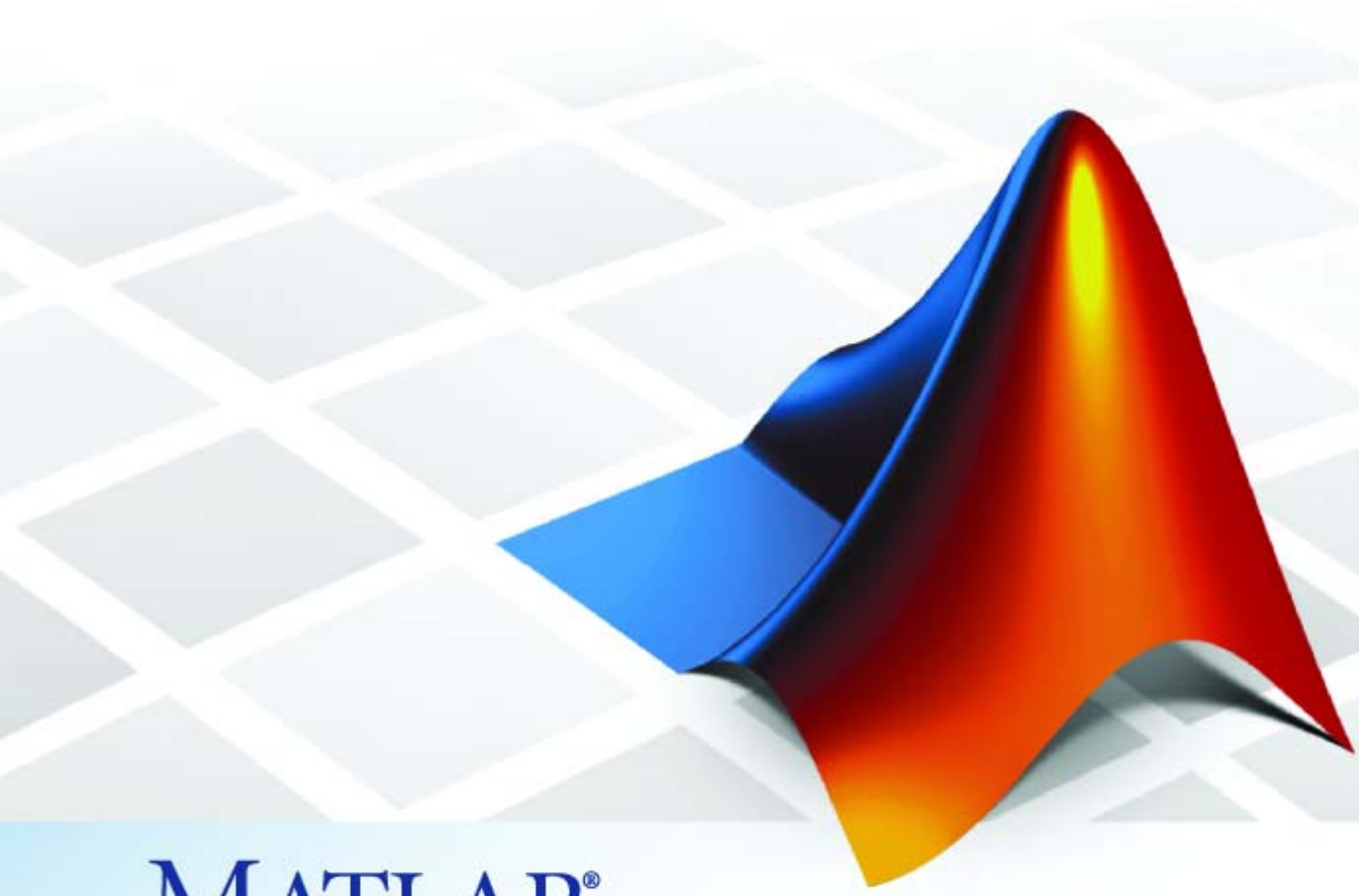
Stateflow® and Stateflow® Coder 6
Reference



MATLAB®
&SIMULINK®

The MathWorks
Accelerating the pace of engineering and science

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.

Stateflow and Stateflow Coder Reference

Trademarks

MATLAB, Simulink, Stateflow, Handle Graphics, Real-Time Workshop, and xPC TargetBox are registered trademarks, and SimBiology, SimEvents, and SimHydraulics are trademarks of The MathWorks, Inc.

