

**Newt**

# Introduction

This is a PHP language extension for RedHat Newt library, a terminal-based window and widget library for writing applications with user friendly interface. Once this extension is enabled in PHP it will provide the use of Newt widgets, such as windows, buttons, checkboxes, radiobuttons, labels, editboxes, scrolls, textareas, scales, etc. Use of this extension is very similar to the original Newt API of C programming language.

# Installing/Configuring

## Requirements

This module uses the functions of the RedHat Newt library. You need libnewt version >= 0.51.0.

## Installation

This » [PECL](#) extension is not bundled with PHP. Information for installing this PECL extension may be found in the manual chapter titled [Installation of PECL extensions](#). Additional information such as new releases, downloads, source files, maintainer information, and a CHANGELOG, can be located here: » <http://pecl.php.net/package/newt>.

In PHP 4 this PECL extensions source can be found in the *ext/* directory within the PHP source or at the PECL link above. In order to use these functions you must compile CGI or CLI PHP with newt support by using the `--with-newt[=DIR]` configure option.

Note
<p>This extension is not available for Windows platform.</p> <p>You may need also <i>curses</i> and <i>slang</i> libraries, in order to compile this extension. To specify locations of these libraries, use the following configuration options: <code>--with-curses-dir=/path/to/libcurses --with-slang-dir=/path/to/libslang</code></p>

## Runtime Configuration

This extension has no configuration directives defined in *php.ini*.

## Resource Types

This extension uses two resource types: "newt component" and "newt grid".

Resource type "newt component" is returned by functions, which create common newt widgets (for example: [newt\\_button\(\)](#) )

Resource type "newt grid" is a special link identifier for components, returned by newt grid factory functions (for example: [newt\\_create\\_grid\(\)](#) )

# Predefined Constants

The constants below are defined by this extension, and will only be available when the extension has either been compiled into PHP or dynamically loaded at runtime.

## Newt form exit reasons

### Newt form exit reasons

constant	meaning
NEWT_EXIT_HOTKEY	hotkey defined by <a href="#">newt_form_add_hot_key()</a> was pressed
NEWT_EXIT_COMPONENT	some component has caused form to exit
NEWT_EXIT_FDREADY	file descriptor specified in <a href="#">newt_form_watch_fd()</a> is ready to be read or written to
NEWT_EXIT_TIMER	time specified in <a href="#">newt_form_set_timer()</a> has elapsed

## Newt colorsets

### Newt colorsets

constant	meaning
NEWT_COLORSET_ROOT	
NEWT_COLORSET_BORDER	
NEWT_COLORSET_WINDOW	
NEWT_COLORSET_SHADOW	
NEWT_COLORSET_TITLE	
NEWT_COLORSET_BUTTON	
NEWT_COLORSET_ACTBUTTON	
NEWT_COLORSET_CHECKBOX	

NEWT_COLORSET_ACTCHECKBOX	
NEWT_COLORSET_ENTRY	
NEWT_COLORSET_LABEL	
NEWT_COLORSET_LISTBOX	
NEWT_COLORSET_ACTLISTBOX	
NEWT_COLORSET_TEXTBOX	
NEWT_COLORSET_ACTTEXTBOX	
NEWT_COLORSET_HELPLINE	
NEWT_COLORSET_ROOTTEXT	
NEWT_COLORSET_ROOTTEXT	
NEWT_COLORSET_EMPTYSCALE	
NEWT_COLORSET_FULLSCALE	
NEWT_COLORSET_DISENTRY	
NEWT_COLORSET_COMPACTBUTTON	
NEWT_COLORSET_ACTSELLISTBOX	
NEWT_COLORSET_SELLISTBOX	

## Newt argument flags

### Newt argument flags

constant	meaning
NEWT_ARG_LAST	
NEWT_ARG_APPEND	

## Newt Flags Sense

### Newt Flags Sense

--	--

constant	meaning
NEWT_FLAGS_SET	
NEWT_FLAGS_RESET	
NEWT_FLAGS_TOGGLE	

## Newt Components Flags

### Newt Components Flags

constant	meaning
NEWT_FLAG_RETURNEXIT	Exit form, when component is activated
NEWT_FLAG_HIDDEN	Component is hidden
NEWT_FLAG_SCROLL	Component is scrollable
NEWT_FLAG_DISABLED	Component is disabled
NEWT_FLAG_BORDER	
NEWT_FLAG_WRAP	Wrap text
NEWT_FLAG_NOF12	Don't exit form on pressing F12
NEWT_FLAG_MULTIPLE	
NEWT_FLAG_SELECTED	Component is selected
NEWT_FLAG_CHECKBOX	Component is checkbox
NEWT_FLAG_PASSWORD	Entry component is password entry
NEWT_FLAG_SHOWCURSOR	Show cursor

## File Descriptor Flags

### File Descriptor Flags

constant	meaning
NEWT_FD_READ	

<b>NEWT_FD_WRITE</b>	
<b>NEWT_FD_EXCEPT</b>	

## Checkbox Tree Flags

### Checkbox Tree Flags

<b>constant</b>	<b>meaning</b>
<b>NEWT_CHECKBOXTREE_UNSELECTABLE</b>	
<b>NEWT_CHECKBOXTREE_HIDE_BOX</b>	
<b>NEWT_CHECKBOXTREE_COLLAPSED</b>	
<b>NEWT_CHECKBOXTREE_EXPANDED</b>	
<b>NEWT_CHECKBOXTREE_UNSELECTED</b>	
<b>NEWT_CHECKBOXTREE_SELECTED</b>	

## Entry Flags

### Entry Flags

<b>constant</b>	<b>meaning</b>
<b>NEWT_ENTRY_SCROLL</b>	
<b>NEWT_ENTRY_HIDDEN</b>	
<b>NEWT_ENTRY_RETURNEXIT</b>	
<b>NEWT_ENTRY_DISABLED</b>	

