TearOff

COLLABORATORS

	TITLE : TearOff		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 7, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

TearOff

1.1 main

```
                This class allows the MUI applications to create       Read more:
 special groups called TearOffPanels. They can be
 freely arranged by the user, dragged outside the
                How to use it
                window or hidden. Panel layout is automaticaly
 stored and reused next time. If you don't like
                Configuration
                standard TearOff frames and images, you can
 reconfigure the appearance with MUI Preferences
                Info for Developers
                program.
                (and the curious)
                It's also very simple to use for developers,
 just change some of your groups into TearOff panels!

 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 Szymon Ulatowski <szulat@friko6.onet.pl>
 WWW support: http://www.friko6.onet.pl/rz/szulat/tearoff/
```

1.2 howtouse

```
Anytime you can use the context menu (press and hold the right mouse button)
to hide, reveal or tear off the panel.
TearOff Bays (parent of TearOff Panel) show the list of associated panels
in the context menu. You can hide/reveal the panels using that menu.

Other actions you can perform on the panels:

 Fixed panels (displayed inside the window)
  - double click to hide
  - drag the panel vertically on the left border to move the whole row
  - drag the panel into the interior of another panel to insert it
    into the row
  - drag the panel to the top/bottom to split the row
  - drag outside the bay to tear off the panel
```

Hidden panels (can be displayed as icon/text buttons or invisible)
- click to reveal the panel

TornOff panels (in separate window)
- close the window (upper left corner) to hide the panel
- double click to send the panel back to its window
- drag into the bay to fix it to the window
  (if the application has linked some bays, you can fix the
  panel in any bay linked to the panel's original bay)

1.3 config

Hint: Try the predefined settings: "TearOff Original", "Netscape-like"
and "Microsoft-like".

Changes are visible after you choose 'Save', 'Use' or 'Test' button.
(Refer to MUI Documentation for further information)

There are 3 pages in the prefs program, for each of 3 possible panel state.

Fixed:

```
    +-------------panel-frame--------------+
    | +---+                                |
    | | i |                                |
    | | m |                                |
    | | a |\__ button frame                |
    | | g |                                |
    . . e .                                .
```

You can define a frame type used by the panel and the button, and the image for the button. There are 2 image selectors, one for horizontal and one for vertical panel layout. This is for images that don't change automatically their appearance when used in horizontal/vertical context (eg. bitmap images).

Hidden:

There is only one frame, because the hidden image has no panel, it is the single button.

As the extra feature you can enable 'Text label' if you want to see the name of the panel. This option changes the usual interpretation of the inner frame margins - they are used to adjust the text position inside the frame.

If you don't want to see the icons for the hidden panels, you can clear the 'Use icon' checkbox (hidden panels will be still available in the bay menu).

Window:

Frame/image gadgets work just like on the 'Fixed' page.

With the additional cycle gadget you can control the automatic 'To Front' feature. It can help you find your panels lost under other windows.
'On Activation' means the torn off panels are brought to front each

time the parent window is selected. ’Always’ does the same all the time.

If you want make your windows a bit smaller you can disable the ’Window Title’ option.

1.4 developers

1.Polymorphware
TearOff classes are Polymorphware. That means, if you use it in your program, you have to pay/send me what you expect from your users, eg. if it is ’Chocolateware’ you should send me a bar of chocolate. Freeware programs can use TearOff for free, of course.
More information and technical stuff is available in developer docs

2.Compatibility problems?
TearOff classes use some undocumented features of MUI and muigfx.library.
This can give one of 2 effects:

- the author of MUI will come to the conclusion that developers need those features and will officially reveal and document it (hahaha) (can you believe that frame selection gadget and frame drawing routine are ’secret’? in graphics user interface system?)
- there will be a new version of MUI (hahahahaha!!!!!) and TearOff will not work with it

As you can see, both possibilities are a science fiction.
Anyway, if you are afraid of the second one, you can sleep well.
TearOff checks each time if the MUI version is ’safe’ and will disable ’tricks’ when it is too new (in ’notricks’ mode, panels are drawn using plain frame and no images :-( ). Of course the new TearOff will consider the new MUI safe again.
But if you are brave, you can check if the new version is also safe.
Set the env. variable:

```
setenv tearoffsafemui 20 (if the new mui version is 20)
```

and test the tearoff panels. If they work ok, you can make the setting permanent: copy env:tearoffmuisafe envarc: