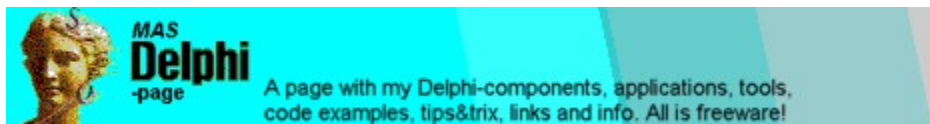


## MAS Components Help

[TLEDArray](#)  
[TMASPlayer](#)  
[TMDate](#)  
[TWKey](#)  
[TPushBtn](#)  
[TRegistTimes](#)  
[TmLED](#)  
[TToggleSwitch](#)  
[THPCounter](#)  
[TRText](#)  
[THLLabel](#)  
[TmWImage](#)  
[TmShortcut](#)  
[TmTagProducer](#)  
[Tm3Tag](#)  
[TmRegApp](#)  
[TmDataSetTableProducer](#)  
[TmGetNewVer](#)  
[TmThumbs](#)  
[TPassOverBtn](#)  
[TExecApp](#)  
[TmFilesScan](#)  
[TAppDlg](#)  
[TURLLabel](#)  
[TMDP](#)

[MASUtils, Code-Library](#)



Visit MAS Delphi page, <http://go.to/masdp>

This help file was created with [HelpScribble](#).

## ADlg unit

This unit declares the TADlg-component.

### Components

[TADlg](#)

This help file was created with [HelpScribble](#).



# TADlg component

[Properties](#)

[Methods](#)

[Tasks](#)

## Inherits from

[TComponent](#)

## Unit

[ADlg](#)

## Description

TADlg is a component for displaying application info to user.

This help file was created with [HelpScribble](#).

## Properties

▶ Run-time only

🔑 Key properties

🔑 [Email](#)

🔑 [Info](#)


🔑 [Productname](#)

🔑 [Site](#)

🔑 [Version](#)

This help file was created with [HelpScribble](#).

## Methods

 Key methods

 [Execute](#)

This help file was created with [HelpScribble](#).

# About the TADlg component

[TADlg reference](#)

## **Purpose**

TADlg is a component for displaying application info to user.

## **Tasks**

This help file was created with [HelpScribble](#).

# Email property

## Applies to

[TADlg](#) component

## Declaration

```
property Email: string;
```

## Description

Holds the E-mail adress of the applications author.

This help file was created with [HelpScribble](#).

# Info property

## Applies to

[TADlg](#) component

## Declaration

```
property Info: string;
```

## Description

Can be used for application info. For example: release date.

This help file was created with [HelpScribble](#).



# Productname property

## Applies to

[TADlg](#) component

## Declaration

```
property Productname: string;
```

## Description

Name of the application.

This help file was created with [HelpScribble](#).

# Site property

## Applies to

[TADlg](#) component

## Declaration

```
property Site: string;
```

## Description

Holds the site adress of the applications author.

This help file was created with [HelpScribble](#).

# Version property

## Applies to

[TADlg](#) component

## Declaration

```
property Version: string;
```

## Description

The applications version number.

This help file was created with [HelpScribble](#).

# Execute method

## Applies to

[TADlg](#) component

## Declaration

```
function Execute: boolean;
```

## Description

Creates an instance of ADlg and shows it in modal form to the user.

This help file was created with [HelpScribble](#).

## dlbox unit

This unit declares the TDIBox-component.

TDLBox is a component for downloading files from any given source into your computer

It's a compound component using TMForm as a wrapper form.

### Components

[TDLBox](#)

### Variables

[TMForm](#)

This help file was created with [HelpScribble](#).

# TMForm variable

[See also](#)

## Unit

[DLBox](#)

## Description

The wrapper form used by TDLBox.

This help file was created with [HelpScribble](#).

## See also

[TDLBox](#)

This help file was created with [HelpScribble](#).

# TDLBox component

[Properties](#)

[Methods](#)

[Events](#)

[Tasks](#)

## Inherits from

[TCompoundGroupBox](#)

## Unit

[dlbox](#)

## Description

TDLBox is a component for downloading files into your computer from any given source.

This help file was created with [HelpScribble](#).



## Events

### BitBtnOnClick

This help file was created with [HelpScribble](#).

## Properties

### ▶ Run-time only

#### Key properties

##### BitBtnEnabled

 BitBtnVisible

 Caption1

 Caption2

 Caption3

 Directory

 Drive

 DriveBoxVisible




#### Form

 Glyph

 Image

## Methods

 Key methods

 [UpdateDirList](#)

 [UpdateFileList](#)

This help file was created with [HelpScribble](#).

# About the TDLBox component

[TDLBox reference](#)

## Tasks

TDLBox is a component for downloading files into your computer from any given source.

This help file was created with [HelpScribble](#).

## BitBtnEnabled property

### Applies to

[TDLBox](#) component

### Declaration

```
property BitBtnEnabled: boolean;
```

### Description

Use Enabled to change the availability of the control to the user. To disable a control, set Enabled to False. Disabled controls appear dimmed. If Enabled is False, the control ignores mouse, keyboard, and timer events.

To re-enable a control, set Enabled to True. The control is no longer dimmed, and the user can use the control.

This help file was created with [HelpScribble](#).

# BitBtnOnClick event

## Applies to

[TDLBox](#) component

## Declaration

```
property BitBtnOnClick: TNotifyEvent;
```

## Description

Use the OnClick event handler to write code that responds when the user clicks the control. If the control has an associated action, and that action has an OnExecute method, the action's OnExecute method responds to click events unless it is superseded by an OnClick event handler.

Usually OnClick occurs because the user presses and releases the primary mouse button with the mouse pointer over the control. This event can also occur when

The user selects an item in a grid, outline, list, or combo box by pressing an arrow key.

The user presses Spacebar while a button or check box has focus.

The user presses Enter when the active form has a default button (specified by the Default property).

The user presses Esc when the active form has a cancel button (specified by the Cancel property).

The user presses the accelerator key for a button or check box. For example, if the value of the Caption property of a check box is '&Bold', the B is underlined at runtime and the OnClick event of the check box is triggered when the user presses Alt+B.

The Checked property of a radio button is set to True.

The value of the Checked property of a check box is changed.

The Click method of a menu item is called.

For a form, an OnClick event occurs when the user clicks a blank area of the form or on a disabled component.

This help file was created with [HelpScribble](#).

## BitBtnVisible property

### Applies to

[TDLBox](#) component

### Declaration

```
property BitBtnVisible: boolean;
```

### Description

Use the Visible property to control the visibility of the control at runtime. If Visible is True, the control appears. If Visible is False, the control is not visible.

