

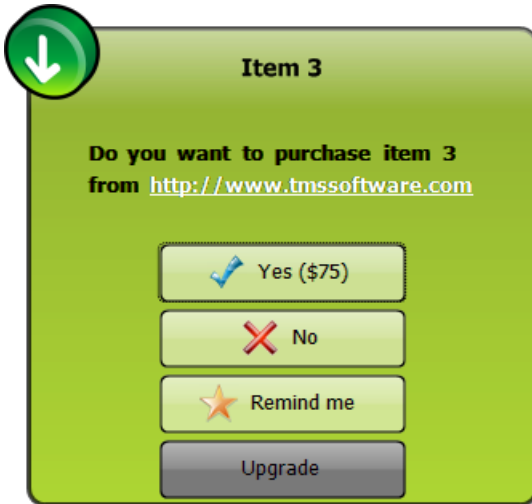
TMS Advanced Smooth Message Dialog DEVELOPERS GUIDE

July 2009
Copyright © 2009 by tmssoftware.com bvba
Web: <http://www.tmssoftware.com>
Email: info@tmssoftware.com

Index

TAdvSmoothMessageDialog.....	3
TAdvSmoothMessageDialog description.....	3
TAdvSmoothMessageDialog features	3
TAdvSmoothMessageDialog use	3

TAdvSmoothMessageDialog



TAdvSmoothMessageDialog description

Replace your default delphi message dialogs with the TAdvSmoothMessageDialog. This feature rich component allows you to create a message dialog with complex and smooth graphics.

TAdvSmoothMessageDialog features

- Full opacity control
- Buttons with separate button result
- HTML text support
- Office Styles
- Horizontal or vertical stacked buttons
- Buttons with customizable color, image
- Standard ShowMessage & MessageDlg call direct replacements
- Margins & capability to have alpha transparent images outside the dialog

TAdvSmoothMessageDialog use

To call the TAdvSmoothMessageDialog with a simple text use ShowSmoothMessage('text'); This allows you to quickly replace the ShowMessage functions with the TAdvSmoothMessageDialog component. By default the TAdvSmoothMessageDialog will apply the Office 2007 Luna style if you use ShowSmoothMessage.

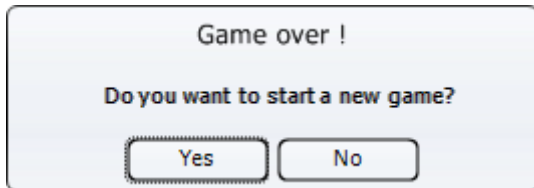
The TAdvSmoothMessageDialog will automatically calculate the best width and height depending of the number of buttons, the HTML and the caption. The buttons can have a vertical or horizontal layout.

Below is a sample of a message dialog with a caption, some html text and two buttons. With a Office 2007 Silver appearance.

Code:

```
var
  md: TAdvSmoothMessageDialog;
begin
  md := TAdvSmoothMessageDialog.Create(Self);
  md.Caption := 'Game over !';
  with md.Buttons.Add do
  begin
    Caption := 'Yes';
    ButtonResult := mrYes;
  end;
  with md.Buttons.Add do
  begin
    Caption := 'No';
    ButtonResult := mrNo;
  end;
  md.HTMLText.Text := 'Do you want to start a new game?';
  md.HTMLText.Location := hlCenterCenter;
  md.SetComponentStyle(tsOffice2007Silver);
  md.Execute;
```

Result:



Replacement Functions

The TAdvSmoothMessageDialog supports functions to quickly show a default message dialog. By doing a find & replace in existing applications of ShowMessage to ShowSmoothMessage and MessageDlg to SmoothMessageDlg, you can quickly change an application to use the new dialogs. Below are the functions signatures that can be used:

```
procedure ShowSmoothMessage(const Msg: string; AStyle: TTMSStyle =
tsOffice2007Luna);
```

```
function SmoothMessageDlg(const Title, Msg: string; DlgType: TMsgDlgType;
  Buttons: TMsgDlgButtons; HelpCtx: Longint; AStyle: TTMSStyle =
tsOffice2007Luna): Integer; overload;
```

```
function SmoothMessageDlg(const Title, Msg: string; DlgType: TMsgDlgType;
  Buttons: TMsgDlgButtons; HelpCtx: Longint; DefaultButton: TMsgDlgBtn;
AStyle: TTMSStyle = tsOffice2007Luna): Integer; overload;
```

```
function SmoothMessageDlgPos(const Title, Msg: string; DlgType:
TMsgDlgType;
  Buttons: TMsgDlgButtons; HelpCtx: Longint; X, Y: Integer; AStyle:
TTMSStyle = tsOffice2007Luna): Integer; overload;
```

```
function SmoothMessageDlgPos(const Title, Msg: string; DlgType:
TMsgDlgType;
  Buttons: TMsgDlgButtons; HelpCtx: Longint; X, Y: Integer;
  DefaultButton: TMsgDlgBtn; AStyle: TTMSStyle = tsOffice2007Luna):
Integer; overload;
```