## Listbox Flags

### Listbox Flags

<b>constant</b>	<b>meaning</b>
-----------------	----------------

<b>NEWT_LISTBOX_RETURNEXIT</b>	
--------------------------------	--

## Textbox Flags

### Textbox Flags

constant	meaning
<b>NEWT_TEXTBOX_WRAP</b>	Wrap text in the textbox
<b>NEWT_TEXTBOX_SCROLL</b>	Scroll text in the textbox

## Form Flags

### Form Flags

constant	meaning
<b>NEWT_FORM_NOF12</b>	Don't exit form on F12 press

## Newt Keys

### Newt Keys

constant	meaning
<b>NEWT_KEY_TAB</b>	
<b>NEWT_KEY_ENTER</b>	
<b>NEWT_KEY_SUSPEND</b>	
<b>NEWT_KEY_ESCAPE</b>	
<b>NEWT_KEY_RETURN</b>	
<b>NEWT_KEY_EXTRA_BASE</b>	
<b>NEWT_KEY_UP</b>	
<b>NEWT_KEY_DOWN</b>	



<b>NEWT_KEY_LEFT</b>	
<b>NEWT_KEY_RIGHT</b>	
<b>NEWT_KEY_BKSPC</b>	
<b>NEWT_KEY_DELETE</b>	
<b>NEWT_KEY_HOME</b>	
<b>NEWT_KEY_END</b>	
<b>NEWT_KEY_UNTAB</b>	
<b>NEWT_KEY_PGUP</b>	
<b>NEWT_KEY_PGDN</b>	
<b>NEWT_KEY_INSERT</b>	
<b>NEWT_KEY_F1</b>	
<b>NEWT_KEY_F2</b>	
<b>NEWT_KEY_F3</b>	
<b>NEWT_KEY_F4</b>	
<b>NEWT_KEY_F5</b>	
<b>NEWT_KEY_F6</b>	
<b>NEWT_KEY_F7</b>	
<b>NEWT_KEY_F8</b>	
<b>NEWT_KEY_F9</b>	
<b>NEWT_KEY_F10</b>	
<b>NEWT_KEY_F11</b>	
<b>NEWT_KEY_F12</b>	
<b>NEWT_KEY_RESIZE</b>	

## Newt Anchors

## Newt Anchors

constant	meaning
NEWT_ANCHOR_LEFT	
NEWT_ANCHOR_RIGHT	
NEWT_ANCHOR_TOP	
NEWT_ANCHOR_BOTTOM	

## Grid Flags

### Grid Flags

constant	meaning
NEWT_GRID_FLAG_GROWX	
NEWT_GRID_FLAG_GROWY	
NEWT_GRID_EMPTY	
NEWT_GRID_COMPONENT	
NEWT_GRID_SUBGRID	

# Examples

## Basic usage

This example is a PHP port of RedHat 'setup' utility dialog, executed in text mode.

### Example #1 - Newt Usage Example

```
<?php
newt_init ();
newt_cls ();

newt_draw_root_text (0, 0, "Test Mode Setup Utility 1.12");
newt_push_help_line (null);
newt_draw_root_text (-30, 0, "(c) 1999-2002 RedHat, Inc");

newt_get_screen_size (&$rows, &$cols);

newt_open_window ($rows/2-17, $cols/2-10, 34, 17, "Choose a Tool");

$form = newt_form ();

$list = newt_listbox (3, 2, 10);

foreach (array (
    "Authentication configuration",
    "Firewall configuration",
    "Mouse configuration",
    "Network configuration",
    "Printer configuration",
    "System services") as $l_item)
{
    newt_listbox_add_entry ($list, $l_item, $l_item);
}

$b1 = newt_button (5, 12, "Run Tool");
$b2 = newt_button (21, 12, "Quit");

newt_form_add_component ($form, $list);
newt_form_add_components ($form, array($b1, $b2));

newt_refresh ();
newt_run_form ($form);

newt_pop_window ();
newt_pop_help_line ();
newt_finished ();
newt_form_destroy ($form);
?>
```

# Newt Functions

# newt\_bell

newt\_bell -- Send a beep to the terminal

## Description

**void newt\_bell** ( void )

This function sends a beep to the terminal.

Note
Depending on the terminal's settings, this beep may or may not be audible.

## Return Values

No value is returned.

# newt\_button\_bar

newt\_button\_bar -- This function returns a grid containing the buttons created.

## Description

resource **newt\_button\_bar** ( array &\$buttons )

This function returns a grid containing the buttons created.

## Parameters

*buttons*

## Return Values

Returns grid containing the buttons created.

# newt\_button

newt\_button -- Create a new button

## Description

resource **newt\_button** ( int \$left, int \$top, string \$text )

Creates a new button.

## Parameters

*left*  
X-coordinate of the button.

*top*  
Y-coordinate of the button.

*text*  
The text which should be displayed in the button.

## Return Values

Returns a resource link to the created button component, or **FALSE** on error.

## Examples

### Example #2 - A [newt\\_button\(\)](#) example

```
<?php

$form = newt_form();

$ok_button = newt_button(5, 12, "Run Tool");

newt_form_add_component($form, $ok_button);

?>
```

## See Also

- [newt\\_button\\_bar\(\)](#)

# newt\_centered\_window

newt\_centered\_window -- Open a centered window of the specified size

## Description

int **newt\_centered\_window** ( int \$width, int \$height [, string \$title ] )

Open a centered window of the specified size.

## Parameters

*width*

Window width

*height*

Window height

*title*

Window title

## Return Values

Undefined value.

## See Also

- [newt\\_pop\\_window\(\)](#)
- [newt\\_open\\_window\(\)](#)



# newt\_checkbox\_get\_value

newt\_checkbox\_get\_value -- Retreives value of checkox resource

## Description

string **newt\_checkbox\_get\_value** ( resource \$checkbox )

This function returns the character in the sequence which indicates the current value of the checkbox.