This help file was created with [HelpScribble](#).

# Caption1 property

## Applies to

[TDLBox](#) component

## Declaration

```
property Caption1: string;
```

## Default caption

Your Computer:

## Description

Use Caption to specify the text string that labels the control.

This help file was created with [HelpScribble](#).



## Caption2 property

### Applies to

[TDLBox](#) component

### Declaration

```
property Caption2: string;
```

### Default caption

Download to:

### Description

Use Caption to specify the text string that labels the control.

This help file was created with [HelpScribble](#).

## Caption3 property

### Applies to

[TDLBox](#) component

### Declaration

```
property Caption3: string;
```

### Default caption

OK

### Description

Use Caption to specify the text string that labels the control.

This help file was created with [HelpScribble](#).

# Directory property

## Applies to

[TDLBox](#) component

## Declaration

```
property Directory: string;
```

## Description

The directory list box displays the value of the Directory property as the current directory in the list box.

This help file was created with [HelpScribble](#).

# Drive property

## Applies to

[TDLBox](#) component

## Declaration

```
property Drive: char;
```

## Description

Set Drive to initialize the combo box to a particular drive. Read Drive to obtain the drive that was selected by the user.

When the user uses the drive combo box to select a new drive, the selected drive becomes the value of the Drive property.

This help file was created with [HelpScribble](#).

# DriveBoxVisible property

## Applies to

[TDLBox](#) component

## Declaration

```
property DriveBoxVisible: boolean;
```

## Description

Use the Visible property to control the visibility of the control at runtime. If Visible is True, the control appears. If Visible is False, the control is not visible.

This help file was created with [HelpScribble](#).

# Form property

## Applies to

[TDLBox](#) component

## Declaration

```
property Form: TForm;
```

## Description

Property used by TDLBox to address the wrapper form.

Run-time only

Read-only

This help file was created with [HelpScribble](#).

## Glyph property

### Applies to

[TDLBox](#) component

### Declaration

```
property Glyph: TBitmap;
```

### Default glyph



### Description

Use the Open dialog box that appears as an editor in the Object Inspector to choose a bitmap file (with a .BMP extension) to use on the button, or specify a bitmap file at runtime.

You can provide up to four images within a single bitmap. All images must be the same size and next to each other in a row. Bit buttons display one of these images depending on their state.

If only one image is present, bit buttons attempt to represent the other states by altering the image slightly for the different states, although the Down state is always the same as the Up state. If you aren't satisfied with the results, you can provide additional images in the bitmap.

If you have multiple images in a bitmap, you must specify the number of images that are in the bitmap with the NumGlyphs property.

#### Note:

The lower left pixel of the bitmap is reserved for the "transparent" color. Any pixel in the bitmap which matches that lower left pixel will be transparent.

This help file was created with [HelpScribble](#).

## Image property

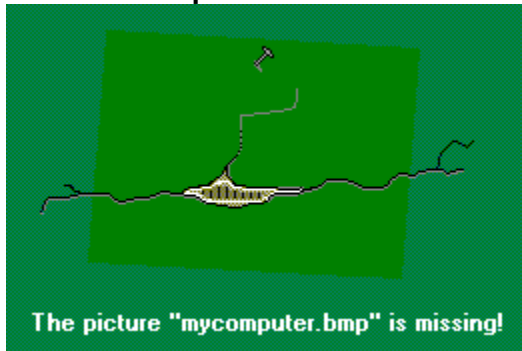
### Applies to

[TDLBox](#) component

### Declaration

```
property Image: TBitmap;
```

### Default bitmap



### Description

Specifies the bitmap picture to be shown in top left corner of the component.

This help file was created with [HelpScribble](#).



# UpdateDirList method

## Applies to

[TDLBox](#) component

## Declaration

```
function UpdateDirList: boolean;
```

## Description

Call Update to refresh the directory list after new directories have been created or existing directories deleted.

This help file was created with [HelpScribble](#).

# UpdateFileList method

## Applies to

[TDLBox](#) component

## Declaration

```
function UpdateFileList: boolean;
```

## Description

Call Update whenever the files listed by the file list box may be out of date. For example, if an application creates, renames, or deletes files, calling Update will ensure that those changes are reflected in the file list box.

This help file was created with [HelpScribble](#).

# GQuit unit

This unit declares the TGQuit-component.

TGQuit is a Quit-dialog component that can be used for program-termination.

## Components

[TGQuit](#)

## Variables

[TGCFORM](#)

This help file was created with [HelpScribble](#).

# TGCFORM variable

[See also](#)

## Unit

[GQuit](#)

## Description

TGCFORM is the wrapper form used by TGQuit.

This help file was created with [HelpScribble](#).

## See also

[TGQuit](#)

This help file was created with [HelpScribble](#).



# TGQuit component

[Properties](#)

[Events](#)

[Tasks](#)

## Inherits from

[TCompoundGroupBox](#)

## Unit

[GQuit](#)


## Description

TGQuit is a Quit-dialog component that can be used for program-termination.

This help file was created with [HelpScribble](#).

## Properties

▶ Run-time only

 Key properties

 Button1Delay

 Button2Delay

 CaptionButton1

 CaptionButton2

 CaptionLabel

▶

 Form

 Pic1

 Pic2

 Pic3

This help file was created with [HelpScribble](#).

## Events

### Key events

 [OnButton1DelayedClick](#)

 [OnButton2DelayedClick](#)

This help file was created with [HelpScribble](#).



# About the TGQuit component

[TGQuit reference](#)

## Tasks

TGQuit is a Quit-dialog component that can be used for program-termination.

This help file was created with [HelpScribble](#).

# Button1Delay property

[See also](#)

## Applies to

[TGQuit](#) component

## Declaration

```
property Button1Delay: integer;
```

## Description

Sets the delay in msec:s for the OnButton1DelayedClick event.

This help file was created with [HelpScribble](#).

## See also

[OnButton1DelayedClick](#)

This help file was created with [HelpScribble](#).

# Button2Delay property

[See also](#)

## Applies to

[TGQuit](#) component

## Declaration

```
property Button2Delay: integer;
```

## Description

Sets the delay in msec:s for the OnButton2DelayedClick event.

This help file was created with [HelpScribble](#).

## See also

[OnButton2DelayedClick](#)

This help file was created with [HelpScribble](#).

# CaptionButton1 property

## Applies to

[TGQuit](#) component

## Declaration

```
property CaptionButton1: string;
```

## Default caption

Yes

## Description

Use Caption to specify the text string that labels the control.

