

# Simulink<sup>®</sup> Fixed Point<sup>™</sup>

## Reference

**R2012b**

MATLAB<sup>®</sup>  
& SIMULINK<sup>®</sup>

## How to Contact MathWorks



[www.mathworks.com](http://www.mathworks.com) Web  
[comp.soft-sys.matlab](mailto:comp.soft-sys.matlab) Newsgroup  
[www.mathworks.com/contact\\_TS.html](http://www.mathworks.com/contact_TS.html) Technical Support



[suggest@mathworks.com](mailto:suggest@mathworks.com) Product enhancement suggestions  
[bugs@mathworks.com](mailto:bugs@mathworks.com) Bug reports  
[doc@mathworks.com](mailto:doc@mathworks.com) Documentation error reports  
[service@mathworks.com](mailto:service@mathworks.com) Order status, license renewals, passcodes  
[info@mathworks.com](mailto: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.

*Simulink® Fixed Point™ Reference*

© COPYRIGHT 2009–2012 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](http://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](http://www.mathworks.com/patents) for more information.

**Revision History**

March 2009 Online only  
September 2009 Online only  
March 2010 Online only  
September 2010 Online only  
April 2011 Online only  
September 2011 Online only  
March 2012 Online only  
September 2012 Online only

New for Version 6.1 (Release 2009a)  
Revised for Version 6.2 (Release 2009b)  
Revised for Version 6.3 (Release 2010a)  
Revised for Version 6.4 (Release 2010b)  
Revised for Version 6.5 (Release 2011a)  
Revised for Version 7.0 (Release 2011b)  
Revised for Version 7.1 (Release 2012a)  
Revised for Version 7.2 (Release 2012b)



## Function Reference

**1**

## Selected Bibliography

**A**

## Glossary

## Index



# Function Reference

---

# autofixexp

---

**Purpose** Automatically change scaling of fixed-point data types

**Syntax** autofixexp

**Description** The autofixexp script automatically changes the scaling for model objects that specify fixed-point data types. However, if an object's **Lock output data type setting against changes by the fixed-point tools** parameter is selected, the script refrains from scaling that object.

This script collects range data for model objects, either from design minimum and maximum values that objects specify explicitly, or from logged minimum and maximum values that occur during simulation. Based on these values, the tool changes the scaling of fixed-point data types in a model so as to maximize precision and cover the range.

You can specify design minimum and maximum values for model objects using parameters typically titled **Output minimum** and **Output maximum**. See “Blocks That Allow Signal Range Specification” in *Simulink® User's Guide* for a list of Simulink blocks that permit you to specify these values. In the autoscaling procedure that the autofixexp script executes, design minimum and maximum values take precedence over the simulation range.

If you intend to scale fixed-point data types using simulation minimum and maximum values, the script yields meaningful results when exercising the full range of values over which your design is meant to run. Therefore, the simulation you run prior to using autofixexp must simulate your design over its full intended operating range. It is especially important that you use simulation inputs with appropriate speed and amplitude profiles for dynamic systems. The response of a linear dynamic system is frequency dependent. For example, a bandpass filter will show almost no response to very slow and very fast sinusoid inputs, whereas the signal of a sinusoid input with a frequency in the passband will be passed or even significantly amplified. The response of nonlinear dynamic systems can have complicated dependence on both the signal speed and amplitude.



---

**Note** If you already know the simulation range you need to cover, you can use an alternate autoscaling technique described in the `fixptbestprec` reference page.

---

To control the parameters associated with automatic scaling, such as safety margins, use the Fixed-Point Tool.

For more information, see “Overview of the Fixed-Point Tool”.

To learn how to use the Fixed-Point Tool, refer to “Propose Fraction Lengths Using Simulation Range Data”.

**See Also**

`fxptdlg`

# fixpt\_instrument\_purge

---

**Purpose** Remove corrupt fixed-point instrumentation from model

---

**Note** `fixpt_instrument_purge` will be removed in a future release.

---

**Syntax**

```
fixpt_instrument_purge  
fixpt_instrument_purge(modelName, interactive)
```

**Description** The `fixpt_instrument_purge` script finds and removes fixed-point instrumentation from a model left by the Fixed-Point Tool and the fixed-point autoscaling script. The Fixed-Point Tool and the fixed-point autoscaling script each add callbacks to a model. For example, the Fixed-Point Tool appends commands to model-level callbacks. These callbacks make the Fixed-Point Tool respond to simulation events. Similarly, the autoscaling script adds instrumentation to some parameter values that gathers information required by the script.

Normally, these types of instrumentation are automatically removed from a model. The Fixed-Point Tool removes its instrumentation when the model is closed. The autoscaling script removes its instrumentation shortly after it is added. However, there are cases where abnormal termination of a model leaves fixed-point instrumentation behind. The purpose of `fixpt_instrument_purge` is to find and remove fixed-point instrumentation left over from abnormal termination.

`fixpt_instrument_purge(modelName, interactive)` removes instrumentation from model `modelName`. `interactive` is `true` by default, which prompts you to make each change. When `interactive` is set to `false`, all found instrumentation is automatically removed from the model.

**See Also** `autofixexp` | `fxptdlg`

**Purpose** Determine fixed-point data type

**Syntax**

```
nts
nts({'block',PORT})
nts({line-handle})
nts({gs1})
```

**Description** nts opens the NumericTypeScope window. To connect to a signal in a Simulink model, select the signal and then, in the NumericTypeScope window, select **File > Connect to Simulink Signal**.

The NumericTypeScope suggests a fixed-point data type in the form of a numerictype object based on the dynamic range of the input data and the criteria that you specify in the Bit Allocation Panel on page 13. The scope allows you to visualize the dynamic range of data in the form of a log<sub>2</sub> histogram. It displays the absolute values of data on the X-axis and the number or percentage of occurrences on the Y-axis. Each bin in the histogram corresponds to a bit in a word. For example,  $2^0$  corresponds to the first integer bit in the binary word,  $2^{-1}$  corresponds to the first fractional bit in the binary word, and the binary point lies between them. The height of the bar on the zeroth bit indicates the number or percentage of cases where the input value is in the range  $0.5 < \text{value} \leq 1$ .

nts({'block',PORT}) opens the NumericTypeScope window and connects the scope to the signal output from block on output port with index PORT. If the block has more than one output port, you must specify the port index. The scope cannot connect to more than one output port.

nts({line-handle}) opens the NumericTypeScope window and connects the scope to the Simulink signal which has the line handle specified in line-handle.

nts({gs1}) opens the NumericTypeScope window and connects the scope to the currently selected Simulink signal. You must select a signal in a Simulink model first, otherwise the scope opens with no signal selected.

