

in

COLLABORATORS

	TITLE : in		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		July 31, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	in	1
1.1	windowclass.guide	1
1.2	windowclass/--background--	1
1.3	windowclass/WM_CLOSE	1
1.4	windowclass/WM_GADGETKEY	2
1.5	windowclass/WM_HANDLEIDCMP	3
1.6	windowclass/WM_KEY[xxx]	3
1.7	windowclass/WM_OPEN	4
1.8	windowclass/WM_SLEEP	4
1.9	windowclass/WM_WAKEUP	5

Chapter 1

in

1.1 windowclass.guide

Search
TABLE OF CONTENTS

windowclass/--background--
windowclass/WM_CLOSE
windowclass/WM_GADGETKEY
windowclass/WM_HANDLEIDCMP
windowclass/WM_KEY[xxx]
windowclass/WM_OPEN
windowclass/WM_SLEEP
windowclass/WM_WAKEUP

1.2 windowclass/--background--

NAME

Class: windowclass
Superclass: ROOTCLASS
Include File: <libraries/bgui.h>

FUNCTION

To provide an easy to use BGUI interface to intuition windows. The window class provides just about everything you need to create windows with a fully font-sensitive and size-adjustable GUI.

Objects of this class send out the following attributes:

WINDOW_IsOpen - TRUE is window open, FALSE is closed.

1.3 windowclass/WM_CLOSE

NAME

WM_CLOSE -- Close the window.

SYNOPSIS

```
succ = DoMethod( obj, WM_CLOSE )
```

```
ULONG succ;
```

FUNCTION

This method will close the window. It is safe to call this method even when the window is not open.

INPUTS

RESULT

succ - TRUE upon success and FALSE upon failure.

SEE ALSO

WM_OPEN

1.4 windowclass/WM_GADGETKEY

NAME

WM_GADGETKEY -- Add a gadget hotkey.

SYNOPSIS

```
succ = DoMethod( obj, WM_GADGETKEY, req, target, key )
```

```
ULONG succ;  
struct Requester *req;  
Object *target;  
STRPTR key;
```

FUNCTION

With this method you can assign a key to trigger a gadget object in the window. The object which is connected to the key will then be controllable by the specified key.

INPUTS

req - This version of the library does not support BGUI gadget objects in requesters so this field must be set to NULL.

target - This must be a pointer to the object which the key will control.

key - This must point to a string in which a single character is located. The character is the key which controls the object when it is pressed.

RESULT

succ - TRUE upon success and FALSE upon failure.

1.5 windowclass/WM_HANDLEIDCMP

NAME

WM_HANDLEIDCMP -- Handle window events.

SYNOPSIS

```
id = DoMethod( obj, WM_HANDLEIDCMP )
```

```
ULONG id;
```

FUNCTION

This method must be used to call the windowclass event handler. The event handler will act upon the messages present at the window's message port and return you ID's on which to act.

INPUTS

RESULT

id - This can be any of the following:

WMHI_CLOSEWINDOW -- The window's close gadget was selected.
WMHI_NOMORE -- No more messages waiting.
WMHI_INACTIVE -- The window was de-activated.
WMHI_ACTIVE -- The window was activated.
WMHI_IGNORE -- Ignore this result.

Any return code which differs from the ones above is the ID of a selected object (gadget or menu).

Please look at the demo programs for more information.

1.6 windowclass/WM_KEY[xxx]

NAME

WM_KEYACTIVE, WM_KEYINPUT, WM_KEY_INACTIVE -- See "methods.doc"

FUNCTION

These three methods are sent to the gadget object during a key-session.

Please refer to the "methods.doc" file for more information.

1.7 windowclass/WM_OPEN

NAME

WM_OPEN -- Open the window.

SYNOPSIS

```
win = DoMethod( obj, WM_OPEN )

struct Window *win;
```

FUNCTION

This method will open up the window. If successful a pointer to the window structure is returned. It is safe to call this method when the window is already open.

INPUTS

RESULT

win - A pointer to the opened window upon success and NULL upon failure.

SEE ALSO

WM_CLOSE

1.8 windowclass/WM_SLEEP

NAME

WM_SLEEP -- Put the window to sleep.

SYNOPSIS

```
succ = DoMethod( obj, WM_SLEEP )

ULONG succ;
```

FUNCTION

With this method you can put your window to sleep when it is open. This is done by setting up a small invisible requester which will block the window input possibilities. Also a standard Workbench busy pointer is set in the window.

This call is nested. This means the window has to be released with WM_WAKEUP as much times as you have locked it with WM_SLEEP.

INPUTS

RESULT

succ - TRUE upon success and FALSE upon failure.

SEE ALSO

WM_WAKEUP

1.9 windowclass/WM_WAKEUP

NAME

WM_WAKEUP -- Wake up the window.

SYNOPSIS

```
succ = DoMethod( obj, WM_WAKEUP )
```

```
ULONG succ;
```

FUNCTION

This method must be used to wake up the window again after being put to sleep with the WM_SLEEP method. Please note that you must wake up the window as many times as you have put it to sleep before you can actually use it again.

INPUTS

RESULT

succ - TRUE upon success and FALSE upon failure.

SEE ALSO

WM_SLEEP