To underline a character in a Caption that labels a component, include an ampersand (&) before the character. This type of character is called an accelerator character. The user can then select the component by pressing Alt while typing the underlined character. To display an ampersand character in the caption, use two ampersands (&&).

### Note:

Controls that display text use either the Caption property or the Text property to specify the text value. Which property is used depends on the type of control. In general, Caption is used for text that appears as a window title or label, while Text is used for text that appears as the content of a control.

This help file was created with [HelpScribble](#).

# CaptionButton2 property

## Applies to

[TGQuit](#) component

## Declaration

```
property CaptionButton2: string;
```

## Default caption

No

## Description

Use Caption to specify the text string that labels the control.

To underline a character in a Caption that labels a component, include an ampersand (&) before the character. This type of character is called an accelerator character. The user can then select the component by pressing Alt while typing the underlined character. To display an ampersand character in the caption, use two ampersands (&&).

### Note:

Controls that display text use either the Caption property or the Text property to specify the text value. Which property is used depends on the type of control. In general, Caption is used for text that appears as a window title or label, while Text is used for text that appears as the content of a control.

This help file was created with [HelpScribble](#).

# CaptionLabel property

## Applies to

[TGQuit](#) component

## Declaration

```
property CaptionLabel: string;
```

## Default caption

Quit playing?

## Description

Use Caption to specify the text string that labels the control.

To underline a character in a Caption that labels a component, include an ampersand (&) before the character. This type of character is called an accelerator character. The user can then select the component by pressing Alt while typing the underlined character. To display an ampersand character in the caption, use two ampersands (&&).

### Note:

Controls that display text use either the Caption property or the Text property to specify the text value. Which property is used depends on the type of control. In general, Caption is used for text that appears as a window title or label, while Text is used for text that appears as the content of a control.

This help file was created with [HelpScribble](#).



# Form property

## Applies to

[TGQuit](#) component

## Declaration

```
property Form: TGCFORM;
```

## Description

Property used by TGQuit to address the wrapper form.

Run-time only

Read-only

This help file was created with [HelpScribble](#).

# Pic1 property

## Applies to

[TGQuit](#) component

## Declaration

```
property Pic1: TBitmap;
```

## Default bitmap



## Description

Specifies the bitmap picture to be shown when user clicks Button1 (default caption = Yes).

This help file was created with [HelpScribble](#).

## Pic2 property

### Applies to

[TGQuit](#) component

### Declaration

```
property Pic2: TBitmap;
```

### Default bitmap



### Description

Specifies the bitmap picture to be shown when there are no buttons clicked.

## Pic3 property

### Applies to

[TGQuit](#) component

### Declaration

```
property Pic3: TBitmap;
```

### Default bitmap



### Description

Specifies the bitmap picture to be shown when user clicks Button2 (default caption = No).

This help file was created with [HelpScribble](#).

# OnButton1DelayedClick event

[See also](#)

## Applies to

[TGQuit](#) component

## Declaration

**property** OnButton1DelayedClick: TNotifyEvent;

## Description

Event raised by TGQuit after Button1Delay when Button1 is clicked.

This help file was created with [HelpScribble](#).

## See also

[Button1Delay](#)

This help file was created with [HelpScribble](#).

# OnButton2DelayedClick event

[See also](#)

## Applies to

[TGQuit](#) component

## Declaration

**property** OnButton2DelayedClick: TNotifyEvent;

## Description

Event raised by TGQuit after Button2Delay when Button2 is clicked.

This help file was created with [HelpScribble](#).

## See also

[Button2Delay](#)

This help file was created with [HelpScribble](#).



# LED unit

This unit declares the TLEDArray-component.  
TLEDArray is a component for displaying a number of LED:s.

## Components

[TLEDArray](#)

## Types

[TShapeType](#)

This help file was created with [HelpScribble](#).

# TShapeType

**Type** TShapeType = (stRectangle, stSquare, stRoundRect, stRoundSquare, stEllipse, stCircle);

**property** Shape: TShapeType;

## Description

Set Shape to the geometric shape that should be drawn on the form. The Shape property has these possible values:

Value	Meaning
stCircle	The shape is a circle.
stEllipse	The shape is an ellipse.
stRectangle	The shape is a rectangle.
stRoundRect	The shape is a rectangle with rounded corners.
stRoundSquare	The shape is a square with rounded corners.
stSquare	The shape is a square.

This help file was created with [HelpScribble](#).



# TLEDArray component

[Properties](#)

[Events](#)

[Tasks](#)

## Inherits from

[TShape](#)

## Unit

[LED](#)

## Description

TLEDArray is a component for displaying a number of LED:s.

This help file was created with [HelpScribble](#).

## Properties

▶ Run-time only

🔑 Key properties

🔑 NoOfLEDs

🔑 OnOff

🔑 Space

Align

Anchors

Brush

Constraints

DragCursor

DragKind

DragMode

Enabled

ParentShowHint

Pen

Shape

ShowHint

Visible

## Events

### Key events

[OnDragDrop](#)  
[OnDragOver](#)  
[OnEndDock](#)  
[OnEndDrag](#)  
[OnMouseDown](#)  
[OnMouseMove](#)  
[OnMouseUp](#)  
[OnStartDock](#)  
[OnStartDrag](#)

This help file was created with [HelpScribble](#).

# About the TLEDArry component

[TLED reference](#)

## Tasks

TLEDArry is a component for displaying a number of LED:s.

This help file was created with [HelpScribble](#).

# NoOfLEDs property

## Applies to

[TLEDArray](#) component

## Declaration

```
property NoOfLEDs: integer;
```

## Description

Specifies how many LED:s to be displayed. Maximal number = 32.

This help file was created with [HelpScribble](#).

# OnOff property

## Applies to

[TLEDArray](#) component

## Declaration

```
property OnOff: cardinal;
```

## Description

Sets the LED:s on/off. See the Led-positions as binary bit positions.

## Example:

An On/Off value of 255 will turn all LED:s on in an array with 8 LED:s.

This help file was created with [HelpScribble](#).



# Space property

## Applies to

[TLEDArray](#) component

## Declaration

```
property Space: integer;
```

## Description

Specifies the space between LED:s in pixels.

This help file was created with [HelpScribble](#).

## MasMeter unit

This unit declares the TMasMeter-component.  
A progress bar control

### Components

[TMasMeter](#)

This help file was created with [HelpScribble](#).

## **TMasMeter component**

[Properties](#)

### **Inherits from**

[TGraphicControl](#)

### **Unit**


[MasMeter](#)


### **Description**


<<< Description of TMasMeter component >>>

This help file was created with [HelpScribble](#).