## Parameters

*checkbox*

## Return Values

Returns character indicating the value of the checkbox.

# newt\_checkbox\_set\_flags

newt\_checkbox\_set\_flags -- Configures checkbox resource

## Description

**void newt\_checkbox\_set\_flags** ( resource *\$checkbox*, int *\$flags*, int *\$sense* )

This function allows to set various flags on checkbox resource.

## Parameters

*checkbox*

*flags*

*sense*

## Return Values

No value is returned.

# newt\_checkbox\_set\_value

newt\_checkbox\_set\_value -- Sets the value of the checkbox

## Description

**void newt\_checkbox\_set\_value** ( resource *\$checkbox*, string *\$value* )

This function allows to set the current value of the checkbox resource.

## Parameters

*checkbox*

*value*

## Return Values

No value is returned.

# newt\_checkbox\_tree\_add\_item

newt\_checkbox\_tree\_add\_item -- Adds new item to the checkbox tree

## Description

```
void newt_checkbox_tree_add_item ( resource $checkboxtree, string $text, mixed $data, int $flags, int $index [, int $... ] )
```

This function allows to add new item to the checkbox tree.

## Parameters

*checkboxtree*

*text*

*data*

*flags*

*index*

## Return Values

No value is returned.

# newt\_checkbox\_tree\_find\_item

newt\_checkbox\_tree\_find\_item -- Finds an item in the checkbox tree

## Description

array **newt\_checkbox\_tree\_find\_item** ( resource \$checkboxtree, **mixed** \$data )

Finds an item in the checkbox tree by item's data.

## Parameters

*checkboxtree*

*data*

## Return Values

Returns checkbox tree item resource, or **NULL** if it wasn't found.

# newt\_checkbox\_tree\_get\_current

newt\_checkbox\_tree\_get\_current -- Returns checkbox tree selected item

## Description

**mixed** newt\_checkbox\_tree\_get\_current ( resource \$checkboxtree )

This method returns checkbox tree selected tem.

## Parameters

*checkboxtree*

## Return Values

Returns current (selected) checkbox tree item.

# newt\_checkbox\_tree\_get\_entry\_value

newt\_checkbox\_tree\_get\_entry\_value --

## Description

string **newt\_checkbox\_tree\_get\_entry\_value** ( resource \$checkboxtree, **mixed** \$data )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*checkboxtree*

*data*

## Return Values

# newt\_checkbox\_tree\_get\_multi\_selection

newt\_checkbox\_tree\_get\_multi\_selection --

## Description

array **newt\_checkbox\_tree\_get\_multi\_selection** ( resource \$checkboxtree, string \$  
seqnum )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*checkboxtree*

*seqnum*

## Return Values



# newt\_checkbox\_tree\_get\_selection

newt\_checkbox\_tree\_get\_selection --

## Description

array **newt\_checkbox\_tree\_get\_selection** ( resource \$checkboxtree )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*checkboxtree*

## Return Values

# newt\_checkbox\_tree\_multi

newt\_checkbox\_tree\_multi --

## Description

resource **newt\_checkbox\_tree\_multi** ( int \$left, int \$top, int \$height, string \$seq [, int \$flags ] )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*left*

*top*

*height*

*seq*

*flags*

## Return Values

# newt\_checkbox\_tree\_set\_current

newt\_checkbox\_tree\_set\_current --

## Description

**void** newt\_checkbox\_tree\_set\_current ( resource \$checkboxtree, **mixed** \$data )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*checkboxtree*

*data*

## Return Values

No value is returned.

# newt\_checkbox\_tree\_set\_entry\_value

newt\_checkbox\_tree\_set\_entry\_value --

## Description

**void** newt\_checkbox\_tree\_set\_entry\_value ( resource \$checkboxtree, **mixed** \$data, string \$value )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*checkboxtree*

*data*

*value*

## Return Values

No value is returned.

# newt\_checkbox\_tree\_set\_entry

newt\_checkbox\_tree\_set\_entry --

## Description

**void** newt\_checkbox\_tree\_set\_entry ( resource \$checkboxtree, **mixed** \$data, string \$text )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*checkboxtree*

*data*

*text*

## Return Values

No value is returned.

# newt\_checkbox\_tree\_set\_width

newt\_checkbox\_tree\_set\_width --

## Description

**void** newt\_checkbox\_tree\_set\_width ( resource \$checkbox\_tree, int \$width )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*checkbox\_tree*

*width*

## Return Values

No value is returned.

# newt\_checkbox\_tree

newt\_checkbox\_tree --

## Description

resource **newt\_checkbox\_tree** ( int \$left, int \$top, int \$height [, int \$flags ] )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

## Parameters

*left*

*top*

*height*

*flags*

## Return Values

# newt\_checkbox

newt\_checkbox --

## Description

resource **newt\_checkbox** ( int \$left, int \$top, string \$text, string \$def\_value [, string \$seq ] )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*left*

*top*

*text*

*def\_value*

*seq*

## Return Values



# newt\_clear\_key\_buffer

newt\_clear\_key\_buffer -- Discards the contents of the terminal's input buffer without waiting for additional input

## Description

`void newt_clear_key_buffer ( void )`

Discards the contents of the terminal's input buffer without waiting for additional input.

## Return Values

No value is returned.

## See Also

- [newt\\_wait\\_for\\_key\(\)](#)

# newt\_cls

newt\_cls --

## Description

`void newt_cls ( void )`

Warning
This function is currently not documented; only its argument list is available.

## Parameters

## Return Values

No value is returned.

# newt\_compact\_button

newt\_compact\_button --

## Description

resource **newt\_compact\_button** ( int *\$left*, int *\$top*, string *\$text* )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

## Parameters

*left*

*top*

*text*

## Return Values

# newt\_component\_add\_callback

newt\_component\_add\_callback --

## Description

**void** newt\_component\_add\_callback ( resource \$component, **mixed** \$func\_name, **mixed** \$data )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*component*

*func\_name*

*data*

## Return Values

No value is returned.