## Tips

- Use the `NumericTypeScope` to help you identify any values that are outside range or below precision based on the current data type.

When the information is available, the scope indicates values that are outside range, below precision, and in range of the data type by color-coding the histogram bars as follows:

- Blue — Histogram bin contains values that are in range of the current data type.
  - Red — Histogram bin contains values that are outside range in the current data type.
  - Yellow — Histogram bin contains values that are below precision in the current data type.
- Select **View > Vertical Units** to select whether to display values as a percentage or as an actual count.
  - Use the **View > Bring All NumericTypeScope Windows Forward** menu option to manage your `NumericTypeScope` windows. Selecting this option or pressing **Ctrl+F** brings all `NumericTypeScope` windows into view.

## Input Arguments

### **block**

Full path to the specified block.

### **line-handle**

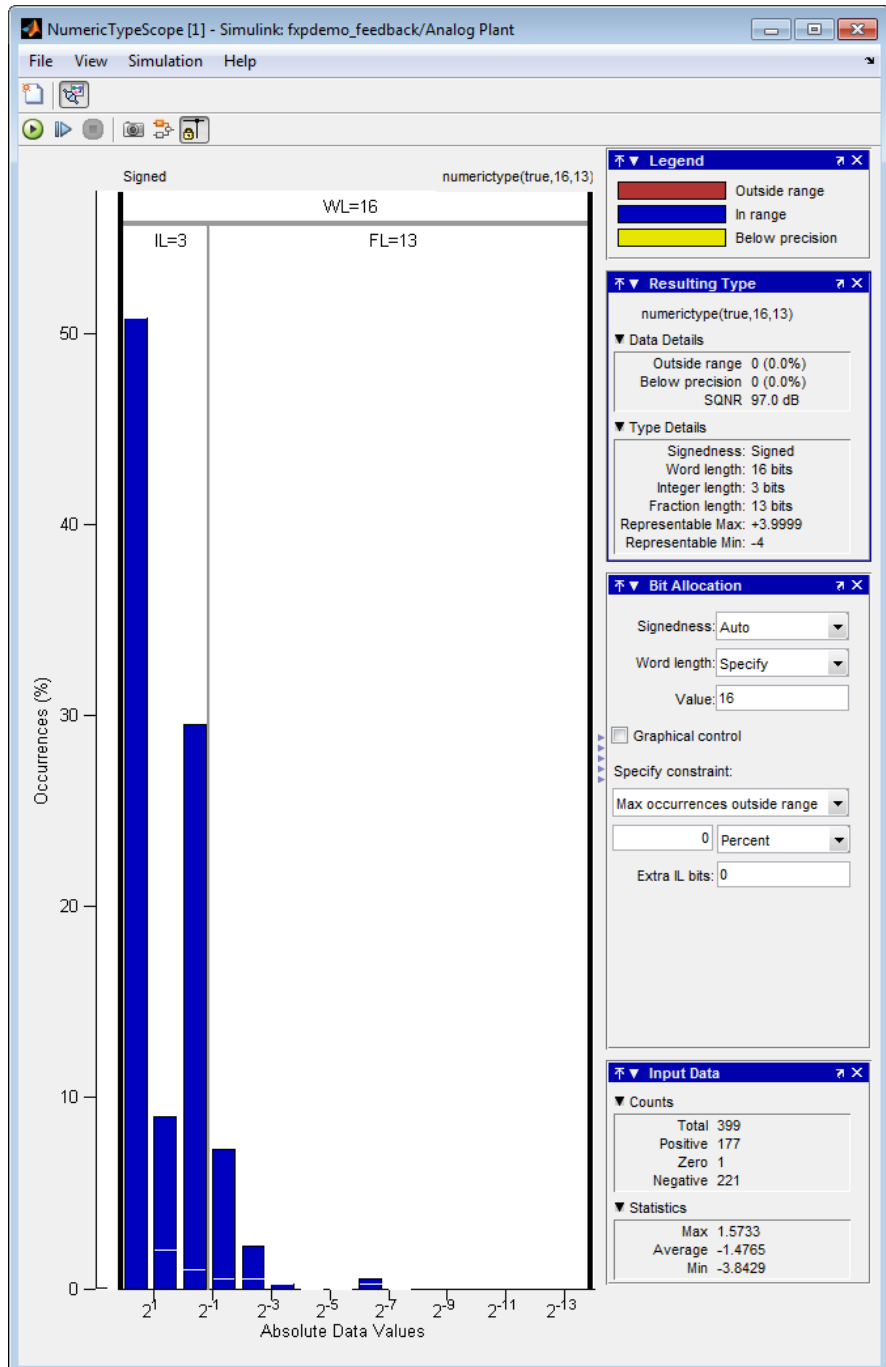
Handle of the Simulink signal that you want to view with the scope. To get the handle of the currently selected signal, at the MATLAB® command line, enter `nts({gs1})`.



### **PORT**

Index of the output port that you want to view with the scope. If the block has more than one output port, you must specify the index. The scope cannot connect to more than one output port.







## The NumericTypeScope Window

The NumericTypeScope opens with the default toolbars displayed at the top of the window and the dialog panels to the right.



Button	Action
	New NumericTypeScope.
	Connect to Simulink signal. The scope connects to the currently selected signal. If a block with only one output port is selected and the <b>Connect scope on selection of</b> is set to <b>Signal lines or blocks</b> , connects to the output port of the selected block. For more information, see Sources Pane on page 12.