Other product or brand names are 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.

Revision History

March 2006	Online only	New for Version 6.4 (Release 2006a)
September 2006	Online only	Revised for Stateflow 6.5 (Release R2006b)

Contents

Functions — By Category

Object Retrieval

`sfclipboard`	Get Stateflow clipboard object
`sfgco`	Get most recently selected objects in Stateflow chart
`sfroot`	Get Stateflow root object

Chart Creation

`sfnew`	Create Simulink® model containing empty Stateflow block
`stateflow`	Create Simulink model containing empty Stateflow chart, and open Stateflow library window

Chart Input/Output

`sfclose`	Close Stateflow chart
`sfopen`	Open Stateflow machine
`sfprint`	Print graphical view of Stateflow charts
`sfsave`	Save Stateflow machine in current directory

Graphical User Interface

`sfdebugger`	Open Stateflow debugger
`sfexplr`	Start Model Explorer
`sflib`	Open Stateflow library window

Help

`sfhelp`	Open Stateflow online help

2

Functions — Alphabetical List

sfclipboard

Purpose

Get Stateflow clipboard object

Syntax

```
object = sfclipboard
```

Description

`object = sfclipboard` returns a handle to the Stateflow clipboard object. Use the clipboard object to copy objects from one container object to another, as described in “Copying Objects” in the online Stateflow API Reference.

See Also

`sfgco`, `sfnew`, `sfroot`, `stateflow`

Purpose

Close Stateflow chart

Syntax

```
sfclose
sfclose( 'Chart_Name' )
sfclose( Chart_Handle )
sfclose( 'All' )
```

Arguments

'Chart_Name'	Name of a Stateflow chart.
Chart_Handle	Handle to a Stateflow chart.
'All'	Literal string that directs Stateflow to close all open or minimized Stateflow charts.

Description

sfclose closes the current Stateflow chart.

sfclose(*'Chart_Name'*) closes the Stateflow chart named **Chart_Name**.

sfclose(*Chart_Handle*) closes the Stateflow chart whose handle is *Chart_Handle*.

sfclose(*'All'*) closes all open or minimized Stateflow charts.

See Also

sfopen, sfnew, stateflow

sfdebugger

Purpose

Open Stateflow debugger

Syntax

```
sfdebugger
sfdebugger( 'Machine_Name' )
sfdebugger( Machine_Handle )
sfdebugger( Machine_Id )
```

Arguments

'Model_Name'	String name of a Stateflow machine.
Machine_Handle	Handle to a Stateflow machine.
Machine_Id	ID of a Stateflow machine.

Description

sfdebugger opens the Stateflow debugger for the currently selected Stateflow machine.

sfdebugger('Machine_Name') opens the Stateflow debugger for the Stateflow machine called **Machine_Name**.

sfdebugger(Machine_Handle) opens the Stateflow debugger for the Stateflow machine whose handle is *Model_Handle*.

sfdebugger(Machine_Id) opens the Stateflow debugger for the Stateflow machine whose `Id` is *Machine_Id*.

See Also

`sfexplr`, `sfhelp`, `sflib`

Purpose

Start Model Explorer

Syntax

```
sfexplr
```

Description

`sfexplr` starts the Model Explorer. For more information, see "The Model Explorer" in the online Simulink documentation.

See Also

`sfdebugger`, `sfhelp`, `sflib`

sfgco

Purpose

Get most recently selected objects in Stateflow chart

Syntax

```
object = sfgco
```

Description

object = *sfgco* returns a handle or vector of handles to the most recently selected objects in a Stateflow chart, as follows:

If ...	sfgco returns ...
No Stateflow charts are open, or no open charts were edited or otherwise manipulated	Empty matrix
There is no selection list	Handle to the Stateflow chart most recently clicked
You select one object in a Stateflow chart	Handle to the selected object
You select multiple objects in a Stateflow chart	Vector of handles to the selected objects
You select multiple objects in multiple Stateflow charts	Vector of handles to the most recently selected objects in the most recently selected chart

See Also

`sfnew`, `stateflow`

Purpose

Open Stateflow online help

Syntax

```
sfhelp
```

Description

`sfhelp` opens Stateflow online help in the Help browser.