# newt\_component\_takes\_focus

newt\_component\_takes\_focus --

## Description

**void newt\_component\_takes\_focus** ( resource \$component, bool \$takes\_focus )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

## Parameters

*component*

*takes\_focus*

## Return Values

No value is returned.

# newt\_create\_grid

newt\_create\_grid --

## Description

resource **newt\_create\_grid** ( int \$cols, int \$rows )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*cols*

*rows*

## Return Values

# newt\_cursor\_off

newt\_cursor\_off --

## Description

void newt\_cursor\_off ( void )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

## Return Values

No value is returned.

# newt\_cursor\_on

newt\_cursor\_on --

## Description

`void newt_cursor_on ( void )`

Warning
This function is currently not documented; only its argument list is available.

## Return Values

No value is returned.



# newt\_delay

newt\_delay --

## Description

**void newt\_delay** ( int *\$microseconds* )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*microseconds*

## Return Values

No value is returned.

# newt\_draw\_form

newt\_draw\_form --

## Description

**void** newt\_draw\_form ( resource *\$form* )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*form*

## Return Values

No value is returned.

# newt\_draw\_root\_text

newt\_draw\_root\_text -- Displays the string text at the position indicated

## Description

**void newt\_draw\_root\_text** ( int \$left, int \$top, string \$text )

Displays the string text at the position indicated.

## Parameters

*left*

Column number

Note
If left is negative, the position is measured from the opposite side of the screen.

*top*

Line number

Note
If top is negative, the position is measured from the opposite side of the screen.

*text*

Text to display.

## Return Values

No value is returned.

## Examples

Example #3 - A <a href="#">newt_draw_root_text()</a> example
This code demonstrates drawing of titles in the both corners of the screen.  <pre>&lt;?php newt_init();</pre>

```
newt_cls();

newt_draw_root_text (2, 0, "Some root text");
newt_refresh();
sleep(1);

newt_draw_root_text (-30, 0, "Root text in the other corner");
newt_refresh();
sleep(1);

newt_finished();
?>
```

## See Also

- [newt\\_push\\_help\\_line\(\)](#)
- [newt\\_pop\\_help\\_line\(\)](#)

# newt\_entry\_get\_value

newt\_entry\_get\_value --

## Description

string **newt\_entry\_get\_value** ( resource *\$entry* )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*entry*

## Return Values

# newt\_entry\_set\_filter

newt\_entry\_set\_filter --

## Description

**void** newt\_entry\_set\_filter ( resource \$entry, **callback** \$filter, **mixed** \$data )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*entry*

*filter*

*data*

## Return Values

No value is returned.

# newt\_entry\_set\_flags

newt\_entry\_set\_flags --

## Description

**void newt\_entry\_set\_flags** ( resource \$entry, int \$flags, int \$sense )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*entry*

*flags*

*sense*

## Return Values

No value is returned.

# newt\_entry\_set

newt\_entry\_set --

## Description

**void** newt\_entry\_set ( resource \$entry, string \$value [, bool \$cursor\_at\_end ] )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

## Parameters

*entry*

*value*

*cursor\_at\_end*

## Return Values

No value is returned.



# newt\_entry

newt\_entry --

## Description

resource **newt\_entry** ( int \$left, int \$top, int \$width [, string \$init\_value [, int \$flags ] ] )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

## Parameters

*left*

*top*

*width*

*init\_value*

*flags*

## Return Values

# newt\_finished

newt\_finished -- Uninitializes newt interface

## Description

int **newt\_finished** ( void )

Uninitializes newt interface. This function be called, when program is ready to exit.

## Return Values

Returns 1 on success, 0 on failure.

## See Also

- [newt\\_init\(\)](#)

# newt\_form\_add\_component

newt\_form\_add\_component -- Adds a single component to the form

## Description

**void newt\_form\_add\_component** ( resource *\$form*, resource *\$component* )

Adds a single component to the *form*.

## Parameters

*form*

Form to which component will be added

*component*

Component to add to the form

## Return Values

No value is returned.

## Examples

### Example #4 - A [newt\\_form\\_add\\_component\(\)](#) example

```
<?php
$form = newt_form();

$options = array("Authentication configuration", "Firewall configuration",
"Mouse configuration", "Network configuration", "Printer configuration",
"System services");

$list = newt_listbox(3, 2, 10);

foreach ($options as $l_item) {
    newt_listbox_add_entry($list, $l_item, $l_item);
}

newt_form_add_component($form, $list);
?>
```

## See Also

- [newt\\_form\\_add\\_components\(\)](#)

# newt\_form\_add\_components

newt\_form\_add\_components -- Add several components to the form

## Description

**void newt\_form\_add\_components** ( resource *\$form*, array *\$components* )

Adds several components to the *form*.

## Parameters

*form*

Form to which components will be added

*components*

Array of components to add to the form

## Return Values

No value is returned.

## Examples

### Example #5 - A [newt\\_form\\_add\\_components\(\)](#) example

```
<?php
$form = newt_form();

$b1 = newt_button(5, 12, "Run Tool");
$b2 = newt_button(21, 12, "Quit");

newt_form_add_components($form, array($b1, $b2));
?>
```

## See Also

- [newt\\_form\\_add\\_component\(\)](#)

# newt\_form\_add\_hot\_key

newt\_form\_add\_hot\_key --

## Description

**void** newt\_form\_add\_hot\_key ( resource \$form, int \$key )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*form*

*key*

## Return Values

No value is returned.

# newt\_form\_destroy

newt\_form\_destroy -- Destroys a form

## Description

**void newt\_form\_destroy** ( resource `$form` )

This function frees the memory resources used by the form and all of the components which have been added to the form (including those components which are on subforms). Once a form has been destroyed, none of the form's components can be used.

