Signal Processing Blockset™ Release Notes

How to Contact The MathWorks

`www.mathworks.com`	Web
`comp.soft-sys.matlab`	Newsgroup
`www.mathworks.com/contact_TS.html`	Technical Support

`suggest@mathworks.com`	Product enhancement suggestions
`bugs@mathworks.com`	Bug reports
`doc@mathworks.com`	Documentation error reports
`service@mathworks.com`	Order status, license renewals, passcodes
`info@mathworks.com`	Sales, pricing, and general information

508-647-7000 (Phone)

508-647-7001 (Fax)

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

For contact information about worldwide offices, see the MathWorks Web site.

Signal Processing Blockset™ Release Notes

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See `www.mathworks.com/trademarks` for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

The MathWorks products are protected by one or more U.S. patents. Please see `www.mathworks.com/patents` for more information.

Contents

Summary by Version

This table provides quick access to what's new in each version. For clarification, see "Using 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 V6.10 (R2009b)	Yes Details	Yes Summary	Bug Reports Includes fixes	Printable Release Notes: PDF Current product documentation
V6.9 (R2009a)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
V6.8 (R2008b)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
V6.7 (R2008a)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
V6.6 (R2007b)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
V6.5 (R2007a)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
V6.4 (R2006b)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
V6.3 (R2006a)	Yes Details	No	Bug Reports Includes fixes	No
V6.2 (R14SP3)	Yes Details	No	Bug Reports Includes fixes	No
V6.1 (R14SP2)	Yes Details	Yes Summary	Bug Reports Includes fixes	No

Version (Release)	New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
V6.0.1 (R14SP1)	Yes Details	No	Fixed bugs	No
V6.0 (R14)	Yes Details	Yes Summary	Fixed bugs	No

Using Release Notes

Use release notes when upgrading to a newer version to learn about:

- New features
- Changes
- Potential impact on your existing files and practices

Review the release notes for other MathWorks™ products required for this product (for example, MATLAB® or Simulink®). Determine if enhancements, bugs, or compatibility considerations in other products impact you.

If you are upgrading from a software version other than the most recent one, review the current release notes and all interim versions. For example, when you upgrade from V1.0 to V1.2, review the release notes for V1.1 and V1.2.

What Is in the Release Notes

New Features and Changes

- New functionality
- Changes to existing functionality

Version Compatibility Considerations

When a new feature or change introduces a reported incompatibility between versions, the **Compatibility Considerations** subsection explains the impact.

Compatibility issues reported after the product release appear under Bug Reports at The MathWorks™ Web site. Bug fixes can sometimes result in incompatibilities, so review the fixed bugs in Bug Reports for any compatibility impact.

Fixed Bugs and Known Problems

The MathWorks offers a user-searchable Bug Reports database so you can view Bug Reports. The development team updates this database at release time and as more information becomes available. Bug Reports include provisions for any known workarounds or file replacements. Information is available for bugs existing in or fixed in Release 14SP2 or later. Information is not available for all bugs in earlier releases.

Access Bug Reports using your MathWorks Account.

Version 6.10 (R2009b) Signal Processing Blockset Software

This table summarizes what's new in Version 6.10 (R2009b):

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

New features and changes introduced in this version are:

"What's This?" Context-Sensitive Help Added for Signal Processing Blockset Blocks

R2009b introduces context-sensitive help for most Signal Processing Blockset blocks. To find out if a block supports context-sensitive help, look for the question mark icon () in the lower-left corner of the block dialog box. When you see this icon, you can get help on any parameter on the block mask by right-clicking on the parameter name.

Periodogram and Spectrum Scope Blocks Support Mean-Square Spectrum

Previously, the Periodogram and Spectrum Scope blocks supported only Power Spectral Density (PSD) visualizations. Now, these blocks also support Mean-Square Spectrum (MSS) computations. The Spectrum Scope block also supports MSS visualizations. As opposed to the PSD, which measures power per unit of frequency, the MSS measures power at a specific frequency.

- To compute the mean-square spectrum using the Periodogram block, set the **Measurement** parameter to `Mean-square spectrum`.
- To compute and visualize the mean-square spectrum using the Spectrum Scope block, set the **Spectrum units** parameter to one of the following:
  - `Watts`
  - `dBW`
  - `dBm`

Compatibility Considerations

The changes in the Spectrum Scope block result in the following two compatibility considerations.

Inherit sample increment from input parameter renamed Inherit sample time from input. The **Inherit sample increment from input** parameter of the Spectrum Scope block has been renamed **Inherit sample time from input**. The behavior of the parameter remains the same.

Tunability Status Changed for Two Parameters. The **Inherit sample time from input** (previously **Inherit sample increment from input**) and **Sample time of original time series** parameters of the Spectrum Scope block are no longer tunable during simulation. To change these parameters while you are working with a model, you must stop a running simulation, change the parameter, and then start the simulation again.

Power Spectrum Estimation Blocks Compute PSD Like the Signal Processing Toolbox Spectrum Objects

The behavior of the following blocks has changed so that they produce identical results to the Signal Processing Toolbox™ `spectrum` objects:

- Burg Method
- Covariance Method
- Modified Covariance Method
- Periodogram
- Yule-Walker Method

To produce correct results and align with the Signal Processing Toolbox `spectrum` objects, these blocks now scale their output by the sampling frequency, *Fs*. In addition to producing different results in R2009b, these five blocks have two new parameters:

- **Inherit sample time from input** — When you select this check box, the block inherits its sample time from the input signal
- **Sample time of original time series** — Specify the sample time of the original time series. The default value of this parameter is 1. This parameter becomes visible only when you clear the **Inherit sample time from input** check box.

Compatibility Considerations

In releases before R2009b, the block did not scale the output by *Fs*. To produce correct results and give the same output as the Signal Processing Toolbox spectrum objects, these five blocks now divide the spectrum by *Fs*.

To produce the old behavior, place a Gain block in front of these blocks in your model to multiply the signal by `sqrt(Fs)`.

Spectrum Scope Computes Both One-Sided and Two-Sided Spectra

The Spectrum Scope block now computes one-sided ([0... Fs]) and two-sided ((-Fs/2...Fs/2]) spectra in the same way that the Signal Processing Toolbox spectrum objects do. A new **Spectrum type** parameter on the Scope Properties tab replaces the **Frequency range** parameter that was on the Axis Properties tab in previous releases. You can set the **Spectrum type** to compute and display either a `One-sided ([0...Fs/2])` or `Two-sided ((-Fs/2...Fs/2])` spectrum.

Compatibility Considerations

The changes in the Spectrum Scope block result in the following compatibility considerations.

Spectrum Scope Output Now Scaled by Fs. To produce correct results and align with the Signal Processing Toolbox spectrum objects, the Spectrum Scope block now scales its output by *Fs*. Because of this change in scaling, the output of the Spectrum Scope block is different than in previous releases. To help you transition to this behavior, the block updates the Y-axis limits to provide the same view as previous releases. This update occurs the first time you run an R2009a or earlier model in R2009b. This update marks your model changed. Therefore, to preserve the updated Y-axis limits, save your model after running it in R2009b for the first time.

Spectrum type parameter replaces Frequency range parameter. In previous releases, the Spectrum Scope block computed two-sided spectra over the interval (0...Fs]. It displayed these spectrums over the interval specified by the **Frequency range** parameter. This release removes the **Frequency range** parameter from the Axis Properties tab and replaces it with a **Spectrum type** parameter on the Scope Properties tab. The Spectrum Scope block now computes and displays the spectrum over the interval specified by the **Spectrum type** parameter, either `One-sided ([0...Fs/2])` or `Two-sided ((-Fs/2...Fs/2])`. The blockset no longer supports the `[0...Fs]` option for the **Frequency range** parameter. The following table shows how values of the **Frequency range** parameter map to the new **Spectrum type** parameter.

Frequency Range	Spectrum Type
`[0...Fs/2]`	`One-sided ([0...Fs/2])`
`[-Fs/2...Fs/2]`	`Two-sided ((-Fs/2...Fs/2])`
`[0...Fs]`	`Two-sided ((-Fs/2...Fs/2])`

These changes mean that the one-sided spectrum for real signals in R2009b differs from the one-sided spectrum computed by the R2009a Spectrum Scope block. Also, the Spectrum Scope block no longer supports one-sided spectrums for complex inputs. To view the spectrum of a complex input in R2009b, set the **Spectrum type** parameter to `Two-sided ((-Fs/2...Fs/2])`.

Y-axis scaling Parameter Removed. This release removes the **Y-axis scaling** parameter from the Axis Properties tab. You now set the Y-axis units using the `Spectrum units` parameter on the Scope Properties tab of the block mask.

Frequency units Parameter Removed. This release also removes the **Frequency units** parameter from the Axis Properties tab. The units of the frequency axis are now always expressed in Hertz.