## Properties

 Run-time only

 Key properties

 BColor

 Color

 PercentShaded

Enabled

Height

Width

Visible

This help file was created with [HelpScribble](#).

# BColor property

## Applies to

[TMasMeter](#) component

## Declaration

```
property BColor: TColor;
```

## Description

Sets the bar's background colour.

This help file was created with [HelpScribble](#).

# Color property

## Applies to

[TMasMeter](#) component

## Declaration

```
property Color: TColor;
```

## Description

Sets the bar colour.

This help file was created with [HelpScribble](#).

# PercentShaded property

[Example](#)

## Applies to

[TMasMeter](#) component

## Declaration

**property** PercentShaded: Integer;

## Description

Sets the bar progress in percent.

This help file was created with [HelpScribble](#).

## PercentShaded property example

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    Pr:=0;  
    MasMeter1.PercentShaded:=0;  
    MasMeter1.Color:=clYellow;  
    MasTimer1.Enabled:=true;  
end;
```

```
procedure TForm1.Timer1Timer(Sender: TObject);  
begin  
    Inc(Pr);  
    if Pr = 25 then MasMeter1.Color:=clLime;  
    if Pr = 50 then MasMeter1.Color:=clBlue;  
    if Pr = 75 then MasMeter1.Color:=clRed;  
    if Pr = 100 then MasTimer1.Enabled:=false;  
    MasMeter1.PercentShaded:=Pr;  
    Label1.Caption:=IntToStr(Pr)+'%';  
end;
```

This help file was created with [HelpScribble](#).



# MasPlayer unit

This unit declares the TMasPlayer-component.  
An improved TMediaPlayer.

## Components

[TMasPlayer](#)

This help file was created with [HelpScribble](#).



# **TMasPlayer component**

[Properties](#)

[Methods](#)

[Example](#)

## **Inherits from**

[TMediaPlayer](#)

## **Unit**


[MasPlayer](#)


## **Description**

An improved TMediaPlayer.

This help file was created with [HelpScribble](#).

## Properties

 Run-time only


 Key properties

 [MediaFileIndex](#)

 [MediaFiles](#)

This help file was created with [HelpScribble](#).

## Methods

 Key methods

 [PlayItem](#)

This help file was created with [HelpScribble](#).

## **TMasPlayer example**

```
procedure TForm1.SpinEdit1Change(Sender: TObject);  
begin  
  with MasPlayer1 do  
    begin  
      Edit1.Text:=MediaFiles[SpinEdit1.Value];  
      PlayItem(SpinEdit1.Value);  
    end;  
end;
```

This help file was created with [HelpScribble](#).

# MediaFileIndex property

## Applies to

[TMasPlayer](#) component

## Declaration

```
property MediaFileIndex: Integer;
```

## Description

Index for the items in the mediafile-list. First one = 0.

This help file was created with [HelpScribble](#).

# MediaFiles property

## Applies to

[TMasPlayer](#) component

## Declaration

```
property MediaFiles: TStrings ;
```

## Description

A TStringList that holds the Mediafile-names.

This help file was created with [HelpScribble](#).

# PlayItem method

[Example](#)

## Applies to

[TMasPlayer](#) component

## Declaration

```
procedure PlayItem(Value: integer);
```

## Description

Retrieves an item in the mediafile-list, opens it and plays it. First item = 0.

This help file was created with [HelpScribble](#).



## MBox unit

This unit declares the TMBBox-component.  
TMBBox displays a messagebox of various types.

### Components

[TMBBox](#)

This help file was created with [HelpScribble](#).



# **TMBox component**

[Properties](#)

[Methods](#)

## **Inherits from**

[TComponent](#)

## **Unit**


[MBox](#)


## **Description**


TMBox displays a messagebox of various types.

This help file was created with [HelpScribble](#).

## Properties


 Run-time only

 Key properties

 BoxType


 Buttons

 Caption

 Text

This help file was created with [HelpScribble](#).

## Methods

 Key methods

 [ButtonPressed](#)

 [Execute](#)

This help file was created with [HelpScribble](#).

# BoxType property

## Applies to

[TMsgDlg](#) component

## Declaration

```
property BoxType: TMsgDlgType;
```

## Description

Specifies the Boxtype to be displayed.

The TMsgDlgType type defines the values describing the type of message box. The TMsgDlgType is used by the MessageDlg and MessageDlgPos functions. The following table lists the possible values:

Value    Meaning

---

mtWarning	A message box containing a yellow exclamation point symbol.
mtError	A message box containing a red stop sign.
mtInformation	A message box containing a blue "i".
mtConfirmation	A message box containing a green question mark.
mtCustom	A message box containing no bitmap. The caption of the message box is the name of the application's executable file.

This help file was created with [HelpScribble](#).

# Buttons property

## Applies to

[TMsgDlgButtons](#) component

## Declaration

**property** Buttons: TMsgDlgButtons;

## Description

The TMsgDlgButtons type defines the set of values a button in a message box can have. The TMsgDlgButtons type is used by the MessageDlg and MessageDlgPos functions. The following tables lists the possible values:

Value    Meaning

---

mbYes	A button with 'Yes' on its face.
mbNo	A button the text 'No' on its face.
mbOK	A button the text 'OK' on its face.
mbCancel	A button with the text 'Cancel' on its face.
mbAbort	A button with the text 'Abort' on its face
mbRetry	A button with the text 'Retry' on its face
mbIgnore	A button the text 'Ignore' on its face
mbAll	A button with the text 'All' on its face
mbNoToAll	A button with the text 'No to All' on its face
mbYesToAll	A button with the text 'Yes to All' on its face
mbHelp	A button with the text 'Help' on its face

In addition, the dialogs unit defines three constants for commonly used TMsgDlgButtons values:

Constant	Meaning
mbYesNoCancel	mbYes, mbNo, and mbCancel
mbOKCancel	mbOK and mbCancel
mbAbortRetryIgnore	mbAbort, mbRetry, and mbIgnore

This help file was created with [HelpScribble](#).

# Caption property

## Applies to

[TMBBox](#) component

## Declaration

```
property Caption: String;
```

## Description

Use Caption to specify the text string that labels the control.

To underline a character in a Caption that labels a component, include an ampersand (&) before the character. This type of character is called an accelerator character. The user can then select the component by pressing Alt while typing the underlined character. To display an ampersand character in the caption, use two ampersands (&&).

### Note:

Controls that display text use either the Caption property or the Text property to specify the text value. Which property is used depends on the type of control. In general, Caption is used for text that appears as a window title or label, while Text is used for text that appears as the content of a control.

This help file was created with [HelpScribble](#).

# Text property

## Applies to

[TMessageBox](#) component

## Declaration

```
property Text: String;
```

## Description

Holds the string with the message to be displayed.

This help file was created with [HelpScribble](#).



# ButtonPressed method

[See also](#)

[Example](#)

## Applies to

[TMessageBox](#) component

## Declaration

```
function ButtonPressed: string;
```

## Description

Read ButtonPressed after Execute to determine which Button pressed.

Buttontype	ButtonPressed returns
mbYes	'Yes'
mbNo	'No'
mbOK	'OK'
mbCancel	'Cancel'
mbAbort	'Abort'
mbRetry	'Retry'
mbIgnore	'Ignore'
mbAll	'All'
mbNoToAll	'No to All'
mbYesToAll	'Yes to All'
mbHelp	'Help'

This help file was created with [HelpScribble](#).

## See also

[Execute](#)

This help file was created with [HelpScribble](#).

## Example ButtonPressed

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  case RadioGroup1.ItemIndex of
    0: begin
        MBox1.Caption:='Information';
        MBox1.Text:='This is an Information-message.';
      end;
    1: begin
        MBox2.Caption:='Warning';
        MBox2.Text:='This is a Warning-message.';
      end;
    2: begin
        MBox3.Caption:='Confirmation';
        MBox3.Text:='This is a Confirmation-message.';
      end;
    3: begin
        MBox4.Caption:='Custom';
        MBox4.Text:='This is a Custom-message.';
      end;
    4: begin
        MBox5.Caption:='Error';
        MBox5.Text:='This is an Error-message.';
      end;
  end;
  TMessageBox(FindComponent('MBox'+IntToStr(RadioGroup1.ItemIndex+1))).Execute;
  Caption:='Button pressed:
'+TMessageBox(FindComponent('MBox'+IntToStr(RadioGroup1.ItemIndex+1))).ButtonPressed;
end;
```

This help file was created with [HelpScribble](#).

# Execute method

[See also](#)

[Example](#)

## Applies to

[TMBox](#) component

## Declaration

```
function Execute: Integer; virtual;
```

## Description

Creates an instance of TMBox and shows it in modal form to the user.

This help file was created with [HelpScribble](#).

## See also

[ButtonPressed](#)

This help file was created with [HelpScribble](#).

# MDate unit

This unit declares the TMDate-component.  
A component for displaying date and time in various ways.

## Components

[TMDate](#)

This help file was created with [HelpScribble](#).



# **TMDate component**

[Properties](#)

## **Unit**


[MDate](#)


## **Description**

A component for displaying date and time in various ways.

This help file was created with [HelpScribble](#).

## Properties

 Run-time only

 Key properties

 [DateFmtUSA](#)



 [DateStr](#)

[Font](#)

 [Numeric1](#)

 [Numeric2](#)

 [ShowDate](#)

 [ShowTime](#)

 [Swedish](#)

 [TimeFmt24hr](#)

 [TimeSep](#)



 [TimeStr](#)

[Visible](#)

This help file was created with [HelpScribble](#).



# DateFmtUSA property

## Applies to

[TMDate](#) component

## Declaration

```
property DateFmtUSA: Boolean;
```

## Description

<<< Description of DateFmtUSA property >>>

This help file was created with [HelpScribble](#).

# DateStr property

## Applies to

[TMDate](#) component

## Declaration

```
property DateStr: String;
```

## Description

<<< Description of DateStr property >>>

Run-time only

Read-only

This help file was created with [HelpScribe](#).

# Numeric1 property

## Applies to

[TMDate](#) component

## Declaration

```
property Numeric1: Boolean;
```

## Description

<<< Description of Numeric1 property >>>

This help file was created with [HelpScribble](#).

# Numeric2 property

## Applies to

[TMDate](#) component

## Declaration

```
property Numeric2: Boolean;
```

## Description

<<< Description of Numeric2 property >>>

This help file was created with [HelpScribble](#).

# ShowDate property

## Applies to

[TMDate](#) component

## Declaration

```
property ShowDate: Boolean;
```

## Description

<<< Description of ShowDate property >>>

This help file was created with [HelpScribble](#).

# ShowTime property

## Applies to

[TMDate](#) component

## Declaration

```
property ShowTime: Boolean;
```

## Description

<<< Description of ShowTime property >>>

This help file was created with [HelpScribble](#).

# Swedish property

## Applies to

[TMDate](#) component

## Declaration

`property` Swedish: Boolean;

## Description

<<< Description of Swedish property >>>

This help file was created with [HelpScribble](#).

# TimeFmt24hr property

## Applies to

[TMDate](#) component

## Declaration

```
property TimeFmt24hr: Boolean;
```

## Description

<<< Description of TimeFmt24hr property >>>

This help file was created with [HelpScribble](#).



# TimeSep property

## Applies to

[TMDate](#) component

## Declaration

```
property TimeSep: String;
```

## Description

<<< Description of TimeSep property >>>

This help file was created with [HelpScribble](#).

# TimeStr property

{keepn}

## Applies to

[TMDate](#) component

## Declaration

```
property TimeStr: String;
```

## Description

<<< Description of TimeStr property >>>

Run-time only

Read-only

This help file was created with [HelpScribble](#).

# MG unit

This unit declares the TMGauge-component.  
A progress bar.

## Components

[TMGauge](#)

## Types

[TMColour](#)

## Variables

[TMGForm](#)

This help file was created with [HelpScribble](#).

## See also

<<< See also of MG unit >>>

This help file was created with [HelpScribe](#).

# TMGForm variable

[See also](#)

[Tasks](#)

**Unit**

[MG](#)

## **Description**

<<< Description of TMGForm component >>>

This help file was created with [HelpScribble](#).

## See also

<<< See also of TMGForm component >>>

This help file was created with [HelpScribble](#).

# About the TMGForm component

See also [TMGForm reference](#)

## **Purpose**

<<< purpose of the TMGForm component >>>

## **Tasks**

<<< how to use the TMGForm component >>>

This help file was created with [HelpScribble](#).

## See also

<<< See also of About the TMGForm component >>>

This help file was created with [HelpScribble](#).





# TMGauge component

[See also](#)

[Properties](#)

[Methods](#)

[Tasks](#)

## Inherits from

[TCompoundPanel](#)

## Unit

[MG](#)

## Description

<<< Description of TMGauge component >>>


This help file was created with [HelpScribble](#).


## See also


<<< See also of TMGauge component >>>

This help file was created with [HelpScribble](#).

## Properties

 Run-time only

 Key properties

 ColourBar




 Form

  ProgressPercent

This help file was created with [HelpScribble](#).