Execute and ExecuteDialog

The TAdvSmoothMessageDialog has an Execute and ExecuteDialog function. The ExecuteDialog function returns a TModalResult. The Execute function returns a Boolean and this Boolean is only true when the ModalResult is mrOk or mrYes. When adding buttons and using the Execute function, make sure that there is a button with the ButtonResult mrOk or mrYes.

```
AdvSmoothMessageDialog1.Buttons.Clear;
AdvSmoothMessageDialog1.Position := poScreenCenter;
with AdvSmoothMessageDialog1.Buttons.Add do
begin
    ButtonResult := mrOk;
    Caption := 'Ok';
end;
with AdvSmoothMessageDialog1.Buttons.Add do
begin
    ButtonResult := mrCancel;
    Caption := 'Cancel';
end;

if AdvSmoothMessageDialog1.Execute then
    Label1.Caption := 'Ok Button clicked';
```

Appearance

When the default message dialog functions are no longer suitable, the message dialog can be fully customized for its 4 different elements:

- **Buttons:** The buttons have a separate appearance and can be formatted with a color, and opacity property. When a picture is needed to make the button more attractive use the picture and pictureposition property. The buttons can also be vertical or horizontal stacked depending on the type of message dialog needed.

The ButtonAreaFill property is used to fill the area of the buttons.

- **Caption:** The caption has a separate fill that is fully customizable. With the CaptionFont and CaptionPosition properties the caption text can be formatted.
- **HTML:** The HTML text has no separate Fill and is drawn directly on the message dialog. The HTML text can be customized in a separate dialog. With the OnAnchorClick event you can call a ShellExecute to open the URL.

```
procedure TForm1.AdvSmoothMessageDialog1AnchorClick(Sender: TObject;
    Anchor: string);
begin
    ShellExecute(0, 'open', pchar(Anchor), nil, nil, SW_NORMAL);
end;
```

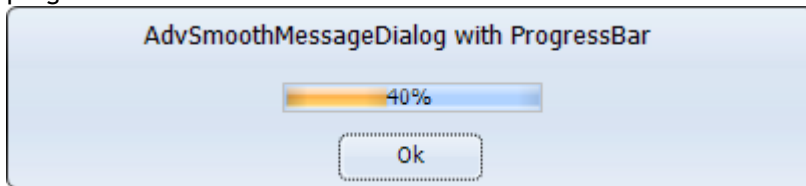
- **Background Fill:** the background fill of the message dialog is the fill behind the caption, html and buttons and can be complete transparent giving many possibilities to the message dialog.

Adding a progressbar

The Message dialog supports a progressbar, that can be positioned inside the messagedialog with the ProgressBarPosition property.

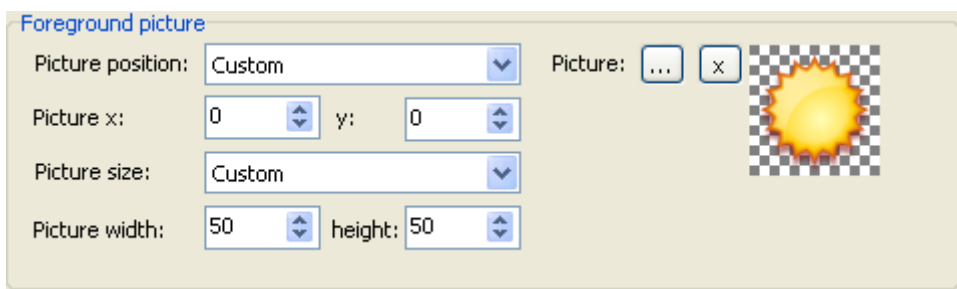
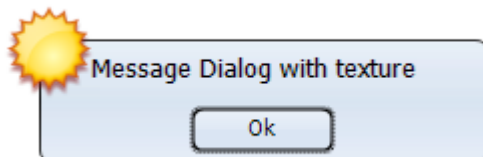
The ProgressBarValue ProgressBarMaximum and ProgressBarMinimum properties determine the Range and position of the ProgressBar.

The ProgressBarAppearance can be fully customized to add text, change the look and feel of the progressbar.



Using overlay textures

The AdvSmoothMessageDialog has a margin property to indent the messagedialog drawing. This opens the possibility to add overlay textures, textures that visually go out the draw area. Below is a sample of a default messagedialog with a margin and a texture. The captionfill.Picture is loaded with a PNG image and the picture position and size are customized.



```
AdvSmoothMessageDialog1.Caption := 'Message Dialog with texture';
AdvSmoothMessageDialog1.Position := poScreenCenter;
AdvSmoothMessageDialog1.Execute;
```