After connecting the scope to a signal in a Simulink model, the scope displays an additional toolbar with the following options:

Button	Action
	Stop simulation
	Start simulation
	Simulate one step
	Snapshot. Freezes the display so that you can examine the results. To reenale display refreshing, click the button again.
	Highlight Simulink signal.
	Persistent. By default, the scope makes a persistent connection to the selected signal. If you want to view different signals during the simulation, click this button to make a floating connection. You can then select any signal in the model and the scope displays it.

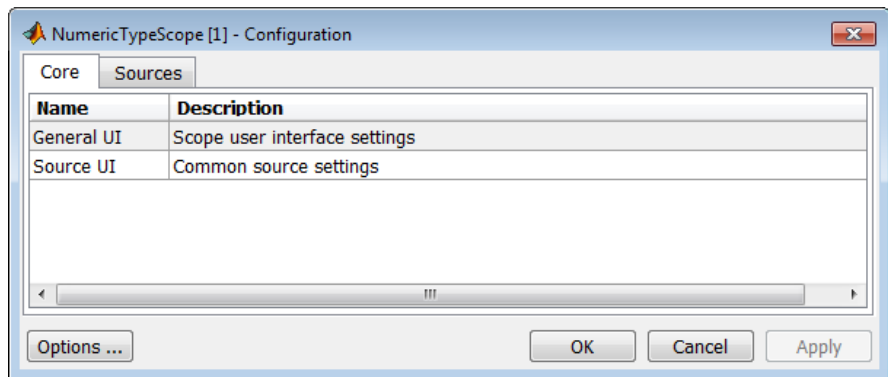
## Dialog Boxes and Panels

- “Configuration Dialog Box” on page 1-9
- “Dialog Panels” on page 1-13

### Configuration Dialog Box

Use the NumericTypeScope configuration dialog box to control the behavior and appearance of the scope window.

To open the **Configuration** dialog box, from the scope main menu, select **File > Configuration > Edit**, or, with the scope as your active window, press the N key.



For information about each pane, see Core Pane on page 10 and Sources Pane on page 12.

To save configuration settings for future use, select **File > Configuration > Save as**. The configuration settings you save become the default configuration settings for the NumericTypeScope.

---

## Caution

Before saving your own set of configuration settings in the `matlab/toolbox/fixpoint` folder, save a backup copy of the default configuration settings in another location. If you do not save a backup copy of the default configuration settings, you cannot restore these settings at a later time.

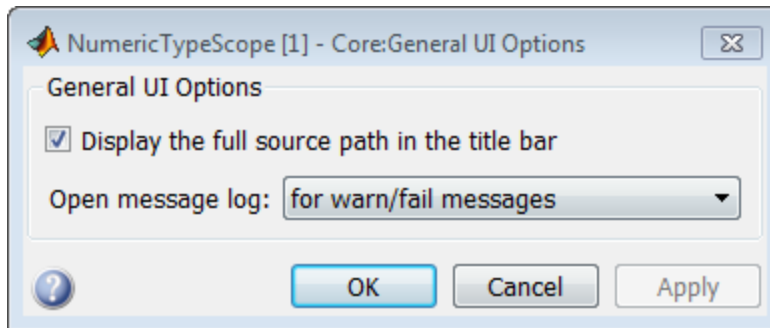
---

To save your configuration settings for future use, save them in the `matlab/toolbox/fixpoint` folder with the file name `NumericTypeScopeSL.cfg`. You can re-save your configuration settings at anytime, but you must save them in this folder with this filename.

## Core Pane

The **Core** pane controls the general settings of the scope.

To open the **Core:General UI Options** dialog box, select **General UI** and then click **Options**.



- **Display the full source path in the title bar**—Select this check box to display the full path to the selected block in the model. Otherwise, the scope displays only the block name.
- **Open message log**—Control when the Message Log window opens. The Message log window helps you debug issues with the scope. Choose to open the Message Log window for any of these conditions:



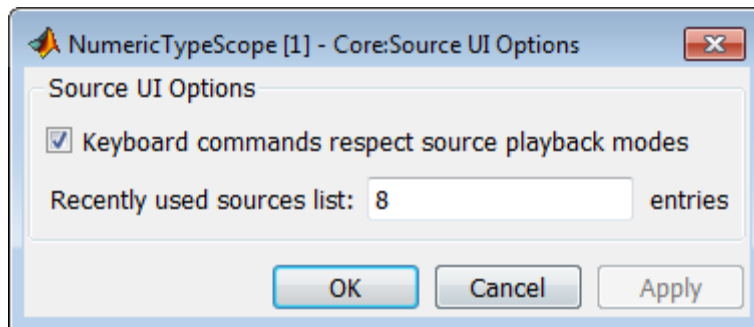
- for any new messages
- for warn/fail messages
- only for fail messages
- manually

The option defaults to for warn/fail messages.

You can open the Message Log at any time by selecting **Help > Message Log** or by pressing **Ctrl+M**. The **Message Log** dialog box provides a system level record of loaded configuration settings and registered extensions. The Message Log displays summaries and details of each message, and you can filter the display of messages by Type and Category.

- **Type**—Select the type of messages to display in the Message Log. You can select All, Info, Warn, or Fail. Type defaults to All.
- **Category**—Select the category of messages to display in the Message Log. You can select All, Configuration, or Extension. The scope uses Configuration messages to indicate when new configuration files are loaded, and Extension messages to indicate when components are registered. Category defaults to All.

To open the **Core:Source UI Options** dialog box, select **General UI** and then click **Options**.



- **Keyboard commands respect source playback modes**—Has no effect. The following table shows the keyboard shortcut mapping. You cannot disable this mapping.

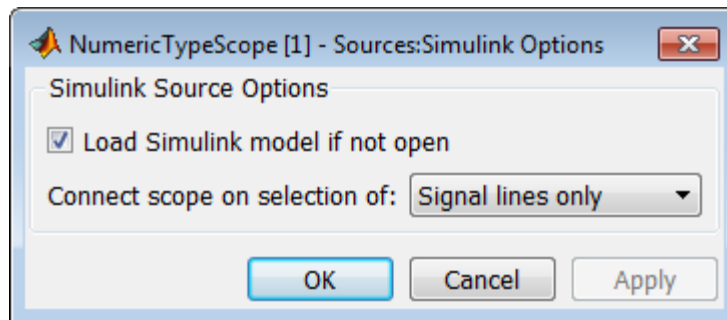
Action	Keyboard Shortcut
Open new NumericTypeScope	Insert
Change configuration	N
Display keyboard help	K
Play simulation	P
Pause simulation	Space
Stop simulation	S
Step forward	Right arrow, Page down

- **Recently used sources list**—Sets the maximum number of recently used sources displayed under the **Files** menu option.