New Data Type Assistant and Ability to Specify Design Minimums and Maximums

Effective in R2009b, a new **Data type attributes** pane replaces the **Fixed-point** pane of the following blocks:

- Array-Vector Add
- Array-Vector Divide
- Array-Vector Multiply
- Array-Vector Subtract
- Autocorrelation
- Backward Substitution
- Convolution
- Correlation
- Cumulative Product
- Cumulative Sum
- Difference
- Forward Substitution
- Levinson-Durbin
- Matrix 1-Norm
- Matrix Product
- Maximum
- Mean
- Median
- Minimum
- Normalization
- Sort

The **Data type attributes** pane provides you with the following new features:

- A Data Type Assistant to help you specify data types on the block mask. See “Using the Data Type Assistant” in the Simulink documentation for more information.
- The ability to enable simulation range checking and automatic fixed-point scaling using minimum and maximum output values. You can specify

minimums and maximums for some data types based on design criteria. See “Checking Signal Ranges” in the Simulink documentation for more information.

Variable Size Data Support

Limited support for variable size data is now available. The following blocks support variable size data as of this release:

- Array-Vector Add
- Array-Vector Divide
- Array-Vector Multiply
- Array-Vector Subtract
- Difference
- FFT
- IFFT
- Inherit Complexity
- Matrix Product
- Matrix Sum
- Maximum
- Mean
- Minimum
- Normalization
- RMS
- Standard Deviation
- To Audio Device
- Variance
- Window Function

See “Working with Variable-Size Signals” for more information about variable size data.

Enumerated Data Type Support

Limited support for enumerated data types is now available. The following blocks support enumerated data types as of this release:

- Check Signal Attributes
- Constant
- Convert 2–D to 1–D
- Data Type Conversion
- Display
- Event-Count Comparator
- Flip
- Frame Conversion
- Matrix Concatenate
- Multiport Selector
- Permute Matrix
- Selector
- Signal To Workspace
- Submatrix
- Time Scope
- Triggered To Workspace
- Variable Selector

See “Using Enumerated Data” in the Simulink documentation for more information about enumerated data types.

New Pulse Shaping Filter Block

The Filter Design Toolbox™ library now includes the Pulse Shaping Filter block. See the block reference page for more information.

To Audio Device Block Handles Sample-Based Signals Differently

In previous releases, the To Audio Device block treated each sample of sample-based signals as a separate signal. In R2009b, the behavior of the block has changed such that it treats each column of both frame and sample-based inputs as separate channels.

Compatibility Considerations

This change in behavior results in a backward incompatibility for sample-based signals with a length greater than 1. For such input signals in R2009b, the To Audio Device Block produces different results than it did in past releases.

Zero-Latency Filter Structure Removed from CIC Blocks

The `Zero-latency decimator` and `Zero-latency interpolator` selections are no longer available. These options were previously available from the **Filter structure** parameter on the CIC Decimation and CIC Interpolation block masks. Because each block now only implements one filter structure, the **Filter structure** parameter has also been removed.

Compatibility Considerations

Before running a previously existing model that uses these modes, run the `slupdate` function. After you do, the block now produces *N* extra latencies, where *N* is the number of filter sections.

Empty Initial Conditions Support Removed from Some Blocks

The following blocks no longer support empty (`[ ]`) initial conditions:

- Buffer
- Delay Line
- FIR Decimation
- FIR Interpolation

- Integer Delay (Obsolete)
- Unbuffer

Compatibility Considerations

Provide the initial conditions parameter on these blocks with a nonempty value.

Version 6.9 (R2009a) Signal Processing Blockset Software

This table summarizes what's new in Version 6.9 (R2009a):

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

New features and changes introduced in this version are:

Zoom Capability for Spectrum Scope and Vector Scope Blocks

Zoom capability has been added to Spectrum Scope and Vector Scope blocks. The zoom feature allows you to

- Zoom in
- Zoom in on the *x*-axis only
- Zoom in on the *y*-axis only
- Zoom out
- Restore the original view
- Save your axes settings

Compatibility Considerations

The incompatibilities in the following sections were introduced with this change.

Autoscaling. When you autoscaled the *y*-axis of the Vector Scope or Spectrum Scope window in previous releases, the **Minimum Y-limit** and **Maximum Y-limit** block parameters updated to reflect the limits set by the autoscaling operation. This functionality is now removed. The *y*-axis in the scope window is still autoscaled, but the **Minimum Y-limit** and **Maximum Y-limit** block parameters do not update with the new *y*-axis limits. If you want to save your axes settings after autoscaling the *y*-axis, use the new **Save Axes Settings** menu item. This feature saves both the *x*- and *y*-axis limits of your current scope window.

Parameter Names. The parameter names listed in the following table have been changed.

Block	Mode	Old Parameter Name	New Parameter Name
Spectrum Scope	–	**Display DC as**	**Frequency display offset**
	–	**Amplitude scaling**	**Y-axis scaling**
	When **Frequency display units** is `User-defined`	**Minimum X-limit**	**Minimum frequency**
	When **Frequency display units** is `User-defined`	**Maximum X-limit**	**Maximum frequency**

Block	Mode	Old Parameter Name	New Parameter Name
Vector Scope	–	**Amplitude scaling**	**Y-axis scaling**
	When **Input domain** is `Frequency` and **Frequency display units** is `User-defined`	**Minimum X-limit**	**Minimum frequency**
	When **Input domain** is `Frequency` and **Frequency display units** is `User-defined`	**Maximum X-limit**	**Maximum frequency**

Units. In previous releases, the units used by some parameters of the Spectrum Scope and Vector Scope blocks were determined by the units used to scale the *x*-axis of the scope window. This behavior caused inconvenience because it required you to look at the scope window to determine the units on the *x*-axis before you could determine what units the block parameter was using.

In R2009a, this behavior has been changed such that you can now determine the units used by these parameters without looking at the *x*-axis of the scope window.

In the frequency domain, the following parameters now use the units specified by the **Frequency units** parameter (`Hertz` or `rad/sec`):

- Spectrum Scope block
  - **Frequency display offset** (formerly **Display DC as**)
  - **Minimum frequency** (formerly **Minimum X-limit**)
  - **Maximum frequency** (formerly **Maximum X-limit**)
- Vector Scope block
  - **Minimum frequency** (formerly **Minimum X-limit**)
  - **Maximum frequency** (formerly **Maximum X-limit**)

In the time domain, seconds are now used as units for the following parameters:

- Vector Scope block
  - **Minimum X-limit (s)**
  - **Maximum X-limit (s)**

Run-Time Library Removal

The Real-Time Workshop® software pack-and-go utility enables code portability. In previous releases, the Signal Processing Blockset™ software pack-and-go .zip file included more files than necessary because the code depended on a run-time library that shipped with the product. In addition, you could not examine the contents of Signal Processing Blockset run-time functions from the Real-Time Workshop HTML report, because the code was accessed through the previously built library. In R2009a, this dependence on the run-time library for code generation is removed. The pack-and-go .zip file is now much smaller than in previous releases, and the Real-Time Workshop HTML report provides links to the source code of the run-time functions.

Variable Fractional Delay Block Enhancements

Enhancements to the Variable Fractional Delay block include the following:

- A `Farrow` interpolation mode has been added.
- Fixed-point support has been added. You can set the fixed-point parameters on the **Fixed-point** pane.
- A new **Disable direct feedthrough by increasing minimum possible delay by one** check box allows you to use the block in feedback loops.
- A new **For small input delays** parameter is available when the block is in `FIR` or `Farrow` interpolation mode. This parameter allows you to specify the block's behavior for small input delay values.
- A new **Valid delay range** area on the block mask displays the possible range of valid delay values based on the settings of the block parameters. All input delay values less than D_{min} or greater than D_{max} are clipped to D_{min} and D_{max}, respectively.

Compatibility Considerations

The Fractional Delay Filter block from the **Filtering > Filter Design Toolbox** library has been removed from the product. Any of your existing models that use this block will continue to work. If you encounter any problems using the Fractional Delay Filter block or would like added functionality, replace the Fractional Delay Filter blocks in your models with Variable Fractional Delay blocks from the Signal Operations library.

Biquad Filter Block Allows Coefficients from Port

The Biquad Filter block has a new selection in the **Coefficients source** area that allows you to enter filter coefficients via `Input port(s)`. Bringing coefficients into the block via such ports allows you to tune the coefficients in your generated code.

Rounding Modes

The following rounding modes have been added to blocks that support fixed-point signals:

- `Convergent` — Rounds the result of a calculation to the closest representable number. In the case of a tie, `Convergent` rounds to the nearest even number. This rounding mode is the least biased method provided by the blockset.
- `Round` — Rounds the result of a calculation to the closest representable number. In the case of a tie, `Round` rounds positive numbers to the closest representable number in the direction of positive infinity, and it rounds negative numbers to the closest representable number in the direction of negative infinity.
- `Simplest` — Rounds the result of a calculation using the rounding mode (`Floor` or `Zero`) that adds the least amount of extra rounding code to your generated code. For more information, see "Rounding Mode: Simplest" in the Simulink® Fixed Point™ documentation.

Unsigned Data Type Support

Unsigned integer and fixed-point support has been added to the following blocks:

- Autocorrelation
- Convolution
- Correlation
- Matrix-1 Norm
- Normalization

New Parametric Audio Equalizer Demo

This release adds a Parametric Audio Equalizer demo to the Audio Processing library. The demo highlights a workflow for designing filters using a custom GUI and algorithmic code generation. Open this demo by typing `dspparameq`.

R12 Filtering Blocks Functionality Changed

The following blocks that appeared in the `dspobslib` library in R2008b have been changed or removed:

- Biquadratic Filter (Obsolete) — Removed. Any Biquadratic Filter blocks that you are using in your models will be automatically upgraded to use the Biquad Filter block.
- Direct Form II Transpose Filter (Obsolete) — Updated to use an implementation of the Digital Filter block.
- Time-Varying Direct-Form II Transpose Filter (Obsolete) — Updated to use an implementation of the Digital Filter block.
- Time-Varying Lattice Filter (Obsolete) — Updated to use an implementation of the Digital Filter block.

Compatibility Considerations

You might need to consider the following compatibility issues for your existing models that use these blocks:

- Empty ([ ]) initial conditions are no longer supported for these blocks.
- The Time-Varying Direct-Form II Transpose Filter block no longer supports non-normalized IIR filters.

- The Time-Varying Direct-Form II Transpose Filter block no longer supports coefficients of mixed complexity on the coefficients input port.

Version 6.8 (R2008b) Signal Processing Blockset Software

This table summarizes what's new in Version 6.8 (R2008b):

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

New features and changes introduced in this version are

Code Generation Support for Signals Up to 128 Bits

Signal Processing Blockset software now supports C code generation and the Simulink Accelerator and Rapid Accelerator modes for fixed-point and integer word lengths up to 128 bits. This enhancement enables you to generate code for anything you can simulate using Signal Processing Blockset blocks. The only exception is the Sort block, which only supports code generation up to 32 bits for fixed-point and integer signals.

New Biquad Filter Block

A new Biquad Filter block has been added to the Signal Processing Blockset Filtering / Filter Designs library. This block allows you to implement IIR filters with optimized numerics.

Increased N-Dimensional Support

The following blocks now support N-D signals:

- Downsample
- DCT
- FFT
- IDCT
- IFFT
- Window Function

New Data Type Support Table

You can now access the Signal Processing Blockset Data Type Support Table through the Simulink model Help menu. The table provides information about data type support and code generation coverage for all Signal Processing Blockset blocks. Select **Help > Block Support Table > Signal Processing Blockset** or **Help > Block Support Table > All Tables**. As always, you can also type `showsignalblockdatatypetable` at the MATLAB command line to bring up the table.

Rectangular ROI Support for Minimum and Maximum Blocks

The Minimum and Maximum blocks now support Rectangular ROI (region of interest). See the block reference pages for more information.

Autocorrelation Block Provides Additional Fixed-Point Support

The Autocorrelation block now supports the scaling of fixed-point signals. In previous releases, the block's **Scaling** parameter had to be set to `None` for fixed-point inputs. See the block reference page for more information.

QR Factorization Block Supports Full-Sized Outputs

The QR Factorization block now supports full-sized output matrices Q and R. In previous releases, the block only produced economy-sized outputs. This release adds the **Output size** parameter to enable you to select either `Economy` or `Full` outputs. Refer to the block reference page for more information.

Counter Block Enhancements

This release brings the following enhancements to the Counter block:

- The former Clk port label is now Inc when the block is counting up, and Dec when the block is counting down.
- The Inc/Dec port now supports 8-, 16-, and 32-bit signed and unsigned integer and fixed-point data types.
- A new Max port was added to enable you to control the **Counter size** via an input port.
- The **Hit values** parameter now accepts vectors as well as scalar values.
- A new check is in place to require that the **Hit values** are integers.
- A new check is in place to require **Maximum count** to be an integer.
- A new check is in place to require **Maximum count** to be representable by the **Count data type**.

Compatibility Considerations

Because of the new checking on the **Maximum count** and **Hit values** parameters, your models might produce new errors. Change these signals to data types supported by each parameter.

FFT Block Correctly Applies Divide-By-Two on Butterfly Outputs

The **Skip divide-by-two on butterfly outputs for fixed-point signals** check box on the **Fixed-point** pane of the FFT block dialog has been moved to the **Main** pane and renamed **Divide butterfly outputs by two**. This check box now correctly applies to both fixed-point and floating-point inputs. In releases prior to R2008b, the block always ignored this check box for floating-point inputs.

Compatibility Considerations

The FFT block can give results in R2008b and later releases that are scaled differently than in previous releases when all the following conditions are met:

- The block input is floating point.
- The **Divide butterfly outputs by two** check box is selected (or the **Skip divide-by-two on butterfly outputs for fixed-point signals** check box on the **Fixed-point** pane was NOT selected, for releases prior to R2008b) .
- The **Logging mode** parameter on the Fixed-Point Tool for the model is not set to `Minimums, maximums and overflows`.

Make sure that this check box is set correctly for your needs. Refer to the block reference page for more information.

Matrix Sum Block Is Now the Simulink Sum Block

The Matrix Sum block is now identical to the Simulink Sum block, with different defaults selected.

Compatibility Considerations

Your existing models that contain the old version of the Matrix Sum block will continue to work in this release, however, the old version of the block will be removed from the product in a future release. You should use the `slupdate` function to replace Matrix Sum blocks in your models with the new version. You might get slightly different results using the new version of the block when `Inherit via internal rule` is used.

Version 6.7 (R2008a) Signal Processing Blockset Software

This table summarizes what's new in Version 6.7 (R2008a):

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

New features and changes introduced in this version are

Increased N-Dimensional Support

The following blocks now support N-D signals:

- Autocorrelation
- Convolution
- Correlation
- Delay
- Histogram
- Mean
- Median
- RMS
- Standard Deviation
- Variable Fractional Delay
- Variable Integer Delay
- Variance

Parameter Data Types Upgrade

Edit boxes on all Signal Processing Blockset blocks now support all built-in MATLAB data types except `char` and `Boolean`.

From Audio Device and To Audio Device Blocks Generate Code on More Platforms

The From Audio Device and To Audio Device blocks can now generate code on any platform supported by MATLAB except Solaris™ platforms.

FIR Interpolation Block Improved for Frame-Based Signals

The FIR Interpolation block now uses less memory and is more efficient for frame-based signals when the **Framing** parameter is set to `Maintain input frame rate`.

Delay Line Block Allows Selective Enabling of Output

The Delay Line block now supports selective enabling of the block output for increased efficiency when you select the **Show En_Out port for selectively enabling output** parameter. This allows the block to be more efficient in applications for which the block output does not always need to be valid.

Increased Scaled Doubles Support

The following blocks now support the scaled doubles data type:

- Discrete Impulse
- Identity Matrix
- Sine Wave
- Window Function

Spectrum Scope Allows Relabeling of X-Axis Scale

The Axis Properties pane of the Spectrum Scope block now has the **Display DC as** parameter. This parameter allows you to specify a new label for the DC frequency (0 Hz) along the *x*-axis of your scope, which is desirable in modulated carrier applications.

Levinson-Durbin Block Coefficients Can Have Different Word Lengths

The polynomial coefficients (A) and reflection coefficients (K) of the Levinson-Durbin block are no longer constrained to have the same word length.

Reference Port Added to Frame Conversion Block

The Frame Conversion block now has a Ref port that allows you to inherit the output sampling mode from an input signal.

Tunability Status Changed for Stack and Queue Blocks

The parameters of the Stack and Queue blocks are no longer tunable.

Compatibility Considerations

All parameters of the Stack and Queue blocks that were previously tunable during simulation are no longer tunable. To change these parameters while you are working with a model, you must stop a running simulation, change the parameter, and then start the simulation again.

DSP Constant Block Removed in Favor of Constant Block

The DSP Constant block has been removed. Use the Simulink Constant block instead.

Compatibility Considerations

Use `slupdate` to replace DSP Constant blocks with Simulink Constant blocks in your models.

Matrix Concatenation Block Renamed to Matrix Concatenate

The Matrix Concatenation block has been renamed Matrix Concatenate. The functionality of this block has not changed.

New and Updated Demos

The following demos are new in this release. Enter the commands provided to open each demo from the MATLAB command line:

- Synthetic Aperture Radar — `SarImageFormation`
- Positional Audio — `dspAudioPos`
- Audio Special Effects — `dspaudioeffects`

The following demos are updated in this release:

- Internet Low Bitrate Codec (iLBC) for VoIP — `dspilbc`
- Pitch Shifting and Time Dilation Using a Phase Vocoder — `dsppitchtime`
- Radar Tracking — `aero_radmod_dsp`

- WWV Digital Receiver — `dspwwv`

Obsolete Blocks

The Kalman Adaptive Filter and Triggered Delay Line blocks are now obsolete.