See Also

`sfexplr`, `sfnew`, `sfprint`, `sfsave`, `stateflow`

Purpose

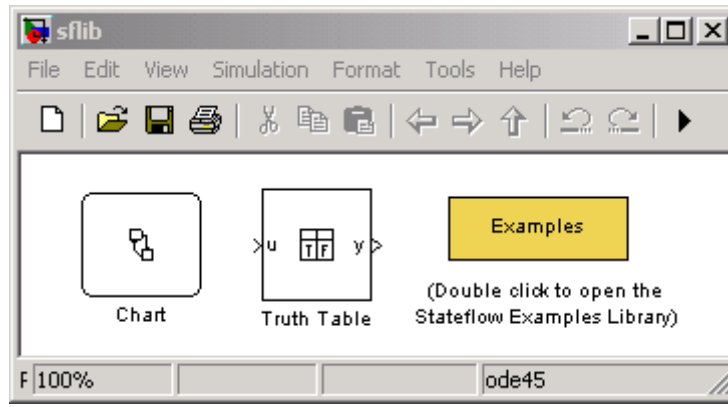
Open Stateflow library window

Syntax

```
sflib
```

Description

`sflib` opens the Stateflow library window:



From this window, you can drag Stateflow charts and truth tables into Simulink models, and access the Stateflow Examples Library.

See Also

`sfdebugger`, `sfexplr`, `sfhelp`

Purpose

Create Simulink® model containing empty Stateflow block

Syntax

```
Model_Handle = sfnew('-Chart_Type''Machine_Name')
```

Arguments

Model_Handle	Handle to the new Simulink model that will contain the Stateflow block.
Chart_Type	Type of Stateflow block to add to the Simulink model. Enter • `'-Classic'` for chart that implements full Stateflow semantics (default) • `'-Mealy'` for chart that implements Mealy state machine semantics • `'-Moore'` for chart that implements Moore state machine semantics • `'-TT'` for truth table Optional.
'Machine_Name'	Name of the Stateflow machine (also becomes the model name). Optional.

Description

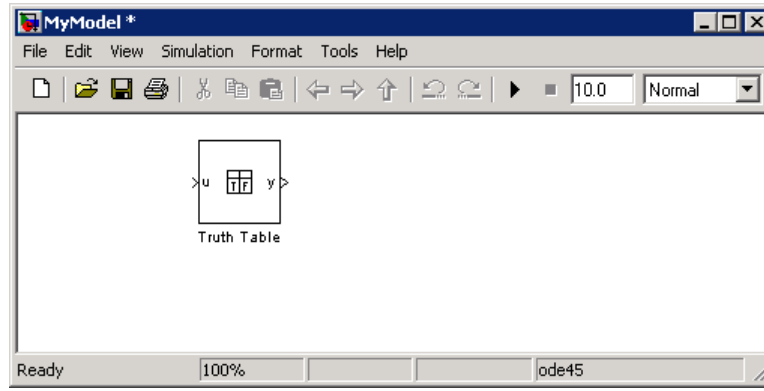
Model_Handle = sfnew('-*Chart_Type*''*Machine_Name*') returns the handle to a new model named **Machine_Name** that contains an empty Stateflow block of type *Chart_Type*, and opens the new model on your desktop. If *Chart_Type* is not specified, the default block is `Classic`. If *Machine_Name* is not specified, the default name is **untitled**.

Examples

Create a Simulink model called **MyModel** that contains an empty Stateflow truth table.

```
m = sfnew('-TT', 'MyModel')
```

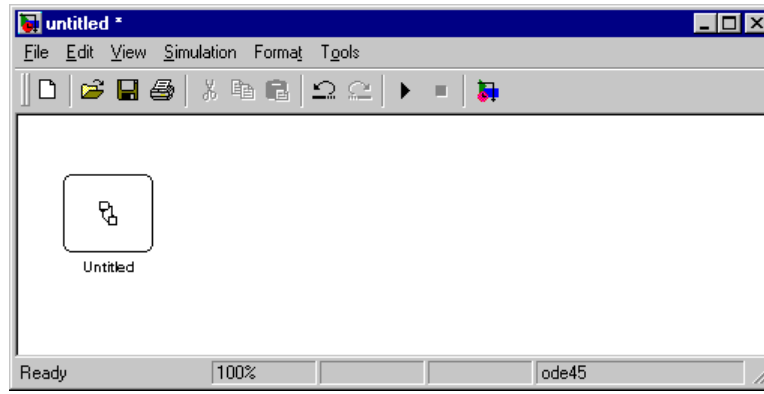
The new model looks like this:



Create an untitled Simulink model that contains an empty Stateflow chart.

```
m = sfnew
```

The new model looks like this:



See Also

`sfhelp`, `sfprint`, `sfroot`, `sfsave`, `stateflow`

Purpose

Open Stateflow machine

Syntax

```
sfopen
```

Description

`sfopen` prompts you for an `.mdl` file and opens the one that you select from your file system.

See Also

`sfclose`, `sfdebugger`, `sfexplr`, `sflib`, `sfnew`, `stateflow`

sfprint

Purpose

Print graphical view of Stateflow charts

Syntax

```
sfprint
sfprint( objects, format, outputOption, printEntireChart )
```

Arguments

objects — Any of the following object identifiers:

- String name of a Stateflow chart, or Simulink model, system, or block
- Handle to a Stateflow chart, or Simulink model, system, or block
- Cell array of names of and/or handles to a Stateflow chart, or Simulink model, system, or block
- Vector of handles to a Stateflow chart, or Simulink model, system, or block
- Simulink model construction commands `gcb`, `gcbh`, or `gcs`

format — Optional literal string that specifies the print destination:

- `'default'` prints to default printer
- `'ps'` generates PostScript file
- `'psc'` generates color PostScript file
- `'eps'` generates Encapsulated PostScript file
- `'epsc'` generates color Encapsulated PostScript file
- `'tif'` generates TIFF file
- `'jpg'` generates JPEG file
- `'png'` generates PNG file
- `'meta'` saves Stateflow image to clipboard as a metafile (Windows only)
- `'bitmap'` saves Stateflow image to clipboard as a bitmap (Windows only)

outputOption	Optional string that specifies an output file or printer: • String that specifies the name of a file to write to (file will be overwritten if more than one chart is printed) • `'promptForFile'` prompts for file name interactively • `'printer'` sends output to default printer (use only with `'default'`, `'ps'`, or `'eps'` formats) • `'file'` sends output to a default file, specified as *<path to object>.<device extension>* • `'clipboard'` copies output to clipboard
printEntireChart	Optional Boolean argument: • `1` (default) prints complete charts • `0` prints current view of charts

Description

`sfprint` prints the current Stateflow chart to a default printer.

`sfprint( objects, format, outputOption, printEntireChart )` prints all Stateflow charts identified in *objects* in the specified *format* to the file or printer specified in *outputOption*. Prints complete or current view of charts as specified in *printEntireChart*. If *format* argument is absent, the format defaults to `'ps'` and output is sent to the default printer. If *outputOption* argument is absent, the name of the Stateflow chart in the current directory is used as the output file name.

Examples

Print the complete chart whose handle is *id* to a TIFF file called **myFilename**.

```
sfprint(id, 'tif', 'myFilename')
```

Print all Stateflow charts in the current system as a PostScript file to the default printer.

```
sfprint(gcs)
```

Print the current Stateflow block to a JPEG file whose name is specified by the user interactively.

```
sfprint(gcb, 'jpg', 'promptForFile')
```

Print the current view of all Stateflow charts in the current system in PNG format using default file names.

```
sfprint(gcs, 'png', 'file', 0)
```

Assume that you loaded into MATLAB® a Simulink model named **myModel** that has two charts named **Chart1** and **Chart2**. Further, both **Chart1** and **Chart2** are represented by the Stateflow chart objects **ch1** and **ch2**, respectively.

Command	Result
`sfprint('myModel')`	Prints the graphical view of both **Chart1** and **Chart2** to the default printer.
`sfprint('myModel','ps')`	Prints the graphical view of both **Chart1** and **Chart2** to a PostScript file.
`sfprint(ch1.Id,'psc')`	Prints the graphical view of **Chart1** to a color PostScript file.
`sfprint([ch1.Id, ch2.Id])`	Prints the graphical views of both **Chart1** and **Chart2** to the default printer.

See Also

sfhelp, sfnew, sfsave, stateflow

Purpose

Get Stateflow root object

Syntax

```
object = sfroot
```

Description

`object = sfroot` returns the handle to the top-level object in the Stateflow machine hierarchy of objects. Use the root object to access all other objects in Stateflow charts, as described in "Access the Model Object" in the online Stateflow API Reference.