## Methods

 Key methods

[Create](#)

[Destroy](#)

This help file was created with [HelpScribble](#).

# About the TMGauge component

See also [TMGauge reference](#)

## **Purpose**

<<< purpose of the TMGauge component >>>

## **Tasks**

<<< how to use the TMGauge component >>>

This help file was created with [HelpScribble](#).

## See also

<<< See also of About the TMGauge component >>>

This help file was created with [HelpScribble](#).

# ColourBar property

[See also](#)      [Example](#)

## Applies to

[TMGauge](#) component

## Declaration

```
property ColourBar: TMCcolour;
```

## Description

<<< Description of ColourBar property >>>

This help file was created with [HelpScribble](#).

## See also

<<< See also of ColourBar property >>>

This help file was created with [HelpScribble](#).



## ColourBar property example

This help file was created with [HelpScribble](#).

# Form property

[See also](#)      [Example](#)

## Applies to

[TMGauge](#) component

## Declaration

**property** Form: TMGForm;

## Description

<<< Description of Form property >>>

Run-time only

Read-only

This help file was created with [HelpScribble](#).

## See also

<<< See also of Form property >>>

This help file was created with [HelpScribble](#).

## Form property example

This help file was created with [HelpScribe](#).

# ProgressPercent property

[See also](#)      [Example](#)

## Applies to

[TMGauge](#) component

## Declaration

```
property ProgressPercent: integer;
```

## Description

<<< Description of ProgressPercent property >>>

This help file was created with [HelpScribble](#).

## See also

<<< See also of ProgressPercent property >>>

This help file was created with [HelpScribble](#).

## ProgressPercent property example

This help file was created with [HelpScribble](#).

# TMCColour type

[See also](#)

**Unit**

[MG](#)

## Declaration

```
type TMCColour = (mcBlue, mcLime, mcRed, mcYellow);
```

## Description

<<< Description of TMCColour type >>>

This help file was created with [HelpScribble](#).



## See also

<<< See also of TMCcolour type >>>

This help file was created with [HelpScribble](#).

## MSearch unit

This unit declares the TMSearch-component.  
TMSearch is a component for easy search of string-occurrences in a text.

### Components

[TMSearch](#)

This help file was created with [HelpScribble](#).



# TMSearch component

[Properties](#)

[Methods](#)

[Tasks](#)

## Inherits from

[TRichEdit](#)

## Unit


[MSearch](#)


## Description


TMSearch is a component for easy search of string-occurrences in a text.

This help file was created with [HelpScribble](#).

## Properties

 Run-time only

 Key properties

 CaseSensitive

 FoundLine


 FoundPos

 SearchLine

 SearchPos

This help file was created with [HelpScribble](#).

## Methods

 Key methods

 [FindString](#)

This help file was created with [HelpScribble](#).

# About the TMSearch component

See also [TMSearch reference](#)

## **Purpose**

<<< purpose of the TMSearch component >>>

## **Tasks**

<<< how to use the TMSearch component >>>

This help file was created with [HelpScribble](#).

## See also

<<< See also of About the TMSearch component >>>

This help file was created with [HelpScribble](#).

# CaseSensitive property

## Applies to

[TMSearch](#) component

## Declaration

```
property CaseSensitive: boolean;
```

## Description

Determines whether the search is to be casesensitive or not.  
True = casesensitive search.

This help file was created with [HelpScribble](#).



# FoundLine property

[See also](#)

[Example](#)

## Applies to

[TMSearch](#) component

## Declaration

```
property FoundLine: integer;
```

## Description

The line where the string was last found. Set to zero by FindString if string was not found.

This help file was created with [HelpScribble](#).

## See also

[FoundPos](#)

[FindString](#)

This help file was created with [HelpScribble](#).

## FoundLine property example

```
procedure TForm1.Button1Click(Sender: TObject);  
var count: integer;  
begin  
    count:=0;  
    Memo1.Clear;  
    Memo1.Lines.Add('Searched for string: '"+Edit1.Text+"'");  
    Memo1.Lines.Add("");  
    if RadioGroup1.ItemIndex = 0 then MSearch1.CaseSensitive:=false  
    else MSearch1.CaseSensitive:=true;  
    MSearch1.SearchLine:=0;  
    MSearch1.SearchPos:=1;  
    while MSearch1.FindString(Edit1.Text) do  
        begin  
            Memo1.Lines.Add('Found at line: '+IntToStr(MSearch1.FoundLine)+' , position:  
'+IntToStr(MSearch1.FoundPos)+' ' "  
                +Copy(MSearch1.Lines[MSearch1.FoundLine], MSearch1.FoundPos, 30)+'...'");  
            Inc(count);  
        end;  
        Memo1.Lines.Add("");  
        Memo1.Lines.Add('Found '+IntToStr(count)+' matches.');
```

**end;**

# FoundPos property

[See also](#)

[Example](#)

## Applies to

[TMSearch](#) component

## Declaration

```
property FoundPos: integer;
```

## Description

The character-position in FoundLine, where the string was last found. Set to zero by FindString if string was not found.

This help file was created with [HelpScribble](#).

## See also

[FoundLine](#)  
[FindString](#)

This help file was created with [HelpScribble](#).

## FoundPos property example

```
procedure TForm1.Button1Click(Sender: TObject);  
var count: integer;  
begin  
    count:=0;  
    Memo1.Clear;  
    Memo1.Lines.Add('Searched for string: '"+Edit1.Text+"'");  
    Memo1.Lines.Add("");  
    if RadioGroup1.ItemIndex = 0 then MSearch1.CaseSensitive:=false  
    else MSearch1.CaseSensitive:=true;  
    MSearch1.SearchLine:=0;  
    MSearch1.SearchPos:=1;  
    while MSearch1.FindString(Edit1.Text) do  
        begin  
            Memo1.Lines.Add('Found at line: '+IntToStr(MSearch1.FoundLine)+' , position:  
'+IntToStr(MSearch1.FoundPos)+' ' "  
                +Copy(MSearch1.Lines[MSearch1.FoundLine], MSearch1.FoundPos, 30)+'...'");  
            Inc(count);  
        end;  
    Memo1.Lines.Add("");  
    Memo1.Lines.Add('Found '+IntToStr(count)+' matches.');
```

**end;**

# SearchLine property

[See also](#)

[Example](#)

## Applies to

[TMSearch](#) component

## Declaration

```
property SearchLine: integer;
```

## Description

The line to be searched. SearchLine is automatically increased when calling FindString.

This help file was created with [HelpScribble](#).

## See also

[SearchPos](#)  
[FindString](#)

This help file was created with [HelpScribble](#).



## SearchLine property example

```
procedure TForm1.Button1Click(Sender: TObject);  
var count: integer;  
begin  
    count:=0;  
    Memo1.Clear;  
    Memo1.Lines.Add('Searched for string: '"+Edit1.Text+"'");  
    Memo1.Lines.Add("");  
    if RadioGroup1.ItemIndex = 0 then MSearch1.CaseSensitive:=false  
    else MSearch1.CaseSensitive:=true;  
    MSearch1.SearchLine:=0;  
    MSearch1.SearchPos:=1;  
    while MSearch1.FindString(Edit1.Text) do  
        begin  
            Memo1.Lines.Add('Found at line: '+IntToStr(MSearch1.FoundLine)+' , position:  
'+IntToStr(MSearch1.FoundPos)+' ' "  
                +Copy(MSearch1.Lines[MSearch1.FoundLine], MSearch1.FoundPos, 30)+'...'");  
            Inc(count);  
        end;  
        Memo1.Lines.Add("");  
        Memo1.Lines.Add('Found '+IntToStr(count)+' matches.');
```

**end**;

# SearchPos property

[See also](#)

[Example](#)

## Applies to

[TMSearch](#) component

## Declaration

```
property SearchPos: integer;
```

## Description

The first character-position in SearchLine to start search from. SearchPos is automatically increased when calling FindString.

This help file was created with [HelpScribble](#).

## See also

[SearchLine](#)  
[FindString](#)

This help file was created with [HelpScribble](#).

## SearchPos property example

```
procedure TForm1.Button1Click(Sender: TObject);  
var count: integer;  
begin  
    count:=0;  
    Memo1.Clear;  
    Memo1.Lines.Add('Searched for string: '"+Edit1.Text+"'");  
    Memo1.Lines.Add("");  
    if RadioGroup1.ItemIndex = 0 then MSearch1.CaseSensitive:=false  
    else MSearch1.CaseSensitive:=true;  
    MSearch1.SearchLine:=0;  
    MSearch1.SearchPos:=1;  
    while MSearch1.FindString(Edit1.Text) do  
        begin  
            Memo1.Lines.Add('Found at line: '+IntToStr(MSearch1.FoundLine)+' , position:  
'+IntToStr(MSearch1.FoundPos)+' ' "  
                +Copy(MSearch1.Lines[MSearch1.FoundLine], MSearch1.FoundPos, 30)+'...'");  
            Inc(count);  
        end;  
    Memo1.Lines.Add("");  
    Memo1.Lines.Add('Found '+IntToStr(count)+' matches.');
```

**end;**

# FindString method

[Example](#)

## Applies to

[TMSearch](#) component

## Declaration

```
function FindString(StringToFind: string): boolean;
```

## Description

Pass the string to search for to this function. Findstring will be set to true if the string was found.

If true:

Read FoundLine and FoundPos to determine the string-position in the text. SearchPos and Searchline will be updated, ready for a new search.

If false:

FoundLine, FoundPos, SearchLine and SearchPos will be set to zero.

This help file was created with [HelpScribble](#).

## See also

[FoundLine](#)  
[FoundPos](#)  
[SearchLine](#)  
[SearchPos](#)

This help file was created with [HelpScribble](#).

## FindString method example

```
procedure TForm1.Button1Click(Sender: TObject);  
var count: integer;  
begin  
    count:=0;  
    Memo1.Clear;  
    Memo1.Lines.Add('Searched for string: '"+Edit1.Text+"'");  
    Memo1.Lines.Add("");  
    if RadioGroup1.ItemIndex = 0 then MSearch1.CaseSensitive:=false  
    else MSearch1.CaseSensitive:=true;  
    MSearch1.SearchLine:=0;  
    MSearch1.SearchPos:=1;  
    while MSearch1.FindString(Edit1.Text) do  
        begin  
            Memo1.Lines.Add('Found at line: '+IntToStr(MSearch1.FoundLine)+' , position:  
'+IntToStr(MSearch1.FoundPos)+' ' "  
                +Copy(MSearch1.Lines[MSearch1.FoundLine], MSearch1.FoundPos, 30)+'...'");  
            Inc(count);  
        end;  
        Memo1.Lines.Add("");  
        Memo1.Lines.Add('Found '+IntToStr(count)+' matches.');
```

**end;**

# PCar unit

This unit declares the TPCar-component.  
TPCar is a kind of funny progressbar.

## Components

[TPCar](#)

## Variables

[TPCForm](#)

This help file was created with [HelpScribble](#).



## TPCForm variable

[See also](#)

### Unit

[PCar](#)

### Description

TPCForm is the wrapper form used by TPCar.

This help file was created with [HelpScribble](#).

## See also

[TPCar component](#)

This help file was created with [HelpScribble](#).



# TPCar component

[See also](#)

[Properties](#)

## Inherits from

[TCompoundPanel](#)

## Unit

[PCar](#)

## Description

TPCar is a kind of funny progressbar.


This help file was created with [HelpScribble](#).


## See also

[TPCForm component](#)

This help file was created with [HelpScribble](#).

## Properties

 Run-time only

 Key properties

 [Bitmap1](#)

 [Bitmap2](#)



 [Form](#)

 [ProgressPercent](#)

This help file was created with [HelpScribble](#).

## Bitmap1 property

[See also](#)

### Applies to

[TPCar](#) component

### Default bitmap



### Declaration

```
property Bitmap1: TBitmap;
```

### Description

Sets, holds Bitmap1. BitMap1 is the moving object.

This help file was created with [HelpScribble](#).

## See also

[Bitmap2](#)

This help file was created with [HelpScribble](#).

## Bitmap2 property

[See also](#)

### Applies to

[TPCar](#) component

### Default bitmap



### Declaration

```
property Bitmap2: TBitmap;
```

### Description

Sets, holds Bitmap2. Bitmap2 is the static object.

This help file was created with [HelpScribble](#).



## See also

[Bitmap1](#)

This help file was created with [HelpScribble](#).

# Form property

## Applies to

[TPCar](#) component

## Declaration

```
property Form: TPCForm;
```

## Description

Property used by TPCar to adress the wrapper form.

Run-time only

Read-only

This help file was created with [HelpScribble](#).

# ProgressPercent property

[Example](#)

## Applies to

[TPCar](#) component

## Declaration

```
property ProgressPercent: integer;
```

## Description

Sets the amount (0-100) of progress for BitMap1;

This help file was created with [HelpScribble](#).