Compatibility Considerations

Replace Kalman Adaptive Filter blocks in your models with the Kalman Filter block. Replace Triggered Delay Line blocks with a Delay Line block inside a Triggered Subsystem.

Version 6.6 (R2007b) Signal Processing Blockset Software

This table summarizes what's new in Version 6.6 (R2007b):

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

New features and changes introduced in this version are

New To Audio Device and From Audio Device Blocks

The From Audio Device and To Audio Device blocks have been added to the Signal Processing Sources and Signal Processing Sinks libraries, respectively. These blocks offer support for more than two audio channels and for Windows®, Macintosh®, and Linux® platforms. See the block reference pages for more information.

Compatibility Considerations

These blocks replace the To Wave Device and From Wave Device blocks, which are obsolete as of this release, and might be completely removed from the product in a future release. Replace To Wave Device and From Wave Device blocks in your models with the new To Audio Device and From Audio Device blocks.

New Array-Vector Math Blocks

The following new array-vector math blocks perform arithmetic operations along a specified dimension of an N-dimensional array:

- Array-Vector Add
- Array-Vector Divide
- Array-Vector Multiply
- Array-Vector Subtract

See the block reference pages for more information.

Note The Array-Vector Multiply block replaces the Matrix Scaling block, which is removed from the product as of this release. Matrix Scaling blocks in your existing models will be automatically replaced with Array-Vector Multiply blocks.

New CIC Filter Block

The CIC Filter block has been added to the Filter Design Toolbox library. See the block reference page for more information.

FFT and IFFT Blocks Are More Optimized for Fixed-Point Signals

The double-signal and half-length optimizations that the FFT and IFFT blocks used to apply only to floating-point signals now also apply to fixed-point signals. See “Algorithms Used for FFT Computation” and “Algorithms Used

for IFFT Computation" in the respective block reference pages for more information.

Rounding Modes Ceiling and Zero Added to Fixed-Point Blocks

The **Rounding Mode** parameter of each fixed-point-capable block has two new rounding modes:

- `Ceiling` rounds the result of a calculation to the closest representable number in the direction of positive infinity.
- `Zero` rounds the result of a calculation to the closest representable number in the direction of zero.

Increased N-Dimensional Support

The following blocks now have support for N-D signals:

- Array-Vector Add
- Array-Vector Divide
- Array-Vector Multiply
- Array-Vector Subtract
- Constant Ramp
- Difference
- Inherit Complexity
- Maximum
- Minimum

Increased Scaled Doubles Support

The following blocks now support the scaled doubles data type:

- Difference
- Normalization
- Matrix Product

- Matrix Sum

Increased Multichannel Support

The following blocks now support multichannel signals:

- LPC to LSF/LSP Conversion
- LPC to/from Cepstral Coefficients
- LPC to/from RC
- LPC/RC to Autocorrelation

DirectX Component Registration Limitations Removed from To Multimedia File and From Multimedia File Blocks

You are now able to use the From Multimedia File or To Multimedia File blocks without first having someone with system administrator privileges register the DirectX® components associated with these blocks on your Windows machine.

Tunability Status Changed for Some Block Parameters

The tunability status for the block parameters in the following table has been changed. This was done to maintain consistency of the tunability status for any given parameter across all simulation and code generation modes.

Block	Parameter	Old Tunability Status	New Tunability Status
Chirp	**Frequency sweep**	Simulation only	Never
	Initial frequency	Simulation only	Always
	Target frequency	Simulation only	Always
Digital Filter	**SOS matrix**	Simulation only	Always
	Scale values	Simulation only	Always

Block	Parameter	Old Tunability Status	New Tunability Status
Extract Triangular Matrix	**Extract**	Simulation only	Never
Histogram	**Normalized**	Simulation only	Never
Multiphase Clock	**Starting phase**	Always	Never
	Number of phase intervals over which clock is active	Simulation only	Never
	Active level	Always	Never
Normalization	**Norm**	Simulation only	Never
	Normalization bias	Simulation only	Always
Sine Wave	**Frequency**	In some modes	Always when **Sample mode** is `Continuous` or **Computation method** is `Trigonometric fcn`
	Phase offset	In some modes	Always when **Sample mode** is `Continuous` or **Computation method** is `Trigonometric fcn`
Sort	**Sort order**	Simulation only	Never

Compatibility Considerations

Due to these changes, some parameters that were previously tunable during simulation are no longer tunable. To change these parameters while you are

working with a model, you now have to stop a running simulation, change the parameter, and then start the simulation again.

Levinson-Durbin Block Now Treats Frame-Based Row Vectors Differently

The Levinson-Durbin block now treats a 1-by-*N* frame-based row vector on its input port as *N* channels with one sample each. Previously, the Levinson-Durbin block treated such an input as one channel with *N* samples. This change makes the Levinson-Durbin block consistent with the way most Signal Processing Blockset blocks treat frame-based row vectors.

Be aware that the block now errors for a 1-by-*N* frame-based row vector input when reflection coefficients (*K*) are output, since the block is required to have at least 2 samples per input channel to calculate *K*.

Compatibility Considerations

To get the old behavior in an existing model, you can introduce a Frame Conversion block before a Levinson-Durbin block in your model to convert the block input to a sample-based signal.

Version 6.5 (R2007a) Signal Processing Blockset Software

This table summarizes what's new in Version 6.5 (R2007a):

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

New features and changes introduced in this version are

R11.1 Blocks Have Been Removed in R2007a — Run Helper Script Before Upgrading

The R11.1 DSP Blockset blocks have been deprecated since R14SP2. These blocks have been completely removed from the Signal Processing Blockset product in R2007a.

Compatibility Considerations

We are providing a script and documentation to help you remove any R11.1 DSP Blockset blocks from your models and replace them with current Signal Processing Blockset blocks. You must run this script *before* upgrading to R2007a. Refer to our MATLAB Central submission titled "Tool for Removing R11 DSP Blockset Blocks from Models" on the Web to download the script and its associated documentation.

New Kalman Filter Block

The Kalman Filter block has been added to the Filtering > Adaptive Filters library. This block predicts or estimates the state of a dynamic system from a series of incomplete or noisy measurements. See the block reference page for more information.

Increased Unsigned Integer and Fixed-Point Support

Unsigned integer and fixed-point data type support has been added to the following blocks:

- Cumulative Product
- Cumulative Sum
- Difference
- FIR Decimation
- FIR Interpolation
- FIR Rate Conversion

Increased N-Dimensional Support

Support for N-D signals has been added to the following blocks:

- dB Conversion
- dB Gain
- Check Signal Attributes
- Frame Conversion
- Normalization

- Pad

X-Axis Control Added to Spectrum Scope and Vector Scope Blocks

More *x*-axis control has been added to the Spectrum Scope and Vector Scope blocks:

- You can now specify the range of the *x*-axis for the Spectrum Scope and Vector Scope blocks.
- You can now specify an *x*-offset for the Vector Scope block.

See the block reference pages for more information.

New Filter Design Toolbox Library Blocks

The following blocks have been added to the Filter Design Toolbox library:

- Arbitrary Magnitude Filter
- Octave Filter
- Parametric Equalizer
- Peak-Notch Filter

See the block reference pages for more information.

Fixed-Point Support and Tunability Added to Filter Design Toolbox Library Blocks

The blocks in the Filter Design Toolbox library now support fixed-point and integer data types on their input and output ports. In addition, parameters of these blocks that do not change filter order or structure are now tunable.

New FFT Length Parameters on FFT and IFFT Blocks

The **Inherit FFT length from input dimensions** and **FFT length** parameters have been added to the FFT and IFFT blocks. See the block reference pages for more information.

Zero Pad Block Removed

The Zero Pad block has been removed from the Signal Processing Blockset product.

Compatibility Considerations

You can use the Pad block with the **Pad value** parameter set to `0` to exactly replicate the functionality of the Zero Pad block. Any Zero Pad blocks in existing models will be automatically replaced by Pad blocks with the **Pad value** parameter set to `0`. Your models will continue to work correctly.

Pad Block Can Truncate Either End of an Input Signal

You can use the Pad block to truncate a signal by specifying an output length that is shorter than the input length in a given dimension. In previous releases, the block ignored the value of the **Pad signal at** parameter and always truncated the end of a signal.

Compatibility Considerations

The Pad block now obeys the **Pad signal at** parameter for truncation as well as for padding, enabling you to truncate a signal at its beginning, end, or both. To get the previous behavior, make sure that the **Pad signal at** parameter is set to `End` for any Pad blocks in your model that are truncating the input signal.

New and Updated Demos

The Vorbis Decoder demo has been added to the Audio Processing library. This demo implements the Vorbis decoder, which is a freeware, open-source alternative to the MP3 standard. This audio decoding standard supports the segmentation of encoded data into small packets for network transmission. Open this demo by typing `dspvorbisdec`.

The Internet Low Bit-Rate Codec (iLBC) demo in the Audio Processing library has been improved. This demo now supports single-precision floating-point data, and both builds and runs faster. Open this demo by typing `dspilbc`.