## Parameters

*form*

Form component, which is going to be destroyed

## Return Values

No value is returned.

## See Also

- [newt\\_form\\_run\(\)](#)
- [newt\\_run\\_form\(\)](#)

# newt\_form\_get\_current

newt\_form\_get\_current --

## Description

resource **newt\_form\_get\_current** ( resource \$form )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

## Parameters

*form*

## Return Values



# newt\_form\_run

newt\_form\_run -- Runs a form

## Description

**void newt\_form\_run** ( resource \$form, array &\$exit\_struct )

This function runs the form passed to it.

## Parameters

*form*

Form component

*exit\_struct*

Array, used for returning information after running the form component. Keys and values are described in the following table:

### Form Exit Structure

Index Key	Value Type	Description
reason	integer	The reason, why the form has been exited. Possible values are defined <a href="#">here</a> .
watch	resource	Resource link, specified in <a href="#">newt_form_watch_fd()</a>
key	integer	Hotkey
component	resource	Component, which caused the form to exit

## Return Values

No value is returned.

## See Also

- [newt\\_run\\_form\(\)](#)

# newt\_form\_set\_background

newt\_form\_set\_background --

## Description

**void** newt\_form\_set\_background ( resource \$from, int \$background )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*from*

*background*

## Return Values

No value is returned.

# newt\_form\_set\_height

newt\_form\_set\_height --

## Description

**void newt\_form\_set\_height** ( resource *\$form*, int *\$height* )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*form*

*height*

## Return Values

No value is returned.

# newt\_form\_set\_size

newt\_form\_set\_size --

## Description

`void newt_form_set_size ( resource $form )`

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*form*

## Return Values

No value is returned.

# newt\_form\_set\_timer

newt\_form\_set\_timer --

## Description

**void newt\_form\_set\_timer** ( resource *\$form*, int *\$milliseconds* )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

## Parameters

*form*

*milliseconds*

## Return Values

No value is returned.

# newt\_form\_set\_width

newt\_form\_set\_width --

## Description

**void newt\_form\_set\_width** ( resource \$form, int \$width )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*form*

*width*

## Return Values

No value is returned.

# newt\_form\_watch\_fd

newt\_form\_watch\_fd --

## Description

**void** newt\_form\_watch\_fd ( resource \$form, resource \$stream [, int \$flags ] )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*form*

*stream*

*flags*

## Return Values

No value is returned.

# newt\_form

newt\_form -- Create a form

## Description

resource **newt\_form** ( [ resource \$vert\_bar [, string \$help [, int \$flags ] ] ] )

Create a new form.

## Parameters

*vert\_bar*

Vertical scrollbar which should be associated with the form

*help*

Help text string

*flags*

Various flags

## Return Values

Returns a resource link to the created form component, or **FALSE** on error.

## Examples

### Example #6 - A [newt\\_form\(\)](#) example

Displays a single button "Quit", which closes the application once it's pressed.

```
<?php
newt_init();
newt_cls();

$myform = newt_form();
$button = newt_button (5, 12, "Quit");

newt_form_add_component ($myform, $button);
newt_refresh ();
newt_run_form ($myform);

newt_finished ();
newt_form_destroy ($myform);
?>
```



## See Also

- [newt\\_form\\_run\(\)](#)
- [newt\\_run\\_form\(\)](#)
- [newt\\_form\\_add\\_component\(\)](#)
- [newt\\_form\\_add\\_components\(\)](#)
- [newt\\_form\\_destroy\(\)](#)

# newt\_get\_screen\_size

newt\_get\_screen\_size -- Fills in the passed references with the current size of the terminal

## Description

**void newt\_get\_screen\_size** ( int &\$cols, int &\$rows )

Fills in the passed references with the current size of the terminal.

## Parameters

*cols*

Number of columns in the terminal

*rows*

Number of rows in the terminal

## Return Values

No value is returned.

## Examples

### Example #7 - A [newt\\_get\\_screen\\_size\(\)](#) example

This code prints out the screen size of your terminal.

```
<?php
newt_init();
newt_get_screen_size (&$cols, &$rows);
newt_finished();

print "Your terminal size is: {$cols}x{$rows}\n";
?>
```

The above example will output:

```
Your terminal size is: 138x47
```

# newt\_grid\_add\_components\_to\_form

newt\_grid\_add\_components\_to\_form --

## Description

`void newt_grid_add_components_to_form ( resource $grid, resource $form, bool $recurse )`

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*grid*

*form*

*recurse*

## Return Values

No value is returned.

# newt\_grid\_basic\_window

newt\_grid\_basic\_window --

## Description

resource **newt\_grid\_basic\_window** ( resource \$text, resource \$middle, resource \$buttons )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*text*

*middle*

*buttons*

## Return Values

# newt\_grid\_free

newt\_grid\_free --

## Description

**void newt\_grid\_free** ( resource \$grid, bool \$recurse )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*grid*

*recurse*

## Return Values

No value is returned.

# newt\_grid\_get\_size

newt\_grid\_get\_size --

## Description

**void** newt\_grid\_get\_size ( **resource** \$grid, int &\$width, int &\$height )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*grid*

*width*

*height*

## Return Values

No value is returned.

# newt\_grid\_h\_close\_stacked

newt\_grid\_h\_close\_stacked --

## Description

resource **newt\_grid\_h\_close\_stacked** ( int \$element1\_type, resource \$element1 [, int \$...  
[, resource \$... ]])

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*element1\_type*

*element1*

## Return Values

# newt\_grid\_h\_stacked

newt\_grid\_h\_stacked --

## Description

resource **newt\_grid\_h\_stacked** ( int \$element1\_type, resource \$element1 [, int \$... [, resource \$... ]])

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*element1\_type*

*element1*

## Return Values



# newt\_grid\_place

newt\_grid\_place --

## Description

**void newt\_grid\_place** ( resource \$grid, int \$left, int \$top )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*grid*

*left*

*top*

## Return Values

No value is returned.