### Sources Pane

The **Sources** pane controls how the scope connects to Simulink. You cannot disable the Simulink source.

To open the **Sources:Simulink Options** dialog box, select the **Sources** tab and then click **Options**.

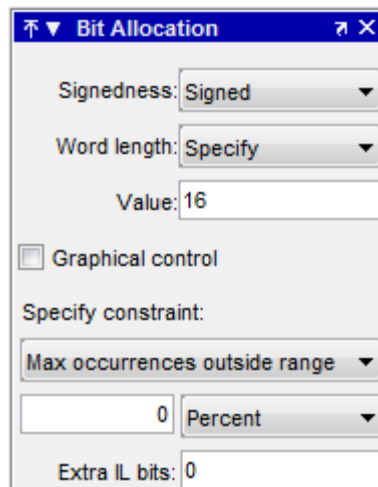


- **Load Simulink model if not open**—When selected, if you specify a signal in a Simulink model that is not currently open, the scope opens the model.
- **Connect scope on selection of**—Connects the scope only when you select signal lines or when you select signal lines or blocks. If you select **Signal lines** or **blocks**, the scope cannot connect to blocks that have more than one output port.

## Dialog Panels

### Bit Allocation Panel

The scope **Bit Allocation** panel provides options for specifying data type criteria. Adjust these criteria to observe the effect on suggested numerictype. For streaming data, the suggested numerictype adjusts over time in order to continue to satisfy the specified criteria.



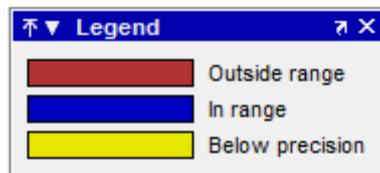
You can:

- Specify a known word length and signedness and, using **Specify constraint**, add additional constraints such as the maximum number of occurrences outside range or the smallest value that the suggested data type must represent.

- Specify **Integer length** and **Fraction length** constraints so that the scope suggests an appropriate word length.
- Set the **Signedness** and **Word length** to Auto so that the scope suggests values for these parameters.
- Enable **Graphical control** and use the cursors on either side of the binary point to adjust the fraction length and observe the effect on the suggested numeric type on the input data. For example, you can see the number of values that are outside range, below precision, or both. You can also view representable minimum and maximum values of the data type.
- Specify extra bits for either the fraction length or the integer length. The extra bits act as a safety margin to minimize the risk of overflow and precision loss.

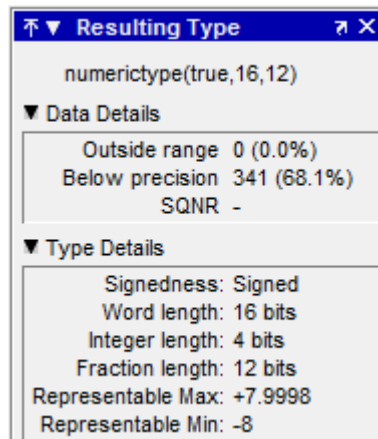
## Legend

The scope **Legend** panel informs you which colors the scope uses to indicate values. These colors represent values that are outside range, in range, or below precision when displayed in the scope.



## Resulting Type

The **Resulting Type** panel describes the fixed-point data type as defined by scope settings. By manipulating the visual display (via the **Bit Allocation** panel or with the cursors), you can change the data type specification.

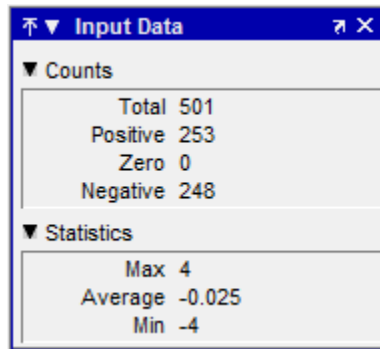


The **Data Details** section displays the percentage of values that fall outside range or below precision with the `numerictype` object located at the top of this panel. SQNR (Signal Quantization Noise Ratio) varies depending on the signal. If the parameter has no value, then there is not enough data to calculate the SQNR. When scope information or the `numerictype` changes, the SQNR resets.

The **Type Details** section provides details about the fixed-point data type. You can copy the `numerictype` specification by right-clicking the **Resulting Type** pane and then selecting `Copy numerictype`.

### **Input Data**

The **Input Data** panel provides statistical information about the values currently displayed in the `NumericScopeType`.



## Examples **Connect a NumericTypeScope to a signal in a Simulink model**

Open a NumericTypeScope window and connect to a signal.

Open the model.