Version 6.4 (R2006b) Signal Processing Blockset Software

This table summarizes what's new in Version 6.4 (R2006b):

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

New features and changes introduced in this version are

R11.1 Blocks Will Be Removed in R2007a

The R11.1 Signal Processing Blockset blocks have been deprecated since R14SP2. In the next release, R2007a, these blocks will be completely removed from the product.

Compatibility Considerations

We strongly recommend that you replace any R11.1 blocks that you are using in your models at this time. For more information, refer to “Obsolete Blocks” on page 50.

New Filter Design Toolbox Block Library

A new Filter Design Toolbox block library has been added for the design and implementation of single- and multirate FIR and IIR filters. The library contains the following blocks:

- Bandpass Filter
- Bandstop Filter
- CIC Compensator
- Differentiator Filter
- Fractional Delay Filter
- Halfband Filter
- Highpass Filter
- Hilbert Filter
- Inverse Sinc Filter
- Lowpass Filter
- Nyquist Filter

More Blocks with Fixed-Point Support

Support for fixed-point data types has been added to the following blocks:

- Backward Substitution
- Forward Substitution
- LDL Factorization
- LU Factorization

From Multimedia File Block Supports Uncompressed AVI Files on UNIX

The From Multimedia File block now supports uncompressed AVI files on UNIX® platforms. As a result, you no longer need to use separate blocks to import multimedia files if you are working on both Windows and UNIX platforms.

To Wave File and From Wave File Blocks Extended to Support More than Two Channels

The To Wave File and From Wave File blocks now support an arbitrary number of audio channels, instead of just mono and stereo.

Enabled Subsystem Support for From Wave File Block

The From Wave File block now supports enabled subsystems.

Diagnostic Output Port Added to Report a Failure to Converge

A new diagnostic output port has been added to the following blocks to report a failure to converge:

- Pseudoinverse
- Singular Value Decomposition
- SVD Solver

To make this port appear, select the **Show error status port** check box on the block dialog.

Compatibility Considerations

In prior releases, these blocks returned an error when the computation failed to converge. This error no longer occurs. Instead, select the **Show error status port** check box on the block dialog to make the error port E appear. You can then connect this port to a block such as the Simulink Assertion block to receive information about the convergence of the output.

2-D Support Added

2-D support has been added to the following blocks:

- Matrix Product
- Matrix Sum
- Maximum
- Minimum

Multichannel Support Added

Multichannel support has been added to the following blocks:

- Autocorrelation LPC
- Levinson-Durbin
- LSF/LSP to LPC Conversion
- Yule-Walker AR Estimator
- Zero Crossing

Blocks Removed from Product

The DSP Gain, DSP Sum, DSP Product, and DSP Fixed-Point Attributes blocks have been removed from the Signal Processing Blockset product.

Compatibility Considerations

You can replace any DSP Gain, DSP Sum, and DSP Product blocks in your models with Simulink Gain, Sum, and Product blocks, respectively. There is no replacement for the DSP Fixed-Point Attributes block.

Version 6.3 (R2006a) Signal Processing Blockset Software

This table summarizes what's new in Version 6.3 (R2006a):

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

Integration of Filter Blocks with Signal Processing Toolbox Filter Objects and FVTool

Significant enhancements were made to the following filter blocks for this release:

- CIC Decimation
- CIC Interpolation
- FIR Decimation
- FIR Interpolation

- FIR Rate Conversion

The changes made to these blocks bring them into closer alignment with the Signal Processing Toolbox product:

- These filter blocks can now operate in two different modes, which you select in the **Coefficient source** group box. If you select **Dialog parameters**, you enter information about the filter in the block mask. If you select **Multirate filter object (MFILT)**, you can now specify the filter using a `mfilt` object from the Signal Processing Toolbox product.
- You can now open the Signal Processing Toolbox `fvtool` from the block masks to view the filter response.

A few minor changes have also been made to the Digital Filter block mask to bring it into closer alignment with these blocks and with the Signal Processing Toolbox product. However, most of the updates to this block for this improvement were made in the previous release. See "Digital Filter Block Enhancements" on page 47.

Transposed Direct Form Structure Added to FIR Decimation Block

You can now implement either a transposed direct form or a direct form structure with the FIR Decimation block using the **Filter structure** parameter.

The addition of the transposed direct form structure to this block brings it into closer alignment to the Signal Processing Toolbox `mfilt.firdecim` object.

Data Type Specification Modes Added to CIC Decimation and CIC Interpolation Blocks

The **Data type specification mode** parameter has been added to the CIC Decimation and CIC Interpolation blocks. This parameter allows you to choose how the word and fraction lengths are specified for the filter sections and outputs. You can choose to fully specify the word and fraction lengths of the filter sections and outputs yourself, or have one or more of these quantities automatically selected for you.

This feature brings these blocks into closer alignment with the Signal Processing Toolbox `mfilt.cicdecim` and `mfilt.cicinterp` objects.

Taylor Window Type Added to Window Function Block

The Taylor window type has been added to the Window Function block. The block functionality in this mode is identical to that of the Signal Processing Toolbox `taylorwin` function.

Reduced Simulation Memory Footprint for Fixed-Point Capable Blocks

Fixed-point capable Signal Processing Blockset blocks now use less memory as they simulate. There is no change to the memory requirements for the generated code from these blocks.

Improved Usability for the To Wave Device Block

The usability of the To Wave Device block has been improved with the addition of the **Automatically determine internal buffer size** and **User-defined internal buffer size** parameters. These parameters allow you to define the size of the chunks of data that are written to the hardware audio device by the block, independently of the input dimensions. The block reference page in the documentation also has significant updates, including a "Troubleshooting" section. Refer to the reference page for more information.

New Demos

Demo Name	Signal Processing Demo Library Location	Launch Command
DTMF Generator and Receiver	Communications	`dspdtmf`
Envelope Detection	Miscellaneous	`dspenvdet`
Internet Low Bitrate Codec (iLBC)	Audio Processing	`dspilbc`

Version 6.2 (R14SP3) Signal Processing Blockset Software

This table summarizes what's new in Version 6.2 (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

New Numerically Controlled Oscillator (NCO) Block

The NCO block in the Signal Operations library is new for this release.

Digital Filter Block Enhancements

Significant enhancements were made to the Digital Filter block for this release:

- Digital Filter can now operate in two different modes, which you select in the **Filter source** group box. If you select **Specify filter characteristics in dialog**, you enter information about the filter in the block mask as in previous releases. If you select **Specify discrete-time filter object (DFILT)**, you can now specify the filter using a `dfilt` object from the Signal Processing Toolbox product.
- You can now open the Signal Processing Toolbox `fvtool` from the Digital Filter block mask to view the filter response.

Fixed-Point Support Added to the Matrix Multiply Block

The Matrix Multiply block now has functionality identical to the Simulink Product block. The block now supports Boolean, integer, and fixed-point data types.

Simulink Virtual Bus Support Added to Key Blocks

Simulink® virtual bus support has been added to the following blocks:

- DCT
- Delay
- Flip
- Overwrite Values
- Submatrix
- Transpose

For more information on virtual buses, refer to "Using Buses" in the Using Simulink documentation.

New Audio Sample Rate Conversion Demo

The new Audio Sample Rate Conversion demo illustrates audio sample rate conversion of a 48 kHz (DAT sampling rate) input audio signal to a 44.1 kHz (CD sampling rate) output audio signal using a multistage multirate FIR rate conversion approach. You can access this demo from the **Demos** pane of the Help browser under **Blocksets > Signal Processing > Audio Processing**.

Version 6.1 (R14SP2) Signal Processing Blockset Software

This table summarizes what's new in Version 6.1 (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

Broader Support for the Logging of Simulation Minimums and Maximums and Fixed-Point Autoscaling

An increased number of fixed-point capable Signal Processing Blockset blocks now support the logging of simulation minimums and maximums and autoscaling via the Fixed-Point Settings interface.

Fixed-Point Support for the DCT and IDCT Blocks

The DCT and IDCT blocks now support fixed-point data types.

New Audio File Source and Sink Blocks

The From Multimedia File and To Multimedia File blocks in the Platform Specific I/O > Windows (WIN32) library are new in this release.

Multirate Support for CIC Filter Blocks

The CIC Decimation and CIC Interpolation blocks now support multirate sample-based processing.

Obsolete Blocks

The blocks in the table below are obsolete, although they are currently still shipped with the product, and may be removed in a future version of the Signal Processing Blockset product. We recommend that you use the replacement blocks listed in the third column.

Compatibility Considerations

You can run the Signal Processing Blockset function `dsp_links` to see if you are using any obsolete blocks in your models. If your models are using obsolete blocks, we strongly recommend that you exchange them for blocks that are currently supported.

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

Obsolete (R11.1) Block	Obsolete Block Library	Replacement Block(s)	Replacement Block(s) Library
Analog Filter Design	`dspddes2`	Analog Filter Design	`dsparch4`
Analytic Signal	`dspbdsp2`	Analytic Signal	`dspxfrm3`
Autocorrelation	`dspvect2`	Autocorrelation	`dspstat3`
Backward Substitution	`dsplinalg`	Backward Substitution	`dspsolvers`
Biquadratic Filter	`dsparch2`	Digital Filter	`dsparch4`
Buffer	`dspbuff2`	Buffer	`dspbuff3`

Obsolete (R11.1) Block	Obsolete Block Library	Replacement Block(s)	Replacement Block(s) Library
Buffered FFT Frame Scope	dspsnks2	Spectrum Scope	dspsnks4
Burg AR Estimator	dspparest2	Burg AR Estimator	dspparest3
Burg Method	dspspect2	Burg Method	dspspect3
Chirp	dspsrcs2	Chirp	dspsrcs4
Cholesky Factorization	dsplinalg	Cholesky Factorization	dspfactors
Cholesky Solver	dsplinalg	Cholesky Solver	dspsolvers
Commutator	dspswit2	Reshape > Frame Conversion > Unbuffer	Simulink block, dspsigattribs, dspbuff3
Complex Cepstrum	dspxfrm2	Complex Cepstrum	dspxfrm3
Complex Exponential	dspelem2	Complex Exponential	dspmathops
Constant Diagonal Matrix	dspmtrx2	Constant Diagonal Matrix	dspmtrx3
Contiguous Copy	dspelem2	Contiguous Copy	dspobslib
Convert Complex DSP to Simulink	dspelem2	No Direct Replacement	N/A
Convert Complex Simulink to DSP	dspelem2	No Direct Replacement	N/A
Convolution	dspvect2	Convolution	dspsigops
Correlation	dspvect2	Correlation	dspstat3
Covariance AR Estimator	dspparest2	Covariance AR Estimator	dapparest3
Covariance Method	dspspect2	Covariance Method	dspspect3
Create Diagonal Matrix	dspmtrx2	Create Diagonal Matrix	dspmtrx3
Cumulative Sum	dspvect2	Cumulative Sum	dspmathops

Obsolete (R11.1) Block	Obsolete Block Library	Replacement Block(s)	Replacement Block(s) Library
Counter	dspswit2	Counter	dspswit3
dB	dspelem2	dB Conversion	dspmathops
dB Gain	dspelem2	dB Gain	dspmathops
DCT	dspxfrm2	DCT	dspxfrm3
Detrend	dspbdsp2	Detrend	dspstat3
Difference	dspvect2	Difference	dspmathops
Digital FIR Filter Design	dspddes2	Digital Filter Design	dsparch4
Digital FIR Raised Cosine Filter Design	dspddes2	Digital Filter Design	dsparch4
Digital IIR Filter Design	dspddes2	Digital Filter Design	dsparch4
Direct-Form II Transpose Filter	dsparch2	Digital Filter	dsparch4
Discrete Constant	dspsrcs2	DSP Constant	dspsrcs4
Discrete Impulse	dspsrcs2	Discrete Impulse	dspsrcs4
Distributor	dspswit2	Buffer	dspbuff3
Downsample	dspbdsp2	Downsample	dspsigops
Dyadic Analysis Filter Bank	dspmlti2	Dyadic Analysis Filter Bank	dspmlti4
Dyadic Synthesis Filter Bank	dspmlti2	Dyadic Synthesis Filter Bank	dspmlti4
Edge Detector	dspswit2	Edge Detector	dspswit3
Event-Count Comparator	dspswit2	Event-Count Comparator	dspswit3
Extract Diagonal	dspmtrx2	Extract Diagonal	dspmtrx3
Extract Triangular Matrix	dspmtrx2	Extract Triangular Matrix	dspmtrx3

Obsolete (R11.1) Block	Obsolete Block Library	Replacement Block(s)	Replacement Block(s) Library
FFT	dspxfrm2	FFT	dspxfrm3
FFT Frame Scope	dspsnks2	Spectrum Scope	dspsnks4
Filter Realization Wizard	dsparch2	Filter Realization Wizard	daparch4
FIR Decimation	dspmlti2	FIR Decimation	dspmlti4
FIR Interpolation	dspmlti2	FIR Interpolation	dspmlti4
FIR Rate Conversion	dspmlti2	FIR Rate Conversion	dspmlti4
Flip	dspvect2	Flip	dspindex
Forward Substitution	dsplinalg	Forward Substitution	dspsolvers
Frequency Frame Scope	dspsnks2	Vector Scope	dspsnks4
From Wave Device	dspsrcs2	From Wave Device	dspwin32
From Wave File	dspsrcs2	From Wave File	dspwin32
Histogram	dspstat2	Histogram	dspstat3
IDCT	dspxfrm2	IDCT	dspxfrm3
IFFT	dspxfrm2	IFFT	dspxfrm3
Inherit Complexity	dspelem2	Inherit Complexity	dspsigattribs
Integer Delay	dspbdsp2	Delay	dspsigops
Kalman Adaptive Filter	dspadpt2	Kalman Adaptive Filter	dspadpt3
LDL Factorization	dsplinalg	LDL Factorization	dspfactors
LDL Solver	dsplinalg	LDL Solver	dspsolvers
Least Squares FIR Filter Design	dspddes2	Digital Filter Design	dsparch4
Levinson Solver	dsplinalg	Levinson-Durbin	dspsolvers
LMS Adaptive Filter	dspadpt2	LMS Filter	dspadpt3
LPC	dspbdsp2	Autocorrelation LPC	dsplp

Obsolete (R11.1) Block	Obsolete Block Library	Replacement Block(s)	Replacement Block(s) Library
LU Factorization	dsplinalg	LU Factorization	dspfactors
LU Solver	dsplinalg	LU Solver	dspsolvers
Magnitude FFT	dspspect2	Magnitude FFT	dspspect3
Matrix 1-Norm	dspmtrx2	Matrix 1-Norm	dspmtrx3
Matrix Constant	dspmtrx2	Constant	Simulink block
Matrix From Workspace	dspmtrx2	Signal From Workspace	dspsrcs4
Matrix Multiplication	dspmtrx2	Matrix Multiply	dspmtrx3
Matrix Product	dspmtrx2	Matrix Product	dspmtrx3
Matrix Scaling	dspmtrx2	Matrix Scaling	dspmtrx3
Matrix Square	dspmtrx2	Matrix Square	dspmtrx3
Matrix Sum	dspmtrx2	Matrix Sum	dspmtrx3
Matrix To Workspace	dspmtrx2	To Workspace	Simulink block
Matrix Viewer	dspsnks2	Matrix Viewer	dspsnks4
Maximum	dspstat2	Maximum	dspstat3
Mean	dspstat2	Mean	dspstat3
Median	dspstat2	Median	dspstat3
Minimum	dspstat2	Minimum	dspstat3
Modified Covariance AR Estimator	dsppparest2	Modified Covariance AR Estimator	dsppparest3
Modified Covariance Method	dspspect2	Modified Covariance Method	dspspect3
Multiphase Clock	dspswit2	Multiphase Clock	dspswit3
Normalization	dspvect2	Normalization	dspmathops
N-Sample Enable	dspswit2	N-Sample Enable	dspswit3
N-Sample Switch	dspswit2	N-Sample Switch	dspswit3

Obsolete (R11.1) Block	Obsolete Block Library	Replacement Block(s)	Replacement Block(s) Library
Overlap-Add FFT Filter	dsparch2	Overlap-Add FFT Filter	dsparch4
Overlap-Save FFT Filter	dsparch2	Overlap-Save FFT Filter	dsparch4
Partial Unbuffer	dspbuff2	Submatrix > Unbuffer	dspmtrx3, dspbuff3
Permute Matrix	dspmtrx2	Permute Matrix	dspmtrx3
Polynomial Evaluation	dspelem2	Polynomial Evaluation	dsppolyfun
Queue	dspbuff2	Queue	dspbuff3
QR Factorization	dsplinalg	QR Factorization	dspfactors
QR Solver	dsplinalg	QR Solver	dspsolvers
Random Source	dspsrcs2	Random Source	dspsrcs4
Repeat	dspbdsp2	Repeat	dspsigops
Real Cepstrum	dspxfrm2	Real Cepstrum	dspxfrm3
Rebuffer	dspbuff2	Buffer	dspbuff3
Reciprocal Condition	dsplinalg	Reciprocal Condition	dspmtrx3
Remez FIR Filter Design	dspddes2	Digital Filter Design	dsparch4
Reshape	dspmtrx2	Reshape	Simulink block
RLS Adaptive Filter	dspadpt2	RLS Filter	dspadpt3
RMS	dspstat2	RMS	dspstat3
Shift Register	dspbuff2	Delay Line	dspbuff3
Sample and Hold	dspswit2	Sample and Hold	dspsigops
Short-Time FFT	dspspect2	Periodogram	dspspect3
Signal From Workspace	dspsrcs2	Signal From Workspace	dspsrcs4
Signal To Workspace	dspsnks2	Signal To Workspace	dspsnks4