# newt\_grid\_set\_field

newt\_grid\_set\_field --

## Description

**void newt\_grid\_set\_field** ( resource \$grid, int \$col, int \$row, int \$type, resource \$val, int \$pad\_left, int \$pad\_top, int \$pad\_right, int \$pad\_bottom, int \$anchor [, int \$flags ] )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*grid*

*col*

*row*

*type*

*val*

*pad\_left*

*pad\_top*

*pad\_right*

*pad\_bottom*

*anchor*

*flags*

## **Return Values**

No value is returned.

# newt\_grid\_simple\_window

newt\_grid\_simple\_window --

## Description

resource **newt\_grid\_simple\_window** ( resource \$text, resource \$middle, resource \$buttons )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*text*

*middle*

*buttons*

## Return Values

# newt\_grid\_v\_close\_stacked

newt\_grid\_v\_close\_stacked --

## Description

resource **newt\_grid\_v\_close\_stacked** ( int \$element1\_type, resource \$element1 [, int \$... [, resource \$... ]])

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*element1\_type*

*element1*

## Return Values

# newt\_grid\_v\_stacked

newt\_grid\_v\_stacked --

## Description

resource **newt\_grid\_v\_stacked** ( int \$element1\_type, resource \$element1 [, int \$... [, resource \$... ]])

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*element1\_type*

*element1*

## Return Values

# newt\_grid\_wrapped\_window\_at

newt\_grid\_wrapped\_window\_at --

## Description

**void** newt\_grid\_wrapped\_window\_at ( resource \$grid, string \$title, int \$left, int \$top )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*grid*

*title*

*left*

*top*

## Return Values

No value is returned.

# newt\_grid\_wrapped\_window

newt\_grid\_wrapped\_window --

## Description

**void** newt\_grid\_wrapped\_window ( resource \$grid, string \$title )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*grid*

*title*

## Return Values

No value is returned.



# newt\_init

newt\_init -- Initialize newt

## Description

int **newt\_init** ( void )

Initializes the newt interface. This function must be called before any other newt function.

## Return Values

Returns 1 on success, 0 on failure.

## See Also

- [newt\\_finished\(\)](#)

# newt\_label\_set\_text

newt\_label\_set\_text --

## Description

**void newt\_label\_set\_text** ( resource *\$label*, string *\$text* )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*label*

*text*

## Return Values

No value is returned.

# newt\_label

newt\_label --

## Description

resource **newt\_label** ( int \$left, int \$top, string \$text )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*left*

*top*

*text*

## Return Values

# newt\_listbox\_append\_entry

newt\_listbox\_append\_entry --

## Description

`void newt_listbox_append_entry ( resource $listbox, string $text, mixed $data )`

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*listbox*

*text*

*data*

## Return Values

No value is returned.

# newt\_listbox\_clear\_selection

newt\_listbox\_clear\_selection --

## Description

**void newt\_listbox\_clear\_selection** ( resource \$listbox )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*listbox*

## Return Values

No value is returned.

# newt\_listbox\_clear

newt\_listbox\_clear --

## Description

`void newt_listbox_clear ( resource $listobx )`

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*listobx*

## Return Values

No value is returned.

# newt\_listbox\_delete\_entry

newt\_listbox\_delete\_entry --

## Description

`void newt_listbox_delete_entry ( resource $listbox, mixed $key )`

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*listbox*

*key*

## Return Values

No value is returned.

# newt\_listbox\_get\_current

newt\_listbox\_get\_current --

## Description

string **newt\_listbox\_get\_current** ( resource `$listbox` )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

## Parameters

*listbox*

## Return Values



# newt\_listbox\_get\_selection

newt\_listbox\_get\_selection --

## Description

array **newt\_listbox\_get\_selection** ( resource \$*listbox* )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*listbox*

## Return Values

# newt\_listbox\_insert\_entry

newt\_listbox\_insert\_entry --

## Description

`void newt_listbox_insert_entry ( resource $listbox, string $text, mixed $data, mixed $key )`

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*listbox*

*text*

*data*

*key*

## Return Values

No value is returned.

# newt\_listbox\_item\_count

newt\_listbox\_item\_count --

## Description

int newt\_listbox\_item\_count ( resource \$listbox )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*listbox*

## Return Values

# newt\_listbox\_select\_item

newt\_listbox\_select\_item --

## Description

`void newt_listbox_select_item ( resource $listbox, mixed $key, int $sense )`

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*listbox*

*key*

*sense*

## Return Values

No value is returned.

# newt\_listbox\_set\_current\_by\_key

newt\_listbox\_set\_current\_by\_key --

## Description

**void** newt\_listbox\_set\_current\_by\_key ( resource \$listbox, **mixed** \$key )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

## Parameters

*listbox*

*key*

## Return Values

No value is returned.

# newt\_listbox\_set\_current

newt\_listbox\_set\_current --

## Description

**void newt\_listbox\_set\_current** ( resource `$listbox`, int `$num` )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

## Parameters

*listbox*

*num*

## Return Values

No value is returned.

# newt\_listbox\_set\_data

newt\_listbox\_set\_data --

## Description

`void newt_listbox_set_data ( resource $listbox, int $num, mixed $data )`

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*listbox*

*num*

*data*

## Return Values

No value is returned.

# newt\_listbox\_set\_entry

newt\_listbox\_set\_entry --

## Description

**void** newt\_listbox\_set\_entry ( resource \$listbox, int \$num, string \$text )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

## Parameters

*listbox*

*num*

*text*

## Return Values

No value is returned.



# newt\_listbox\_set\_width

newt\_listbox\_set\_width --

## Description

**void newt\_listbox\_set\_width** ( resource \$listbox, int \$width )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*listbox*

*width*

## Return Values

No value is returned.

# newt\_listbox

newt\_listbox --

## Description

resource **newt\_listbox** ( int \$left, int \$top, int \$height [, int \$flags ] )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*left*

*top*

*height*

*flags*

## Return Values

# newt\_listitem\_get\_data

newt\_listitem\_get\_data --

## Description

**mixed** newt\_listitem\_get\_data ( resource \$item )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*item*

## Return Values

# newt\_listitem\_set

newt\_listitem\_set --

## Description

`void newt_listitem_set ( resource $item, string $text )`

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*item*

*text*

## Return Values

No value is returned.

# newt\_listitem

newt\_listitem --

## Description

resource **newt\_listitem** ( int \$left, int \$top, string \$text, bool \$is\_default, resouce \$prev\_item, mixed \$data [, int \$flags ] )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*left*

*top*

*text*

*is\_default*

*prev\_item*

*data*

*flags*

## Return Values

# newt\_open\_window

newt\_open\_window -- Open a window of the specified size and position

## Description

int **newt\_open\_window** ( int \$left, int \$top, int \$width, int \$height [, string \$title ] )

Open a window of the specified size and position.

## Parameters

*left*

Location of the upper left-hand corner of the window (column number)

*top*

Location of the upper left-hand corner of the window (row number)

*width*

Window width

*height*

Window height

*title*

Window title

## Return Values

Returns 1 on success, 0 on failure.

## See Also

- [newt\\_pop\\_window\(\)](#)
- [newt\\_centered\\_window\(\)](#)

# newt\_pop\_help\_line

newt\_pop\_help\_line -- Replaces the current help line with the one from the stack

## Description

**void newt\_pop\_help\_line ( void )**

Replaces the current help line with the one from the stack.

Note
It's important not to call to <a href="#">newt_pop_help_line()</a> more than <a href="#">newt_push_help_line()</a> .

## Return Values

No value is returned.

## See Also

- [newt\\_push\\_help\\_line\(\)](#)

# newt\_pop\_window

newt\_pop\_window -- Removes the top window from the display

## Description

**void newt\_pop\_window ( void )**

Removes the top window from the display, and redraws the display areas which the window overwrote.

## Return Values

No value is returned.

## See Also

- [newt\\_open\\_window\(\)](#)
- [newt\\_centered\\_window\(\)](#)



# newt\_push\_help\_line

newt\_push\_help\_line -- Saves the current help line on a stack, and displays the new line

## Description

`void newt_push_help_line ( [ string $text ] )`

Saves the current help line on a stack, and displays the new line.

## Parameters

*text*

New help text message

Note
If not specified, the help line is cleared.

## Return Values

No value is returned.

## See Also

- [newt\\_pop\\_help\\_line\(\)](#)

# newt\_radio\_get\_current

newt\_radio\_get\_current --

## Description

resource **newt\_radio\_get\_current** ( resource *\$set\_member* )

<b>Warning</b>
This function is currently not documented; only its argument list is available.

## Parameters

*set\_member*

## Return Values

# newt\_radiobutton

newt\_radiobutton --

## Description

resource **newt\_radiobutton** ( int \$left, int \$top, string \$text, bool \$is\_default [, resource \$prev\_button ] )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*left*

*top*

*text*

*is\_default*

*prev\_button*

## Return Values

# newt\_redraw\_help\_line

newt\_redraw\_help\_line --

## Description

`void newt_redraw_help_line ( void )`

Warning
This function is currently not documented; only its argument list is available.

## Parameters

## Return Values

No value is returned.

# newt\_reflow\_text

newt\_reflow\_text --

## Description

string **newt\_reflow\_text** ( string \$text, int \$width, int \$flex\_down, int \$flex\_up, int &\$actual\_width, int &\$actual\_height )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*text*

*width*

*flex\_down*

*flex\_up*

*actual\_width*

*actual\_height*

## Return Values

# newt\_refresh

newt\_refresh -- Updates modified portions of the screen

## Description

`void newt_refresh ( void )`

To increase performance, newt only updates the display when it needs to, not when the program tells it to write to the terminal. Applications can force newt to immediately update modified portions of the screen by calling this function.

## Return Values

No value is returned.

# newt\_resize\_screen

newt\_resize\_screen --

## Description

`void newt_resize_screen ( [ bool $redraw ] )`

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*redraw*

## Return Values

No value is returned.

# newt\_resume

newt\_resume -- Resume using the newt interface after calling [newt\\_suspend\(\)](#)

## Description

`void newt_resume ( void )`

Resume using the newt interface after calling [newt\\_suspend\(\)](#).

## Return Values

No value is returned.

## See Also

- [newt\\_suspend\(\)](#)



# newt\_run\_form

newt\_run\_form -- Runs a form

## Description

resource **newt\_run\_form** ( resource *\$form* )

This function runs the form passed to it.

## Parameters

*form*

Form component

## Return Values

The component which caused the form to stop running.

Note
Notice that this function doesn't fit in with newt's normal naming convention. It is an older interface which will not work for all forms. It was left in newt only for legacy applications. It is a simpler interface than the new <a href="#">newt_form_run()</a> though, and is still used quite often as a result. When an application is done with a form, it destroys the form and all of the components the form contains.

## See Also

- [newt\\_form\\_run\(\)](#)
- [newt\\_form\\_destroy\(\)](#)

# newt\_scale\_set

newt\_scale\_set --

## Description

`void newt_scale_set ( resource $scale, int $amount )`

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*scale*

*amount*

## Return Values

No value is returned.

# newt\_scale

newt\_scale --

## Description

resource **newt\_scale** ( int \$left, int \$top, int \$width, int \$full\_value )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*left*

*top*

*width*

*full\_value*

## Return Values

# newt\_scrollbar\_set

newt\_scrollbar\_set --

## Description

`void newt_scrollbar_set ( resource $scrollbar, int $where, int $total )`

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*scrollbar*

*where*

*total*

## Return Values

No value is returned.

# newt\_set\_help\_callback

newt\_set\_help\_callback --

## Description

`void newt_set_help_callback ( mixed $function )`

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*function*

## Return Values

No value is returned.