fxpdemo\_approx

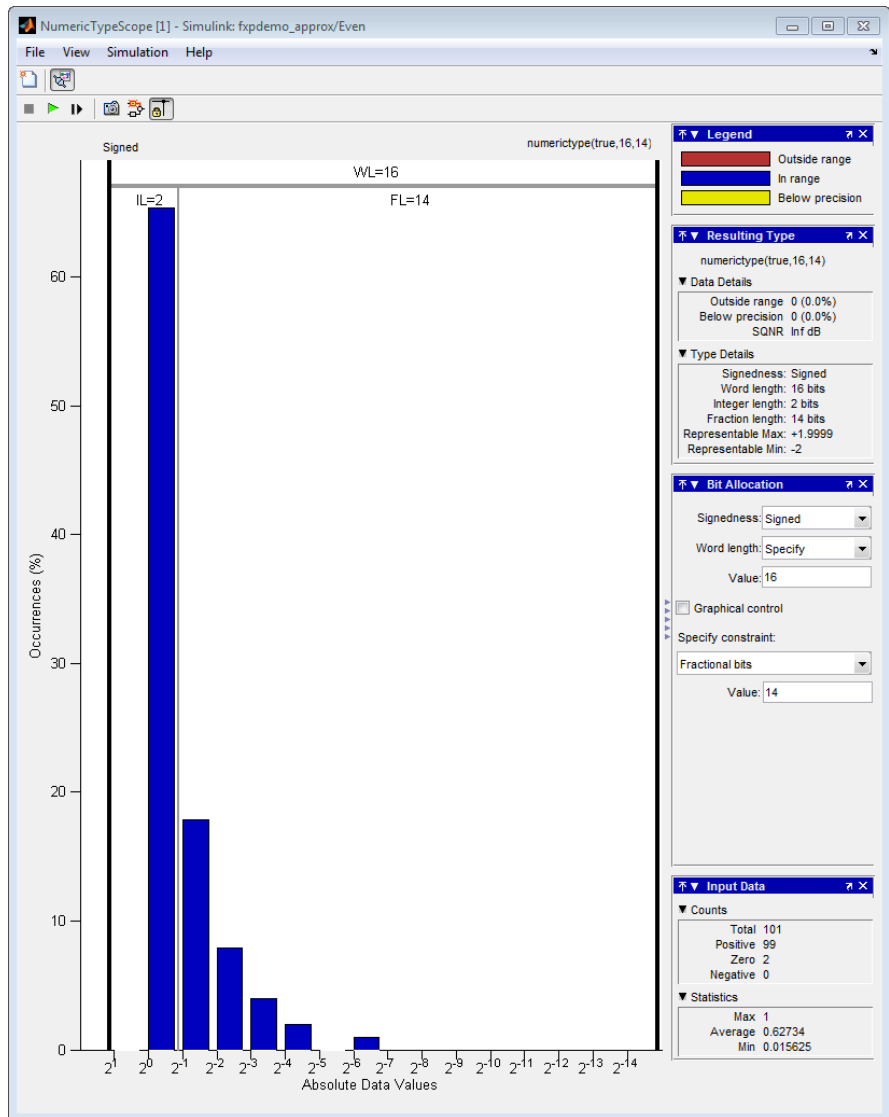
Open a NumericTypeScope.

nts

In the fxpdemo\_approx model, select the yEven signal.

In the NumericTypeScope window, select **File > Connect to Simulink Signal**.

Run the simulation to view the dynamic range of the output. The NumericTypeScope suggests a data type for the output.



## **Connect a NumericTypeScope to a block output port**

Connect a NumericTypeScope to a block output port and view the dynamic range of block output.

Specify the block path and name and the output port number.

```
blk='fxpdemo_approx/Even';  
nts({blk,1})
```

Run the simulation to view the dynamic range of the output. The NumericTypeScope suggests a data type for the output.

## **Specify a Simulink signal to display**

Connect a NumericTypeScope to a signal selected in a model.

Open the model.

```
fxpdemo_approx
```

In the fxpdemo\_approx model, select the yEven signal.

Open a NumericTypeScope, specifying the line handle of the selected signal.

```
nts({gs1})
```

## **See Also**

[hist](#) | [log2](#) | [numerictypescope](#)



**Purpose** Show overflows from most recent fixed-point simulation

---

**Note** showfixptsimerrors will be removed in a future release. Use fxptdlg instead.

---

**Syntax** showfixptsimerrors

**Description** The showfixptsimerrors script displays any overflows from the most recent fixed-point simulation. This information is also visible in the Fixed-Point Tool.

**See Also** autofixexp | fxptdlg

# showfixptsimranges

---

**Purpose** Show logged maximum values, minimum values, and overflow data from fixed-point simulation

---

**Note** `showfixptsimranges` will be removed in a future release. Use `fxptdlg` instead.

---

**Syntax** `showfixptsimranges`  
`showfixptsimranges(action)`

**Description** `showfixptsimranges` displays the logged maximum values, minimum values, and overflow data from the most recent fixed-point simulation in the MATLAB Command Window.

`showfixptsimranges(action)` stores the logged maximum values, minimum values, and overflow data from the most recent fixed-point simulation in the workspace variable `FixPtSimRanges`. If `action` is `'verbose'`, the logged data also appears in the MATLAB Command Window. If `action` is `'quiet'`, no data appears.

**See Also** `autofixexp` | `fxptdlg`

# Selected Bibliography

---

- [1] Burrus, C.S., J.H. McClellan, A.V. Oppenheim, T.W. Parks, R.W. Schafer, and H.W. Schuessler, *Computer-Based Exercises for Signal Processing Using MATLAB*, Prentice Hall, Englewood Cliffs, New Jersey, 1994.
- [2] Franklin, G.F., J.D. Powell, and M.L. Workman, *Digital Control of Dynamic Systems, Second Edition*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1990.
- [3] *Handbook For Digital Signal Processing*, edited by S.K. Mitra and J.F. Kaiser, John Wiley & Sons, Inc., New York, 1993.
- [4] Hanselmann, H., "Implementation of Digital Controllers — A Survey," *Automatica*, Vol. 23, No. 1, pp. 7-32, 1987.
- [5] Jackson, L.B., *Digital Filters and Signal Processing, Second Edition*, Kluwer Academic Publishers, Seventh Printing, Norwell, Massachusetts, 1993.
- [6] Middleton, R. and G. Goodwin, *Digital Control and Estimation — A Unified Approach*, Prentice Hall, Englewood Cliffs, New Jersey. 1990.
- [7] Moler, C., "Floating points: IEEE Standard unifies arithmetic model," Cleve's Corner, The MathWorks, Inc., 1996. You can find this article at [http://www.mathworks.com/company/newsletters/news\\_notes/clevescorner/index.html](http://www.mathworks.com/company/newsletters/news_notes/clevescorner/index.html).
- [8] Ogata, K., *Discrete-Time Control Systems, Second Edition*, Prentice Hall, Englewood Cliffs, New Jersey, 1995.
- [9] Roberts, R.A. and C.T. Mullis, *Digital Signal Processing*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1987.



This glossary defines terms related to fixed-point data types and numbers. These terms may appear in some or all of the documents that describe MathWorks® products that have fixed-point support.

## **arithmetic shift**

Shift of the bits of a binary word for which the sign bit is recycled for each bit shift to the right. A zero is incorporated into the least significant bit of the word for each bit shift to the left. In the absence of overflows, each arithmetic shift to the right is equivalent to a division by 2, and each arithmetic shift to the left is equivalent to a multiplication by 2.

*See also* binary point, binary word, bit, logical shift, most significant bit

## **bias**

Part of the numerical representation used to interpret a fixed-point number. Along with the slope, the bias forms the scaling of the number. Fixed-point numbers can be represented as

$$\textit{real-world value} = (\textit{slope} \times \textit{stored integer}) + \textit{bias}$$

where the slope can be expressed as

$$\textit{slope} = \textit{fractional slope} \times 2^{\textit{exponent}}$$

*See also* fixed-point representation, fractional slope, integer, scaling, slope, [Slope Bias]

## **binary number**

Value represented in a system of numbers that has two as its base and that uses 1's and 0's (bits) for its notation.

*See also* bit

**binary point**

Symbol in the shape of a period that separates the integer and fractional parts of a binary number. Bits to the left of the binary point are integer bits and/or sign bits, and bits to the right of the binary point are fractional bits.

*See also* binary number, bit, fraction, integer, radix point

**binary point-only scaling**

Scaling of a binary number that results from shifting the binary point of the number right or left, and which therefore can only occur by powers of two.

*See also* binary number, binary point, scaling

**binary word**

Fixed-length sequence of bits (1's and 0's). In digital hardware, numbers are stored in binary words. The way in which hardware components or software functions interpret this sequence of 1's and 0's is described by a data type.

*See also* bit, data type, word

**bit**

Smallest unit of information in computer software or hardware. A bit can have the value 0 or 1.

**ceiling (round toward)**

Rounding mode that rounds to the closest representable number in the direction of positive infinity. This is equivalent to the `ceil` mode in Fixed-Point Toolbox™ software.

*See also* convergent rounding, floor (round toward), nearest (round toward), rounding, truncation, zero (round toward)

**contiguous binary point**

Binary point that occurs within the word length of a data type. For example, if a data type has four bits, its contiguous binary point must be understood to occur at one of the following five positions:

.0000

0.000

00.00

000.0

0000.

*See also* data type, noncontiguous binary point, word length

**convergent rounding**

Rounding mode that rounds to the nearest allowable quantized value. Numbers that are exactly halfway between the two nearest allowable quantized values are rounded up only if the least significant bit (after rounding) would be set to 0.

*See also* ceiling (round toward), floor (round toward), nearest (round toward), rounding, truncation, zero (round toward)

**data type**

Set of characteristics that define a group of values. A fixed-point data type is defined by its word length, its fraction length, and whether it is signed or unsigned. A floating-point data type is defined by its word length and whether it is signed or unsigned.

*See also* fixed-point representation, floating-point representation, fraction length, signedness, word length

**data type override**

Parameter in the Fixed-Point Tool that allows you to set the output data type and scaling of fixed-point blocks on a system or subsystem level.

*See also* data type, scaling

**exponent**

Part of the numerical representation used to express a floating-point or fixed-point number.

1. Floating-point numbers are typically represented as

$$\textit{real-world value} = \textit{mantissa} \times 2^{\textit{exponent}}$$

2. Fixed-point numbers can be represented as

$$\textit{real-world value} = (\textit{slope} \times \textit{stored integer}) + \textit{bias}$$

where the slope can be expressed as

$$\textit{slope} = \textit{fractional slope} \times 2^{\textit{exponent}}$$

The exponent of a fixed-point number is equal to the negative of the fraction length:

$$\textit{exponent} = -1 \times \textit{fraction length}$$

*See also* bias, fixed-point representation, floating-point representation, fraction length, fractional slope, integer, mantissa, slope

### **fixed-point representation**

Method for representing numerical values and data types that have a set range and precision.

1. Fixed-point numbers can be represented as

$$\textit{real-world value} = (\textit{slope} \times \textit{stored integer}) + \textit{bias}$$

where the slope can be expressed as

$$\textit{slope} = \textit{fractional slope} \times 2^{\textit{exponent}}$$

The slope and the bias together represent the scaling of the fixed-point number.

2. Fixed-point data types can be defined by their word length, their fraction length, and whether they are signed or unsigned.

*See also* bias, data type, exponent, fraction length, fractional slope, integer, precision, range, scaling, slope, word length



**floating-point representation**

Method for representing numerical values and data types that can have changing range and precision.

1. Floating-point numbers can be represented as

$$\text{real-world value} = \text{mantissa} \times 2^{\text{exponent}}$$

2. Floating-point data types are defined by their word length.

*See also* data type, exponent, mantissa, precision, range, word length

**floor (round toward)**

Rounding mode that rounds to the closest representable number in the direction of negative infinity.

*See also* ceiling (round toward), convergent rounding, nearest (round toward), rounding, truncation, zero (round toward)

**fraction**

Part of a fixed-point number represented by the bits to the right of the binary point. The fraction represents numbers that are less than one.

*See also* binary point, bit, fixed-point representation

**fraction length**

Number of bits to the right of the binary point in a fixed-point representation of a number.

*See also* binary point, bit, fixed-point representation, fraction

**fractional slope**

Part of the numerical representation used to express a fixed-point number. Fixed-point numbers can be represented as

$$\text{real-world value} = (\text{slope} \times \text{stored integer}) + \text{bias}$$

where the slope can be expressed as

$$\text{slope} = \text{fractional slope} \times 2^{\text{exponent}}$$

The term *slope adjustment* is sometimes used as a synonym for fractional slope.

*See also* bias, exponent, fixed-point representation, integer, slope

**full range**

The broadest range available for a data type. From  $-\infty$  to  $\infty$  for floating-point types. For integer types, the representable range is the range from the smallest to largest integer value (finite) the type can represent. For example, from -128 to 127 for a signed 8-bit integer. Also known as representable range.

**guard bits**

Extra bits in either a hardware register or software simulation that are added to the high end of a binary word to ensure that no information is lost in case of overflow.

