

Wie erzeuge ich Dialog-Widgets?

Inhalt:

Wie erzeuge ich eine einfache Meldung?

Wie kann ich vor dem Beenden eines Programms noch was abfragen?

Wie kann ich ein eigenes Bitmap benutzen?

Häufig muss man den Benutzer eines Programms vor einer Aktion noch etwas sagen oder man möchte dem Benutzer noch etwas mitteilen. So möchte man vor dem Schließen des Programms noch fragen, ob die Datei gespeichert werden soll oder der Benutzer hat eine falsche Eingabe gemacht und man möchte ihm das mitteilen und so weiter.

Anwendungsbeispiele für den Einsatz des Dialog-Widgets gibt es viele.

Wie erzeuge ich eine einfache Meldung?

Angenommen, der User hat eine ungültige E-Mail-Adresse eingegeben und wir möchten ihn darauf aufmerksam machen. Das geht ziemlich einfach:

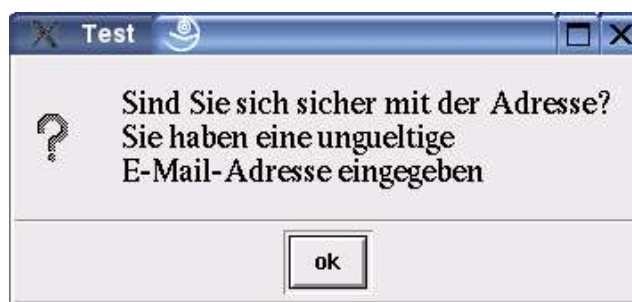
```
#!/usr/bin/perl

use strict;
use warnings;
use Tk;
use Tk::Dialog;

my $mw = MainWindow->new();
$mw->Label(-text => 'Dies ist ein Test')->pack();
my $text = qq~Sind Sie sich sicher mit der Adresse?
Sie haben eine ungueltige E-Mail-Adresse eingegeben~;
my $dialog = $mw->Dialog(-title           => 'Test',
                        -text             => $text,
                        -bitmap           => 'question',
                        -default_button   => 'ok',
                        -buttons          => ['ok']);

$dialog->Show();
MainLoop;
```

Das sieht dann so aus:



Wie kann ich vor dem Beenden eines Programms noch was abfragen?

Häufig beendet der Benutzer ein Programm, obwohl die letzten Änderungen noch nicht gespeichert sind. Wird das Beenden nicht abgefangen, werden die letzten Änderungen nicht gespeichert und der Benutzer ärgert sich dann das nächste Mal. Dieses Problem kann man durch folgenden Code umgehen

```
#!/usr/bin/perl

use warnings;
use strict;
use Tk;
use Tk::Dialog;

my $mw = MainWindow->new();
$mw->protocol('WM_DELETE_WINDOW', \&before_exit);
MainLoop;

sub before_exit {
    my $text = 'Save all your changes before exit?';
    my $dialog = $mw->Dialog(-text          => $text,
                             -bitmap       => 'question',
                             -title       => 'Test',
                             -default_button => 'ok',
                             -buttons     => ['cancel', 'ok'],
                             );

    my $b_exit = $dialog->Show();

    if ($b_exit eq 'ok'){
        # save all the changes
    }
    exit;
} # end before_exit
```

Der Dialog sieht dann so aus:



Wie kann ich ein eigenes Bitmap benutzen?

In den beiden oberen Dialogen ist links immer ein Fragezeichen zu sehen. Dies ist jedoch nicht immer angebracht. Bei Tk sind 8 Bitmaps "mitgeliefert".

- error
- question
- info
- hourglass
- warnings
- questhead
- gray50
- gray12

Möchte man ein eigenes Bitmap verwenden, so ist dies ohne größere Probleme möglich. Die Bitmaps müssen jedoch im X11 Bitmap format vorliegen (jedenfalls ist mir nicht bekannt, dass man auch andere Formate einbinden kann). Hat man jedoch ein gif-Bild zur Verfügung, so kann man dieses z.B. mit `convert infile.gif outfile.xbm` konvertieren.

Zum Einbinden des eigenen Bitmaps muss einfach ein @ vor die Pfadangabe gestellt werden.
Beispiel:

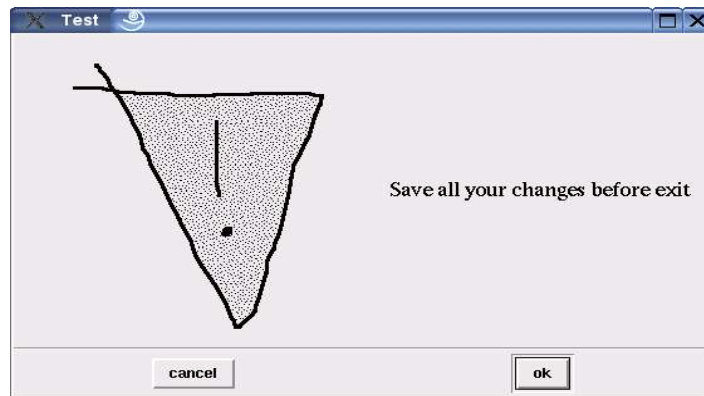
```
#!/usr/bin/perl

use strict;
use warnings;
use Tk;
use Tk::Dialog;

my $mw = MainWindow->new();
my $text = 'Save all your changes before exit?';
my $dialog = $mw->Dialog(-title      => 'Test',
                        -text        => $text,
                        -bitmap      => "\@./test.xbm",
                        -default_button => 'ok',
                        -buttons     => ['cancel', 'ok']);

$dialog->Show();
MainLoop;
```

Das sieht dann so aus:



Eine weitere (und schönere) Methode , um eine eigene Grafik zu verwenden: Mit Hilfe eines Photo-Objekts und mit `-image` statt `-bitmap`.

```
#!/usr/bin/perl

use strict;
use warnings;
use Tk;
use Tk::Dialog;

my $mw = MainWindow->new();
my $photo = $mw->Photo(-file => './test.bmp');
my $text = 'Save all your changes before exit?';
my $dialog = $mw->Dialog(-title      => 'Test',
                        -text        => $text,
                        -image       => $photo,
                        -default_button => 'ok',
                        -buttons     => ['cancel', 'ok']);

$dialog->Show();
MainLoop;
```

und so sieht das ganze dann aus:

