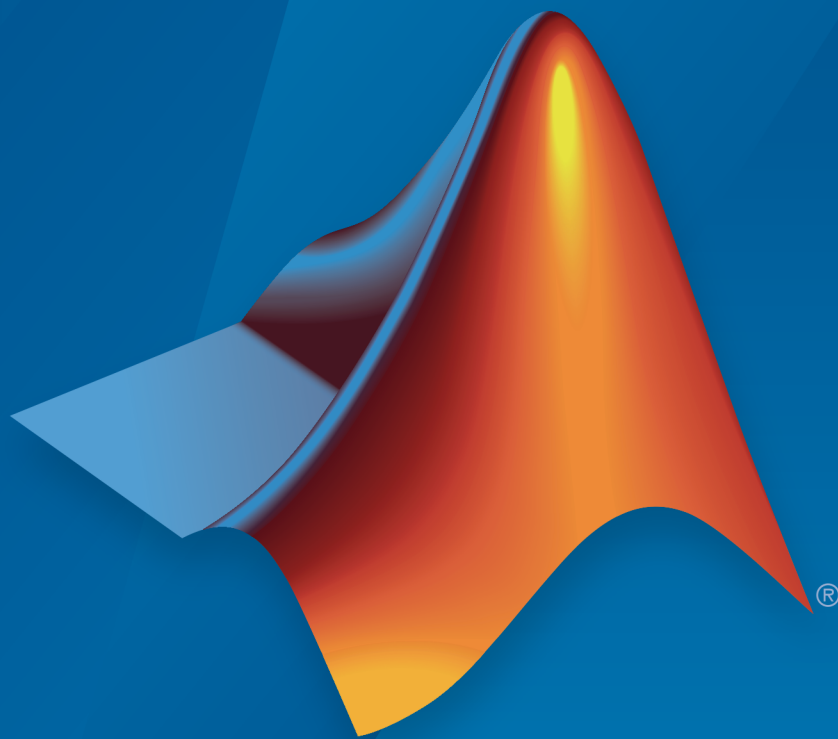


Audio System Toolbox™ Release Notes



MATLAB® & SIMULINK®

How to Contact MathWorks



Latest news: www.mathworks.com
Sales and services: www.mathworks.com/sales_and_services
User community: www.mathworks.com/matlabcentral
Technical support: www.mathworks.com/support/contact_us



Phone: 508-647-7000



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

Audio System Toolbox™ Release Notes

© COPYRIGHT 2016 - 2017 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

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

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

R2017a

Enhanced VST Workflow in Audio Test Bench: Interactively tune hosted VST plugins and test MATLAB objects in VST mode	1-2
Synchronized Playback and Acquisition: Play back and acquire audio signals synchronously in MATLAB via a single audioPlayerRecorder object	1-2
WASAPI Driver Support on Windows: Stream signals from and to audio devices equipped with WASAPI drivers	1-2
File browsing in Audio Test Bench	1-2
Additional fractional bandwidth option for octave filtering	1-2
configureMIDI support for hosted audio plugins	1-3
Tab completion for parameter names and options	1-3
Additional audio plugin examples	1-3

R2016b

Audio Plugin Hosting: Run and test VST plugins directly in MATLAB	2-2
--	-----

Improved Audio Test Bench: Choose from a wider range of input signals, and generate VST plugins directly from the app	2-2
Loudness Metering: Measure standard-compliant loudness parameters	2-2
Octave-Band Filters: Select octave and fractional-octave signal bands using standard-compliant digital filters	2-2
Audio Weighting Filters: Compensate signal magnitude for perceptual measurements using standard-compliant A-, C-, and K-weighted filters	2-3
Plugin class creation and MIDI support for multiband parametric equalizer	2-3
Simpler way to call System objects	2-3

R2016a

VST plugin generation for digital audio workstations	3-2
Interfaces to ASIO, ALSA, CoreAudio, and Windows Direct Sound	3-2
Interfaces to MIDI controls for real-time tuning of MATLAB and Simulink simulations	3-2
Audio processing algorithms, sources, and measurements for MATLAB and Simulink	3-2
Audio test bench to automatically generate an interactive audio simulation environment	3-2
Support for C code generation	3-3
Support for MATLAB Compiler	3-3

R2017a

Version: 1.2

New Features

Enhanced VST Workflow in Audio Test Bench: Interactively tune hosted VST plugins and test MATLAB objects in VST mode

The Audio Test Bench is a graphical debugging and testing suite for audio processing modules.