See Also

Stateflow functions `stateflow`, `sfnew`, `sfgco`, `sfclipboard`

sfsave

Purpose

Save Stateflow machine in current directory

Syntax

```
sfsave
sfsave( Model_Handle )
sfsave( Model_Handle, 'New_Model_Name' )
sfsave( Machine_Handle )
sfsave( 'Model_Name' )
sfsave( 'Defaults' )
```

Arguments

Model_Handle	Handle to a Simulink model that contains a Stateflow block.
'New_Model_Name'	Name to assign to the model being saved.
Machine_Handle	Handle to a Stateflow machine.
'Model_Name'	Name of a Simulink model that contains a Stateflow block.
'Defaults'	Literal string that directs Stateflow to save current settings as defaults.

Description

`sfsave` saves the current Stateflow machine in the current directory.

`sfsave( Model_Handle )` saves the Simulink model specified by *Model_Handle* in the current directory.

`sfsave( Model_Handle, 'New_Model_Name' )` saves Simulink model specified by *Model_Handle* as **New_Model_Name** in the current directory.

`sfsave( Machine_Handle )` saves the Simulink model that contains the Stateflow machine specified by `Machine_Handle` in the current directory.

`sfsave( 'Model_Name' )` saves the Simulink model called **Model_Name** in the current directory.

`sfsave( 'Defaults' )` saves the settings of the current Stateflow machine as defaults.

Examples

Save the model whose handle is `m` as **MyModel** in the current directory.

```
sfsave(m, 'MyModel')
```

Save the model that contains a Stateflow machine whose handle is `sf` in the current directory.

```
sfsave(sf)
```

See Also

`sfclose`, `sfnew`, `sfopen`, `sfprint`

Purpose

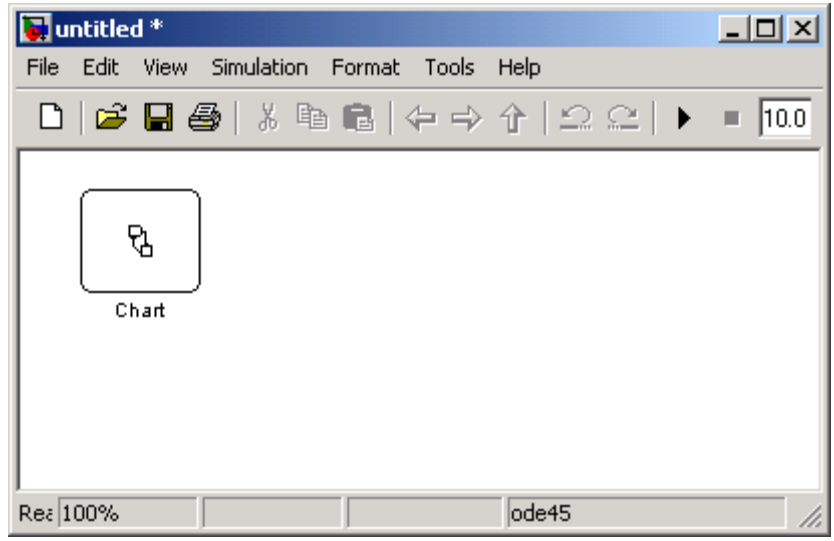
Create Simulink model containing empty Stateflow chart, and open Stateflow library window

Syntax

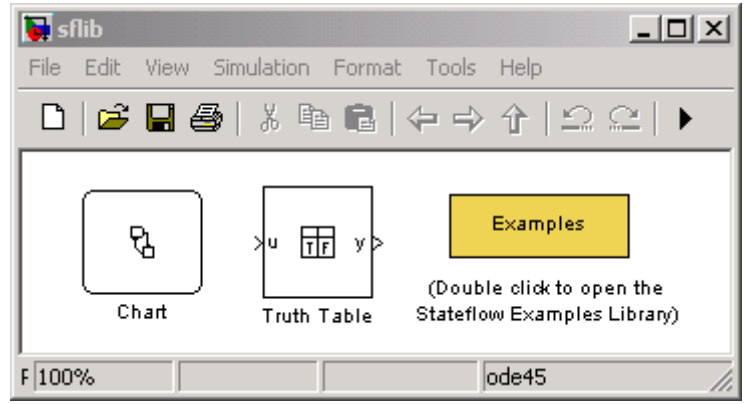
```
stateflow
```

Description

`stateflow` creates a new Simulink model that is preconfigured with an empty Stateflow chart:



The function also opens the Stateflow library window:



From this window, you can drag other Stateflow blocks into your Simulink model and access the Stateflow Examples Library.

See Also

`sflib`, `sfnew`, `sfroot`,

Index

F

S