# newt\_set\_suspend\_callback

`newt_set_suspend_callback` -- Set a callback function which gets invoked when user presses the suspend key

## Description

`void newt_set_suspend_callback ( callback $function, mixed $data )`

Set a callback function which gets invoked when user presses the suspend key (normally ^Z). If no suspend callback is registered, the suspend keystroke is ignored.

## Parameters

*function*

A callback function, which accepts one argument: data

*data*

This data is been passed to the callback function

## See Also

- [`newt\_suspend\(\)`](#)
- [`newt\_resume\(\)`](#)

# newt\_suspend

newt\_suspend -- Tells newt to return the terminal to its initial state

## Description

`void newt_suspend ( void )`

Tells newt to return the terminal to its initial state. Once this is done, the application can suspend itself (by sending itself a SIGTSTP, fork a child program, or do whatever else it likes).

## Return Values

No value is returned.

## See Also

- [newt\\_resume\(\)](#)

# newt\_textbox\_get\_num\_lines

newt\_textbox\_get\_num\_lines --

## Description

int **newt\_textbox\_get\_num\_lines** ( resource \$textbox )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*textbox*

## Return Values



# newt\_textbox\_reflowed

newt\_textbox\_reflowed --

## Description

resource **newt\_textbox\_reflowed** ( int \$left, int \$top, char \$\*text, int \$width, int \$flex\_down, int \$flex\_up [, int \$flags ] )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*left*

*top*

*\*text*

*width*

*flex\_down*

*flex\_up*

*flags*

## Return Values

# newt\_textbox\_set\_height

newt\_textbox\_set\_height --

## Description

**void** newt\_textbox\_set\_height ( resource \$textbox, int \$height )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*textbox*

*height*

## Return Values

No value is returned.

# newt\_textbox\_set\_text

newt\_textbox\_set\_text --

## Description

**void** newt\_textbox\_set\_text ( resource \$textbox, string \$text )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*textbox*

*text*

## Return Values

No value is returned.

# newt\_textbox

newt\_textbox --

## Description

resource **newt\_textbox** ( int \$left, int \$top, int \$width, int \$height [, int \$flags ] )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*left*

*top*

*width*

*height*

*flags*

## Return Values

# newt\_vertical\_scrollbar

newt\_vertical\_scrollbar --

## Description

resource **newt\_vertical\_scrollbar** ( int \$left, int \$top, int \$height [, int \$normal\_colorset [, int \$thumb\_colorset ] ] )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*left*

*top*

*height*

*normal\_colorset*

*thumb\_colorset*

## Return Values

# newt\_wait\_for\_key

newt\_wait\_for\_key -- Doesn't return until a key has been pressed

## Description

`void newt_wait_for_key ( void )`

This function doesn't return until a key has been pressed. The keystroke is then ignored. If a key is already in the terminal's buffer, this function discards a keystroke and returns immediately.

## Return Values

No value is returned.

## See Also

- [newt\\_clear\\_key\\_buffer\(\)](#)

# newt\_win\_choice

newt\_win\_choice --

## Description

```
int newt_win_choice ( string $title, string $button1_text, string $button2_text, string $
format [, mixed $args [, mixed $... ] ] )
```

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*title*

*button1\_text*

*button2\_text*

*format*

*args*

## Return Values

# newt\_win\_entries

newt\_win\_entries --

## Description

int **newt\_win\_entries** ( string \$title, string \$text, int \$suggested\_width, int \$flex\_down, int \$flex\_up, int \$data\_width, array &\$items, string \$button1 [, string \$... ] )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*title*

*text*

*suggested\_width*

*flex\_down*

*flex\_up*

*data\_width*

*items*

*button1*

*button2*

## Return Values



## Examples

### Example #8 - A [newt\\_win\\_entries\(\)](#) example

```
<?php

newt_init();
newt_cls();

$entries[] = array('text' => 'First name:', 'value' => &$f_name);
$entries[] = array('text' => 'Last name:', 'value' => &$l_name);

$rc = newt_win_entries("User information", "Please enter your credentials:", 50,
7, 7, 30, $entries, "Ok", "Back");
newt_finished ();

if ($rc != 2) {
    echo "Your name is: $f_name $l_name\n";
}
?>
```

# newt\_win\_menu

newt\_win\_menu --

## Description

int newt\_win\_menu ( string \$title, string \$text, int \$suggestedWidth, int \$flexDown, int \$flexUp, int \$maxListHeight, array \$items, int &\$listItem [, string \$button1 [, string \$... ]])

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*title*

*text*

*suggestedWidth*

*flexDown*

*flexUp*

*maxListHeight*

*items*

*listItem*

*button1*

## Return Values

No value is returned.

# newt\_win\_message

newt\_win\_message --

## Description

**void newt\_win\_message** ( string \$title, string \$button\_text, string \$format [, **mixed** \$args [, **mixed** \$... ] ] )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*title*

*button\_text*

*format*

*args*

## Return Values

No value is returned.

# newt\_win\_messagev

newt\_win\_messagev --

## Description

**void newt\_win\_messagev** ( string \$title, string \$button\_text, string \$format, array \$args )

Warning
This function is currently not documented; only its argument list is available.

## Parameters

*title*

*button\_text*

*format*

*args*

## Return Values

No value is returned.

# newt\_win\_ternary

newt\_win\_ternary --

## Description

```
int newt_win_ternary ( string $title, string $button1_text, string $button2_text, string $
button3_text, string $format [, mixed $args [, mixed $... ] ] )
```

Warning
This function is currently not documented; only its argument list is available.

## Parameters

- title*  
Its description
- button1\_text*  
Its description
- button2\_text*  
Its description
- button3\_text*  
Its description
- format*  
Its description
- args*  
Its description

## Return Values

What the function returns, first on success, then on failure. See also the &return.success; entity