QAM Demodulator Baseband
QASK Demod CE Circle Constellation	General QAM Demodulator Baseband

Blocks Not in v2, and Similar v2 Blocks (Continued)

Obsolete Block	Similar Block(s), if Any
QASK Demod CE Square Constellation	Rectangular QAM Demodulator Baseband
QASK Demod Circle Constellation	General QAM Demodulator Passband
QASK Demod Square Constellation	Rectangular QAM Demodulator Passband
QASK Map Arbitrary Constellation	
QASK Map Square Constellation	
QASK Mod Arbitrary Constellation	General QAM Modulator Passband
QASK Mod CE Arbitrary Constellation	General QAM Modulator Baseband
QASK Mod CE Circle Constellation	General QAM Modulator Baseband
QASK Mod CE Square Constellation	Rectangular QAM Modulator Baseband
QASK Mod Circle Constellation	General QAM Modulator Passband
QASK Mod Square Constellation	Rectangular QAM Modulator Passband
Raised Cosine Filter	
Rayleigh Fading CE Channel	Multipath Rayleigh Fading Channel
Rayleigh Noise CE Channel	Rayleigh Noise Generator
Reed-Solomon Decode Binary Sequence In/Out	Binary-Output RS Decoder.
Reed-Solomon Decode Integer Sequence In/Out	Integer-Output RS Decoder.
Reed-Solomon Encode Binary Sequence In/Out	Binary-Input RS Encoder.
Reed-Solomon Encode Integer Sequence In/Out	Integer-Input RS Encoder.

Blocks Not in v2, and Similar v2 Blocks (Continued)

Obsolete Block	Similar Block(s), if Any
Register Shift	Queue in DSP Blockset
Rician Noise CE Channel	Rician Noise Generator
Sampled Read From Workspace	Signal From Workspace in DSP Blockset
Sinc	
Time-Share Demux	
Time-Share Mux	
Triggered Read from Workspace	Triggered Signal From Workspace in DSP Blockset
Triggered Write to Workspace	Triggered To Workspace in DSP Blockset
Varying AWGN Channel	
Varying Rayleigh Fading CE Channel	
Varying Rayleigh Noise CE Channel	
Varying Rician Noise CE Channel	
Vector Pulse	Discrete Pulse Generator in Simulink
Vector Redistributor	

Compatibility Summary for Communications Blockset

This table summarizes new features and changes that might cause incompatibilities when you upgrade from an earlier version, or when you use files on multiple versions. Details are provided in the description of the new feature or change.

Version (Release)	New Features and Changes with Version Compatibility Impact
Latest Version V3.6 (R2007b)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "TCM Decoder Blocks Enhanced" on page 6
V3.5 (R2007a)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "M-FSK Modulator Block Errors When Used with Sample-Based Signals and Multitasking Solver" on page 10
V3.4 (R2006b)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "Binary Error Pattern Generator Block Is Obsoleted" on page 12 • "Version 1.5 Blocks Removed" on page 13 • "Obsolete Blocks" on page 13

Version (Release)	New Features and Changes with Version Compatibility Impact
V3.3 (R2006a)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "Rician Fading Channel Block Is Obsoleted" on page 19 • "SSB AM Passband Block Output" on page 20 • "saveas_commblks Obsoleted" on page 20
V3.2 (R14SP3)	None
V3.1 (R14SP2)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "commdigbbndam2 and commdigbbndpm2 Libraries Updated" on page 26 • "Improvements and Changes to the Multipath Rayleigh Fading Channel Block" on page 26 • "Discrete-Time VCO Block" on page 27 • "Voltage Controlled Oscillator Block Renamed to Continuous-Time VCO" on page 27 • "CPM Modulator Baseband and GMSK Modulator Baseband Blocks" on page 27

Version (Release)	New Features and Changes with Version Compatibility Impact
V3.0.1 (R14SP1)	None
V3.0 (R14)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "Changes in BCH Encoder and BCH Decoder" on page 37 • "Changes in Fading Channel Blocks" on page 37 • "Changes in Integrators" on page 37 • "Change in Error Rate Calculation Block" on page 39 • "Version 1.3 Libraries Removed" on page 40 • "Obsolete Blocks" on page 40 • "Blocks Now in Different Library Locations" on page 42 • "Changes in Block Dialog Boxes" on page 44 • "Changes in commstartup Function" on page 45 • "Simulation Settings of Legacy Models" on page 45

Version (Release)	New Features and Changes with Version Compatibility Impact
V2.5 (R13)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "Old Models Using the Baseband or Passband SSB Modulators Must Be Resaved" on page 54 • "Change the Boolean Logic Signals Parameter to Off" on page 54
V2.0.1 (R12.1)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "Binary Symmetric Channel Block" on page 57 • "Digital Passband Modulation Blocks" on page 57
V2.0 (R12)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "New Block Libraries" on page 63 • "New Signal Support" on page 65 • "Functionality Changes in Specific Blocks" on page 65 • "Block Name Changes" on page 66 • "Obsolete Blocks" on page 71