*See also* binary word, bit, overflow

**incorrect range**

A range that is too restrictive and does not include values that can actually occur in the model element. A range that is too broad is not considered incorrect because it will not lead to overflow.

*See also* range analysis

**integer**

1. Part of a fixed-point number represented by the bits to the left of the binary point. The integer represents numbers that are greater than or equal to one.

2. Also called the "stored integer." The raw binary number, in which the binary point is assumed to be at the far right of the word. The integer is part of the numerical representation used to express a fixed-point number. Fixed-point numbers can be represented as

$$\text{real-world value} = 2^{-\text{fraction length}} \times \text{stored integer}$$

or

$$\text{real-world value} = (\text{slope} \times \text{stored integer}) + \text{bias}$$

where the slope can be expressed as

$$\text{slope} = \text{fractional slope} \times 2^{\text{exponent}}$$

*See also* bias, fixed-point representation, fractional slope, integer, real-world value, slope

### **integer length**

Number of bits to the left of the binary point in a fixed-point representation of a number.

*See also* binary point, bit, fixed-point representation, fraction length, integer

### **least significant bit (LSB)**

Bit in a binary word that can represent the smallest value. The LSB is the rightmost bit in a big-endian-ordered binary word. The weight of the LSB is related to the fraction length according to

$$\text{weight of LSB} = 2^{-\text{fraction length}}$$

*See also* big-endian, binary word, bit, most significant bit

**logical shift**

Shift of the bits of a binary word, for which a zero is incorporated into the most significant bit for each bit shift to the right and into the least significant bit for each bit shift to the left.

*See also* arithmetic shift, binary point, binary word, bit, most significant bit

**mantissa**

Part of the numerical representation used to express a floating-point number. Floating-point numbers are typically represented as

$$\text{real - world value} = \text{mantissa} \times 2^{\text{exponent}}$$

*See also* exponent, floating-point representation

**model element**

Entities in a model that range analysis software tracks, for example, blocks, signals, parameters, block internal data (such as accumulators, products).

*See also* range analysis

**most significant bit (MSB)**

Bit in a binary word that can represent the largest value. The MSB is the leftmost bit in a big-endian-ordered binary word.

*See also* binary word, bit, least significant bit

**nearest (round toward)**

Rounding mode that rounds to the closest representable number, with the exact midpoint rounded to the closest representable number in the direction of positive infinity. This is equivalent to the nearest mode in Fixed-Point Toolbox software.

*See also* ceiling (round toward), convergent rounding, floor (round toward), rounding, truncation, zero (round toward)

**noncontiguous binary point**

Binary point that is understood to fall outside the word length of a data type. For example, the binary point for the following 4-bit word is understood to occur two bits to the right of the word length,

$$0000\_.$$

thereby giving the bits of the word the following potential values:

$$2^5 2^4 2^3 2^2 \_.$$

*See also* binary point, data type, word length

**one's complement representation**

Representation of signed fixed-point numbers. Negating a binary number in one's complement requires a bitwise complement. That is, all 0's are flipped to 1's and all 1's are flipped to 0's. In one's complement notation there are two ways to represent zero. A binary word of all 0's represents "positive" zero, while a binary word of all 1's represents "negative" zero.

*See also* binary number, binary word, sign/magnitude representation, signed fixed-point, two's complement representation

**overflow**

Situation that occurs when the magnitude of a calculation result is too large for the range of the data type being used. In many cases you can choose to either saturate or wrap overflows.

*See also* saturation, wrapping

**padding**

Extending the least significant bit of a binary word with one or more zeros.

*See also* least significant bit

**precision**

1. Measure of the smallest numerical interval that a fixed-point data type and scaling can represent, determined by the value of the number's least significant bit. The precision is given by the slope, or the number

of fractional bits. The term *resolution* is sometimes used as a synonym for this definition.

2. Measure of the difference between a real-world numerical value and the value of its quantized representation. This is sometimes called quantization error or quantization noise.

*See also* data type, fraction, least significant bit, quantization, quantization error, range, slope

### **Q format**

Representation used by Texas Instruments™ to encode signed two's complement fixed-point data types. This fixed-point notation takes the form

$$Qm.n$$

where

- $Q$  indicates that the number is in Q format.
- $m$  is the number of bits used to designate the two's complement integer part of the number.
- $n$  is the number of bits used to designate the two's complement fractional part of the number, or the number of bits to the right of the binary point.

In Q format notation, the most significant bit is assumed to be the sign bit.

*See also* binary point, bit, data type, fixed-point representation, fraction, integer, two's complement

### **quantization**

Representation of a value by a data type that has too few bits to represent it exactly.

*See also* bit, data type, quantization error

### **quantization error**

Error introduced when a value is represented by a data type that has too few bits to represent it exactly, or when a value is converted from

one data type to a shorter data type. Quantization error is also called quantization noise.

*See also* bit, data type, quantization

**radix point**

Symbol in the shape of a period that separates the integer and fractional parts of a number in any base system. Bits to the left of the radix point are integer and/or sign bits, and bits to the right of the radix point are fraction bits.

*See also* binary point, bit, fraction, integer, sign bit

**range**

Span of numbers that a certain data type can represent.

*See also* data type, full range, precision, representable range

**range analysis**

Static analysis of model to derive minimum and maximum range values for elements in the model. The software statically analyzes the ranges of the individual computations in the model based on specified design ranges, inputs, and the semantics of the calculation.

**real-world value**

Stored integer value with fixed-point scaling applied. Fixed-point numbers can be represented as

$$\text{real-world value} = 2^{-\text{fraction length}} \times \text{stored integer}$$

or

$$\text{real-world value} = (\text{slope} \times \text{stored integer}) + \text{bias}$$

where the slope can be expressed as

$$\text{slope} = \text{fractional slope} \times 2^{\text{exponent}}$$

*See also* integer

**representable range**

The broadest range available for a data type. From  $-\infty$  to  $\infty$  for floating-point types. For integer types, the representable range is the range from the smallest to largest integer value (finite) the type can represent. For example, from -128 to 127 for a signed 8-bit integer. Also known as full range.

**resolution**

*See* **precision**

**rounding**

Limiting the number of bits required to express a number. One or more least significant bits are dropped, resulting in a loss of precision. Rounding is necessary when a value cannot be expressed exactly by the number of bits designated to represent it.