New abilities of the **Audio Test Bench** include:

- Host external VST and VST3 plugins. You can now interact with external plugins using the graphical UI of the **Audio Test Bench**, as you would in a DAW.
- Run audio plugins created in MATLAB® as VST plugins.

Synchronized Playback and Acquisition: Play back and acquire audio signals synchronously in MATLAB via a single audioPlayerRecorder object

The audioPlayerRecorder System object™ reads and writes from an audio device simultaneously, enabling real-time system measurements when using ASIO, Core Audio, or ALSA drivers. Combining the play and record into a single object enables easy configuration and consistent latency between input and output.

WASAPI Driver Support on Windows: Stream signals from and to audio devices equipped with WASAPI drivers

The audioDeviceWriter and audioDeviceReader System objects and the Audio Device Reader and Audio Device Writer blocks now support WASAPI drivers on Windows machines.

File browsing in Audio Test Bench

You can now browse for audio input files and the object under test directly from the Audio Test Bench.

Additional fractional bandwidth option for octave filtering

The octaveFilter System object and Octave Filter block now support $2/3$ octave bandwidth. This octave bandwidth is popular in graphic equalizer designs.

configureMIDI support for hosted audio plugins

You can now use the `configureMIDI` function to quickly synchronize your hosted audio plugins with MIDI devices.

Tab completion for parameter names and options

You can now use tab completion to complete parameter names and options for all System objects in Audio System Toolbox™. Tab completions also work for the `validateAudioPlugin`, `loadAudioPlugin`, and `generateAudioPlugin` functions.

Additional audio plugin examples

The Audio Plugin Gallery contains audio plugin example code that can be used as building blocks in larger systems, as models for design patterns, or as benchmarks for comparison.

New plugins in the gallery include:

- `audiopluginexample.FastConvolver`
- `audiopluginexample.Phaser`
- `audiopluginexample.MultiNotchFilter`
- `audiopluginexample.SpeechPitchDetector`
- `audiopluginexample.BeatDetector`

Enhancements to existing plugins in the gallery include:

- The `audiopluginexample.SpectralSubtractor` example now includes an analysis and synthesis buffering object. The `audiopluginexample.private.AnalysisAndSynthesisBuffer` object enables easy input/output buffering for the audio plugin API so that you can concentrate on your algorithm.

R2016b

Version: 1.1

New Features

Audio Plugin Hosting: Run and test VST plugins directly in MATLAB

The `loadAudioPlugin` function enables you to host external VST and VST3 plugins in MATLAB. You can process audio using the algorithm of the hosted plugin. You can interact with the hosted plugin programmatically by getting and setting parameters.

Improved Audio Test Bench: Choose from a wider range of input signals, and generate VST plugins directly from the app

The Audio Test Bench is a graphical debugging and testing suite for audio processing modules.

New abilities of the **Audio Test Bench** include:

- Switch the object under test in a single instance of the test bench.
- New input choices: `wavetableSynthesizer`, `audioOscillator`, `dsp.Chirp`, and `dsp.ColoredNoise`.
- Validate and generate VST plugins directly from the test bench.
- Track overrun and underrun in frames, seconds, or samples.

Loudness Metering: Measure standard-compliant loudness parameters

Measure integrated loudness and loudness range of an audio signal using the `integratedLoudness` function.