Obsolete (R11.1) Block	Obsolete Block Library	Replacement Block(s)	Replacement Block(s) Library
Sine Wave	dspsrcs2	Sine Wave	dspsrcs4
Sort	dspstat2	Sort	dspstat3
Stack	dspbuff2	Stack	dspbuff3
Standard Deviation	dspstat2	Standard Deviation	dspstat3
Submatrix	dspmtrx2	Submatrix	dspmtrx3
Time Frame Scope	dspsnks2	Vector Scope	dspsnks4
Time-Varying Direct-Form II Transpose Filter	dsparch2	Digital Filter	dsparch4
Time-Varying Lattice Filter	dsparch2	Digital Filter	dsparch4
Toeplitz	dspmtrx2	Toeplitz	dspmtrx3
To Wave Device	dspsnks2	To Wave Device	dspwin32
To Wave File	dspsnks2	To Wave File	dspwin32
Transpose	dspmtrx2	Transpose	dspmtrx3
Triggered Matrix To Workspace	dspsnks2	Triggered To Workspace	dspsnks4
Triggered Shift Register	dspbuff2	Triggered Delay Line	dspbuff3
Triggered Signal From Workspace	dspbdsp2	Triggered Signal From Workspace	dspsigops
Triggered Signal To Workspace	dspsnks2	Triggered To Workspace	dspsnks4
Unbuffer	dspbuff2	Unbuffer	dspbuff3
Uniform Decoder	dspquant	Uniform Decoder	dspquant2
Uniform Encoder	dspquant	Uniform Encoder	dspquant2
Unwrap	dspvect2	Unwrap	dspsigops
Upsample	dspbdsp2	Upsample	dspsigops

Obsolete (R11.1) Block	Obsolete Block Library	Replacement Block(s)	Replacement Block(s) Library
User-defined Frame Scope	dspsnks2	Vector Scope	dspsnks4
Variable Fractional Delay	dspbdsp2	Variable Fractional Delay	dspsigops
Variable Integer Delay	dspbdsp2	Variable Integer Delay	dspsigops
Variable Selector	dspelem2	Variable Selector	dspindex
Variance	dspstat2	Variance	dapstat3
Wavelet Analysis	dspmlti2	Wavelet Analysis	dspobslib
Wavelet Synthesis	dspmlti2	Wavelet Synthesis	dspobslib
Window Function	dspbdsp2	Window Function	dspsigops
Yule-Walker AR Estimator	dsppparest2	Yule-Walker AR Estimator	dsppparest3
Yule-Walker IIR Filter Design	dspddes2	Digital Filter Design	dsparch4
Yule-Walker Method	dspspect2	Yule-Walker Method	dspspect3

Version 6.0.1 (R14SP1) Signal Processing Blockset Software

This table summarizes what's new in Version 6.0.1 (R14SP1):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	No	Fixed bugs	No

New features and changes introduced in this version are

Changes from the Previous Release

In this release, the following blocks have been affected by changes in the behavior of source block dialog boxes and the Model Explorer. See the "Changed Source Dialog Box Behavior" section in the Simulink Release Notes.

- Chirp
- Constant Diagonal Matrix
- DSP Constant
- Multiphase Clock
- N-Sample Enable
- Random Source
- Sine Wave

New Demos

Demo Name	Signal Processing Demo Library Location	Launch Command
Cochlear implant speech processor	Audio Processing	`dspcochlear_all` (Platform independent) `dspcochlear_all_fixpt` (Platform independent, fixed-point version)
Creating sample-based signals	Working with Signals	`dspcreatesbsigs`
Creating frame-based signals	Working with Signals	`dspcreatefbsigs`
Creating multichannel signals	Working with Signals	`dspcreatemltichansigs`
Splitting and reordering multichannel signals	Working with Signals	`dspsplitreordmltichansigs`
Importing signals	Working with Signals	`dspimportsigs`
Exporting signals	Working with Signals	`dspexportsigs`

Enhanced Blocks

The following blocks have been enhanced for Release 14SP1:

- Sample and Hold
- Spectrum Scope

The Sample and Hold block has a new parameter, the **Latch (buffer) input** check box. If you select this check box, the block outputs the value of the input from the previous time step until the next triggering event occurs. This parameter enables this block to be used in a feedback loop.

The Spectrum Scope block has two new parameters, **Window type** and **Window sampling**. Use the **Window type** parameter to specify which window to apply to the input. Use the **Window sampling** parameter to specify whether the window samples are computed in a periodic or a symmetric manner.

Version 6.0 (R14) Signal Processing Blockset Software

This table summarizes what's new in Version 6.0 (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

Product Name Change

DSP Blockset has been renamed. The new name is Signal Processing Blockset.

Additional Fixed-Point Support

For this release, significant support for fixed-point development has been added to the Signal Processing Blockset product.

New Fixed-Point Blocks

The following new blocks support fixed-point data types:

- CIC Decimation
- CIC Interpolation
- Offset
- Peak Finder
- Scalar Quantizer Decoder
- Scalar Quantizer Encoder
- Vector Quantizer Decoder
- Vector Quantizer Encoder
- Zero Crossing

Blocks with Added Fixed-Point Support

The following blocks now support fixed-point data types:

- Constant Ramp
- Cumulative Product
- Cumulative Sum
- Difference
- Digital Filter — more structures now support fixed-point data types
- FIR Rate Conversion
- Histogram

- Levinson-Durbin
- LMS Filter
- Matrix 1-Norm
- Matrix Scaling
- Mean
- Median
- Normalization
- Short-Time FFT
- Signal From Workspace
- Signal To Workspace
- Sort
- Triggered Signal From Workspace
- Triggered To Workspace
- Toeplitz
- Two-Channel Analysis Subband Filter
- Two-Channel Synthesis Subband Filter

Fixed-Point Blocks with New Complex Support

The following blocks supported real fixed-point data types in the last major release. They now also support complex fixed-point data types:

- Autocorrelation
- Convolution
- Correlation
- FIR Decimation
- FIR Interpolation
- Sort

Fixed-Point Blocks with a New Interface

Many of the Signal Processing Blockset blocks that support fixed-point data types have a new, easier-to-use interface. For more information, see Setting Block Parameters in the Signal Processing Blockset User's Guide.

New Automatic Selection of Fixed-Point Word and Fraction Lengths

Many fixed-point capable Signal Processing Blockset blocks allow you to set intermediate data types via block mask parameters. The **Accumulator**, **Product output**, and **Output** parameters on many such blocks have a new `Inherit via internal rule` setting. When you select `Inherit via internal rule`, the accumulator, product output, or block output word and fraction lengths will be automatically calculated for you. In general, all the bits are preserved in the internal block algorithm for quantities using this selection. That is, the accumulator, product output, or block output word and fraction lengths are selected such that

- No overflow occurs
- No precision loss occurs

Internal rule equations specific to each block are given in the block reference pages.

New Logging of Simulation Minimums and Maximums and Autoscaling

A number of fixed-point capable Signal Processing Blockset blocks now support the logging of simulation minimums and maximums and autoscaling via the Fixed-Point Settings interface.

New Blocks

This section gives a brief description of each of the new blocks.

CIC Decimation and CIC Interpolation

The CIC Decimation and CIC Interpolation blocks are in the Filtering/ Multirate Filters library. These blocks decimate or interpolate a signal using a Cascaded Integrator-Comb filter.

G711 Codec

The G711 Codec block is in the Quantizers library. This block encodes a linear, pulse code modulation (PCM) narrowband speech signal using an A-law or mu-law encoder. The block decodes index values into quantized output values using an A-law or mu-law decoder. The block converts between A-law and mu-law index values.

Inverse Short-Time FFT

The Inverse Short-Time FFT block is in the Transforms library. This block recovers the time-domain signal by performing an inverse short-time, fast Fourier transform operation.

LPC to/from Cepstral Coefficients

The LPC to/from Cepstral Coefficients block is in the Linear Prediction library. This block converts linear prediction coefficients (LPCs) to cepstral coefficients (CCs) or cepstral coefficients to linear prediction coefficients.

Offset

The Offset block is in the Signal Operations library. This block truncates vectors by removing or keeping beginning or ending values.

Peak Finder

The Peak Finder block is in the Signal Operations library. This block finds the local maxima and/or minima of an input signal.

Scalar Quantizer Decoder

The Scalar Quantizer Decoder block is in the Quantizers library. This block converts each index value into a quantized output value.

Scalar Quantizer Encoder

The Scalar Quantizer Encoder block is in the Quantizers library. This block encodes each input value by associating it with the index value of a quantization region.

Short-Time FFT

The Short-Time FFT block is in the Transforms library. This block computes a nonparametric estimate of the spectrum using the short-time, fast Fourier transform method. The Short-Time FFT block that was located in the Power Spectrum Estimation library has been renamed the Periodogram block.