*See also* bit, ceiling (round toward), convergent rounding, floor (round toward), least significant bit, nearest (round toward), precision, truncation, zero (round toward)

**saturation**

Method of handling numeric overflow that represents positive overflows as the largest positive number in the range of the data type being used, and negative overflows as the largest negative number in the range.

*See also* overflow, wrapping

**scaled double**

A double data type that retains fixed-point scaling information. For example, in Simulink and Fixed-Point Toolbox software you can use data type override to convert your fixed-point data types to scaled doubles. You can then simulate to determine the ideal floating-point behavior of your system. After you gather that information you can turn data type override off to return to fixed-point data types, and your quantities still have their original scaling information because it was held in the scaled double data types.

**scaling**

1. Format used for a fixed-point number of a given word length and signedness. The slope and bias together form the scaling of a fixed-point number.
2. Changing the slope and/or bias of a fixed-point number without changing the stored integer.

*See also* bias, fixed-point representation, integer, slope

**shift**

Movement of the bits of a binary word either toward the most significant bit ("to the left") or toward the least significant bit ("to the right"). Shifts to the right can be either logical, where the spaces emptied at the front of the word with each shift are filled in with zeros, or arithmetic, where the word is sign extended as it is shifted to the right.

*See also* arithmetic shift, logical shift, sign extension

**sign bit**

Bit (or bits) in a signed binary number that indicates whether the number is positive or negative.

*See also* binary number, bit

**sign extension**

Addition of bits that have the value of the most significant bit to the high end of a two's complement number. Sign extension does not change the value of the binary number.

*See also* binary number, guard bits, most significant bit, two's complement representation, word

**sign/magnitude representation**

Representation of signed fixed-point or floating-point numbers. In sign/magnitude representation, one bit of a binary word is always the dedicated sign bit, while the remaining bits of the word encode the magnitude of the number. Negation using sign/magnitude representation consists of flipping the sign bit from 0 (positive) to 1 (negative), or from 1 to 0.

*See also* binary word, bit, fixed-point representation, floating-point representation, one's complement representation, sign bit, signed fixed-point, signedness, two's complement representation

**signed fixed-point**

Fixed-point number or data type that can represent both positive and negative numbers.

*See also* data type, fixed-point representation, signedness, unsigned fixed-point

**signedness**

The signedness of a number or data type can be signed or unsigned. Signed numbers and data types can represent both positive and negative values, whereas unsigned numbers and data types can only represent values that are greater than or equal to zero.

*See also* data type, sign bit, sign/magnitude representation, signed fixed-point, unsigned fixed-point

**slope**

Part of the numerical representation used to express a fixed-point number. Along with the bias, the slope forms the scaling of a fixed-point number. Fixed-point numbers can be represented as

$$\textit{real-world value} = (\textit{slope} \times \textit{stored integer}) + \textit{bias}$$

where the slope can be expressed as

$$\textit{slope} = \textit{fractional slope} \times 2^{\textit{exponent}}$$

*See also* bias, fixed-point representation, fractional slope, integer, scaling, [Slope Bias]

**slope adjustment**

*See fractional slope*

**[Slope Bias]**

Representation used to define the scaling of a fixed-point number.

*See also* bias, scaling, slope

**stored integer**

*See integer*

**trivial scaling**

Scaling that results in the real-world value of a number being simply equal to its stored integer value:

$$\textit{real - world value} = \textit{stored integer}$$

In [Slope Bias] representation, fixed-point numbers can be represented as

$$\textit{real-world value} = (\textit{slope} \times \textit{stored integer}) + \textit{bias}$$

In the trivial case, slope = 1 and bias = 0.

In terms of binary point-only scaling, the binary point is to the right of the least significant bit for trivial scaling, meaning that the fraction length is zero:

$$\textit{real - world value} = \textit{stored integer} \times 2^{-\textit{fraction length}} = \textit{stored integer} \times 2^0$$

Scaling is always trivial for pure integers, such as `int8`, and also for the true floating-point types `single` and `double`.

*See also* bias, binary point, binary point-only scaling, fixed-point representation, fraction length, integer, least significant bit, scaling, slope, [Slope Bias]

**truncation**

Rounding mode that drops one or more least significant bits from a number.

*See also* ceiling (round toward), convergent rounding, floor (round toward), nearest (round toward), rounding, zero (round toward)

**two's complement representation**

Common representation of signed fixed-point numbers. Negation using signed two's complement representation consists of a translation into one's complement followed by the binary addition of a one.

*See also* binary word, one's complement representation, sign/magnitude representation, signed fixed-point

**unsigned fixed-point**

Fixed-point number or data type that can only represent numbers greater than or equal to zero.

*See also* data type, fixed-point representation, signed fixed-point, signedness

**word**

Fixed-length sequence of binary digits (1's and 0's). In digital hardware, numbers are stored in words. The way hardware components or software functions interpret this sequence of 1's and 0's is described by a data type.

*See also* binary word, data type

**word length**

Number of bits in a binary word or data type.

*See also* binary word, bit, data type

**wrapping**

Method of handling overflow. Wrapping uses modulo arithmetic to cast a number that falls outside of the representable range the data type being used back into the representable range.

*See also* data type, overflow, range, saturation

**zero (round toward)**

Rounding mode that rounds to the closest representable number in the direction of zero. This is equivalent to the `fix` mode in Fixed-Point Toolbox software.

*See also* ceiling (round toward), convergent rounding, floor (round toward), nearest (round toward), rounding, truncation

## A

- autofixexp function 1-2
- automatic scaling
  - autofixexp 1-2
  - fixpt\_instrument\_purge 1-4
  - script 1-2
- autoscaling
  - autofixexp 1-2
  - fixpt\_instrument\_purge 1-4
  - script 1-2

## F

- fixpt\_instrument\_purge function 1-4

## functions

- autofixexp 1-2
- fixpt\_instrument\_purge 1-4
- showfixptsimerrors 1-19
- showfixptsimranges 1-20

## S

- showfixptsimerrors function 1-19
- showfixptsimranges function 1-20