Measure momentary loudness, short-term loudness, integrated loudness, loudness range, and true-peak of streaming audio using the `loudnessMeter System` object. You can also open an 'EBU-Mode' visualization for loudness metering.

Measure momentary loudness, short-term loudness, and true-peak in the Simulink® environment using the Loudness Meter block.

Octave-Band Filters: Select octave and fractional-octave signal bands using standard-compliant digital filters

Perform octave-band and fractional octave-band filtering for arbitrary center frequency using the `octaveFilter System` object. With this object, you can tune center frequency and bandwidth while the simulation is running. To check your compliance to the ANSI

S1.11-2004 standard, use the `isStandardCompliant` method. To visualize and validate your filter response, use the `visualize` method.

In the Simulink environment, use the Octave Filter block.

Audio Weighting Filters: Compensate signal magnitude for perceptual measurements using standard-compliant A-, C-, and K-weighted filters

Perform frequency-weighted filtering using the `weightingFilter` System object. With this object, you can design A-weighted and C-weighted filters based on the ANSI S1.42-2001 standard, or K-weighted filters based on the ITU-R BS.1770-4 standard. To check your compliance to the IEC 61672-1:2002 standard, use the `isStandardCompliant` method. To visualize and validate your filter response, use the `visualize` method.

In the Simulink environment, use the Weighting Filter block.

Plugin class creation and MIDI support for multiband parametric equalizer

New functionality for the `multibandParametricEQ` System object includes:

- Plugin class creation using `createAudioPluginClass`
- MIDI support using `configureMIDI`

`multibandParametricEQ` is now enabled for the **Audio Test Bench**.

Simpler way to call System objects

Instead of using the `step` method to perform the operation defined by a System object, you can call the object with arguments, as if it were a function. The `step` method will continue to work. This feature improves the readability of scripts and functions that use many different System objects.

For example, if you create a `weightingFilter` System object named `Cweight`, then you call the System object as a function with that name.

```
Cweight = weightingFilter('C-weighting');  
Cweight(x)
```

The equivalent operation using the `step` method is:

```
Cweight = weightingFilter('C-weighting');  
step(Cweight,x)
```

When the `step` method has the System object as its only argument, the function equivalent has no arguments. This function must be called with empty parentheses. For example, `step(sysobj)` and `sysobj()` perform equivalent operations.

R2016a

Version: 1.0

New Features

VST plugin generation for digital audio workstations

Audio System Toolbox enables the design and generation of VST plugins.

For more information, see [Export a MATLAB Plugin to a DAW](#).

Interfaces to ASIO, ALSA, CoreAudio, and Windows Direct Sound

Audio System Toolbox enables real-time audio processing using low-latency audio drivers.

For more information, see [Audio I/O: Buffering, Latency, and Throughput](#).

Interfaces to MIDI controls for real-time tuning of MATLAB and Simulink simulations

Audio System Toolbox enables real-time tuning in MATLAB and Simulink using MIDI controls.

For more information, see [configureMIDI](#) and [Musical Instrument Digital Interface \(MIDI\)](#).

Audio processing algorithms, sources, and measurements for MATLAB and Simulink

Audio System Toolbox provides algorithms and tools for the design, simulation, and desktop prototyping of audio processing systems.

For more information, see [Audio Processing Algorithm Design](#).

Audio test bench to automatically generate an interactive audio simulation environment

Audio System Toolbox provides an all-in-one graphical debugging and testing suite.

For more information, see [Audio Test Bench](#) and [Use the Audio Test Bench](#).

Support for C code generation

You can use MATLAB Coder™ to generate efficient C and C++ code for most Audio System Toolbox functions, classes, and System objects.

For a list of supported functions and objects, see Audio System Toolbox.

For a guide to developing code capable of C code generation, see MATLAB Programming for Code Generation.

Support for MATLAB Compiler

You can use MATLAB Compiler™ to share MATLAB programs as standalone applications.

For an example, see Deploy Audio Applications with MATLAB Compiler.