Vector Quantizer Decoder

The Vector Quantizer Decoder block is in the Quantizers library. This block finds the vector quantizer codeword that corresponds to a given, zero-based index value.

Vector Quantizer Design

The Vector Quantizer Design block is in the Quantizers library. This block designs a vector quantizer using the Vector Quantizer Design Tool (VQDTool).

Vector Quantizer Encoder

The Vector Quantizer Encoder block is in the Quantizers library. This block finds the index of the nearest codeword based on a Euclidean or weighted Euclidean distance measure.

Waterfall

The Waterfall block is in the DSP Sinks library. This block enables you to view vectors of data over time.

Zero Crossing

The Zero Crossing block is in the Signal Operations library. This block counts the number of times a signal crosses zero.

Enhanced Blocks

This section gives a brief description of each of the block enhancements.

Counter

The **Count data type** parameter of the Counter block now supports signed and unsigned integers.

Digital Filter

The Digital Filter block now supports these additional filter structures:

- FIR
  - Direct form symmetric
  - Direct form antisymmetric
- IIR Biquad (SOS)
  - Direct form I
  - Direct form I transposed
  - Direct form II

Every filter structure now supports fixed-point data types.

Biquad (SOS) filter structures support interstage floating-point and fixed-point scale values.

Matrix Viewer

The Matrix Viewer block parameters dialog box has been upgraded.

Scalar Quantizer Design

You can now use the Scalar Quantizer Design Tool to create Scalar Quantizer Encoder and Scalar Quantizer Decoder blocks inside your models.

Sort

The Sort block now supports an additional sorting algorithm. Now, for the **Sort algorithm** parameter, you can choose either `Quick sort` or `Insertion sort`. Previously, only the quick sort algorithm was supported.

Renamed Blocks

Periodogram

The Short-Time FFT block that was located in the Power Spectrum Estimation library has been renamed the Periodogram block. This block computes a

nonparametric estimate of the spectrum. All instances of the old Short-Time FFT block have been replaced by the Periodogram block.

New Demos

Demo Name	Signal Processing Demo Library Location	Launch Command
Adaptive filter convergence	Adaptive Processing	`lmsxyplot`
CELP speech coder	Audio Processing	`dspcelpcoder`
G711 A-law and A-Mu-A conversion	Audio Processing	`dspg711amua`
G711 Mu-law and Mu-A-Mu conversion	Audio Processing	`dspg711muamu`
G711 and PCM encoding	Audio Processing	`dspg711cmp`
Phase vocoder	Audio Processing	`dsppitchtime`
Plucked string	Audio Processing	`dsppluck`
Radar tracking demonstration	Aerospace	`aero_radmod_dsp`
Short-Time Spectral Attenuation	Spectral Analysis	`dspstsa`
Vector quantizer design	Miscellaneous	`dspvqtwodim`

The Short-Time FFT demo in Spectral Analysis demo library is now the Periodogram demo.

The Acoustic Noise Canceler demo (`dspanc`) is now supported on all platforms. It also has a fixed-point version (`dspanc_fixpt`).

The Signal Processing Blockset product has a new demo library called Fixed-Point. This library contains demo models that support fixed-point data types.

Triggered Subsystem Support

Signal Processing Blockset blocks now support triggered subsystems. The exceptions are

- Chirp
- Multiphase Clock
- Sine Wave
- Blocks with multiple sample times

Constant Sample Time Support

The Signal Processing Blockset product has extended support of constant sample times to its blocks. The output of blocks with constant sample times does not change during the simulation. You can remove all blocks having constant sample times from the simulation "loop" by setting the **Inline parameters** option. If you select the **Inline parameters** check box on the **Optimization** pane of the Configuration Parameters dialog box, the parameters of these blocks cannot be changed during a simulation, and simulation speed is improved.

Source Blocks Obey New Simulink Inherited Sample Time Parameter

Signal Processing Blockset source blocks capable of inheriting their sample time obey a new Simulink inherited sample time parameter. To view this parameter, open the Configuration Parameters dialog box. In the **Select** pane, expand **Diagnostics** and click **Sample Time**. The new parameter, **Source block specifies -1 sample time** appears in the left pane. This parameter can be set to `none`, `warning` (default), or `error`.

The Random Source block is the only block that does not obey this parameter. If its **Sample time** parameter is set to `-1`, the Random Source block inherits its sample time from its output port and never produces warnings or errors.

Signal & Scope Manager Support

You can use the Signal & Scope Manager to create and view signals without using blocks. The Signal Processing Blockset product provides signal generators and viewers that you can associate with your model using the Signal & Scope Manager. To view these generators and viewers, right-click in your model, and select **Signal & Scope Manager**. From the **Generators** and **Viewers** lists, expand **Signal Processing**.

For information on how to use the Signal & Scope Manager, see "Introducing the Signal and Scope Manager" in the Simulink documentation.

Multitasking Support

If you have a multirate model that you want to run in `MultiTasking` mode and your model contains any of the blocks listed below, your reset event can be delayed as much as one reset time interval so your model behaves deterministically:

- Minimum
- Maximum
- Mean
- Standard Deviation
- Variance
- RMS
- Cumulative Sum
- Cumulative Product
- Delay

To minimize delay in multirate models, run them in `SingleTasking` mode.

Multirate Models

The following blocks no longer support different sample rates at their input ports:

- Permute Matrix
- Variable Selector
- Variable Integer Delay

Scalar Quantizer Block Obsoleted

The Scalar Quantizer block has been replaced by the Scalar Quantizer Encoder and Scalar Quantizer Decoder blocks.

Obsolete Product Versions

As of Version 6.0 (Release 14) of the Signal Processing Blockset product, DSP Blockset Versions 2.2 (Release 10) and earlier are obsolete and no longer supported. DSP Blockset Version 3.x (Release 11) might also be obsoleted in a future release.

Compatibility Considerations

Models that contain blocks from Versions 2.2 and earlier will have broken links when loaded into Simulink 6.0 (Release 14). If you have models that contain blocks from DSP Blockset Versions 2.2 or earlier, replace the older blocks by blocks from DSP Blockset Versions 4.0 (Release 12) or later before upgrading to Signal Processing Blockset 6.0 software (Release 14). Use the command `dsp_links` to facilitate this process.

Compatibility Summary for Signal Processing Blockset Software

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 **V6.10 (R2009b)**	See the **Compatibility Considerations** subheading for each of these new features or changes: • "Periodogram and Spectrum Scope Blocks Support Mean-Square Spectrum" on page 5 • "Power Spectrum Estimation Blocks Compute PSD Like the Signal Processing Toolbox Spectrum Objects" on page 6 • "Spectrum Scope Computes Both One-Sided and Two-Sided Spectra" on page 7 • "To Audio Device Block Handles Sample-Based Signals Differently" on page 12 • "Zero-Latency Filter Structure Removed from CIC Blocks" on page 12 • "Empty Initial Conditions Support Removed from Some Blocks" on page 12

Version (Release)	New Features and Changes with Version Compatibility Impact
V6.9 (R2009a)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "Zoom Capability for Spectrum Scope and Vector Scope Blocks" on page 14 • "Variable Fractional Delay Block Enhancements" on page 17 • "R12 Filtering Blocks Functionality Changed" on page 19
V6.8 (R2008b)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "Counter Block Enhancements" on page 23 • "FFT Block Correctly Applies Divide-By-Two on Butterfly Outputs" on page 24 • "Matrix Sum Block Is Now the Simulink Sum Block" on page 24

Version (Release)	New Features and Changes with Version Compatibility Impact
V6.7 (R2008a)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "Tunability Status Changed for Stack and Queue Blocks" on page 27 • "DSP Constant Block Removed in Favor of Constant Block" on page 28 • "Obsolete Blocks" on page 29
V6.6 (R2007b)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "New To Audio Device and From Audio Device Blocks" on page 30 • "Tunability Status Changed for Some Block Parameters" on page 33 • "Levinson-Durbin Block Now Treats Frame-Based Row Vectors Differently" on page 35

Version (Release)	New Features and Changes with Version Compatibility Impact
V6.5 (R2007a)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "R11.1 Blocks Have Been Removed in R2007a — Run Helper Script Before Upgrading" on page 36 • "Zero Pad Block Removed" on page 39 • "Pad Block Can Truncate Either End of an Input Signal" on page 39
V6.4 (R2006b)	See the **Compatibility Considerations** subheading for each of these new features or changes: • "R11.1 Blocks Will Be Removed in R2007a" on page 40 • "Diagnostic Output Port Added to Report a Failure to Converge" on page 42 • "Blocks Removed from Product" on page 43
V6.3 (R2006a)	None
V6.2 (R14SP3)	None
V6.1 (R14SP2)	See the **Compatibility Considerations** subheading for this new feature or change: • "Obsolete Blocks" on page 50

Version (Release)	New Features and Changes with Version Compatibility Impact
V6.0.1 (R14SP1)	None
V6.0 (R14)	See the **Compatibility Considerations** subheading for this new feature or change: • "Obsolete Product Versions" on page 70