## ProgressPercent property example

```
// car-race
procedure TForm1.Button1Click(Sender: TObject);
begin
    PCar1.ProgressPercent:=0;
    PCar2.ProgressPercent:=0;
    Caption:='Start';
    Timer1.Enabled:=true;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
    Caption:="";
    Randomize;
    if Random(2) = 0 then PCar1.ProgressPercent:=PCar1.ProgressPercent+1;
    if Random(2) = 0 then PCar2.ProgressPercent:=PCar2.ProgressPercent+1;
    if (PCar1.ProgressPercent = 100) then
    begin
        Caption:='Car no.1 is the winner !';
        Timer1.Enabled:=false;
    end;
    if (PCar2.ProgressPercent = 100) then
    begin
        Caption:='Car no.2 is the winner !';
        Timer1.Enabled:=false;
    end;
end;
```

# Stepper unit

This unit declares the TStepper-component.  
A step control component.

## Components

[TStepper](#)

## Types

[TStepperSize](#)

## Variables

[TStepForm](#)

This help file was created with [HelpScribble](#).

# TStepForm variable

## Unit

[Stepper](#)

## Description

The wrapper Form used by TStepper.

This help file was created with [HelpScribble](#).



# TStepper component

[Properties](#)

[Events](#)

## Inherits from

[TCompoundPanel](#)

## Unit


[Stepper](#)


## Description

A step control component.

This help file was created with [HelpScribble](#).

## Properties

 Run-time only

 Key properties

 CaptionBitBtn1

 CaptionBitBtn2

 Max

 Min

 Size


 StepBy

 ValueOut

This help file was created with [HelpScribble](#).



## Events

 Key events

 [OnValueChanged](#)

This help file was created with [HelpScribble](#).

# CaptionBitBtn1 property

## Applies to

[TStepper](#) component

## Declaration

```
property CaptionBitBtn1: string;
```

## Description

The text on the downstepping (left) BitButton.

This help file was created with [HelpScribble](#).

# CaptionBitBtn2 property

## Applies to

[TStepper](#) component

## Declaration

```
property CaptionBitBtn2: string;
```

## Description

The text on the upstepping (right) BitButton.

This help file was created with [HelpScribble](#).

# Max property

## Applies to

[TStepper](#) component

## Declaration

```
property Max: integer;
```

## Description

The maximum-value of the control. Reaching this value disenables BitBtn2.

This help file was created with [HelpScribble](#).

# Min property

## Applies to

[TStepper](#) component

## Declaration

```
property Min: integer;
```

## Description

The minimum-value of the control. Reaching this value disenables BitBtn1.

This help file was created with [HelpScribble](#).

# Size property

## Applies to

[TStepper](#) component

## Declaration

**property** Size: [TStepperSize](#);

## Description

Sets the component size.

This help file was created with [HelpScribble](#).

# StepBy property

## Applies to

[TStepper](#) component

## Declaration

```
property StepBy: integer;
```

## Description

Sets the increase-, decrement-amount of TStepper.

This help file was created with [HelpScribble](#).

# ValueOut property

[Example](#)

## Applies to

[TStepper](#) component

## Declaration

```
property ValueOut: integer;
```

## Description

Read this value to determine the Stepper-value.

This help file was created with [HelpScribble](#).



## Example ValueOut

```
procedure TForm1.Stepper1ValueChanged(Sender: TObject);  
begin  
    Edit1.Text:=IntToStr(Stepper1.ValueOut);  
    ProgressBar1.Position:=Stepper1.ValueOut;  
end;
```

```
procedure TForm1.FormActivate(Sender: TObject);  
begin  
    Edit1.Text:=IntToStr(Stepper1.ValueOut);  
    ProgressBar1.Position:=Stepper1.ValueOut;  
end;
```

This help file was created with [HelpScribble](#).

# OnValueChanged event

## Applies to

[TStepper](#) component

## Declaration

```
property OnValueChanged: TNotifyEvent;
```

## Description

This event occurs when Valueout is changed by user.

This help file was created with [HelpScribble](#).

# TStepperSize type

## Unit

[stepper](#)

## Declaration

```
type TStepperSize = (ssSmall, ssNormal, ssBig);
```

## Description

Tree different sizes of TStepper: Small, normal and big.

This help file was created with [HelpScribble](#).

## passoverbtn unit

TPassOverBtn is a button that changes fontcolor and glyph when the mouse-pointer passes over it.

### Components

[TPassOverBtn](#)

### Routines

[Register](#)

This help file was created with [HelpScribble](#).



# **TPassOverBtn component**

[Properties](#)

[Events](#)

## **Inherits from**

[TBitBtn](#)

## **Unit**

[passoverbtn](#)

## **Inherits from**


TBitBtn


## **Description**


TPassOverBtn is a button that changes fontcolor and glyph when the mouse-pointer passes over it.

This help file was created with [HelpScribble](#).

## Properties

 Run-time only

 Key properties

 FontColorIn

 FontColorOut



 GlyphIn

 GlyphOut

This help file was created with [HelpScribble](#).

## Events

### Key events

-  [OnMouseOut](#)
-  [OnMouseOver](#)

This help file was created with [HelpScribble](#).

# FontColorIn property

## Applies to

[TPassOverBtn](#) component

## Declaration

```
property FontColorIn: TColor;
```

## Description

Fontcolor when mouse is over the control.

This help file was created with [HelpScribble](#).



# FontColorOut property

## Applies to

[TPassOverBtn](#) component

## Declaration

```
property FontColorOut: TColor;
```

## Description

Fontcolor when mouse is out of the control.

This help file was created with [HelpScribble](#).

# GlyphIn property

## Applies to

[TPassOverBtn](#) component

## Declaration

```
property GlyphIn: TBitmap;
```

## Description

Holds the Glyph to be shown when mouse is over the control.

This help file was created with [HelpScribble](#).

# GlyphOut property

## Applies to

[TPassOverBtn](#) component

## Declaration

```
property GlyphOut: TBitmap;
```

## Description

Holds the Glyph to be shown when mouse is out of the control.

This help file was created with [HelpScribble](#).

# OnMouseOut event

## Applies to

[TPassOverBtn](#) component

## Declaration

```
type TOnMouseOutEvent = procedure (Sender: TObject); of object;
```

## Description

Event raised when mouse passes out of button.

This help file was created with [HelpScribble](#).

# OnMouseOver event

## Applies to

[TPassOverBtn](#) component

## Declaration

```
type TOnMouseOverEvent = procedure (Sender: TObject); of object;
```

## Description

Event raised when mouse is over button.

This help file was created with [HelpScribble](#).

# Drivescan unit

This unit declares the TDriveScan-component.

A component that scans a whole device and outputs the directorynames and filenames in separate stringlists. Multiple file-filters can be set.

## Components

[TDriveScan](#)

## Types

[TFileType](#)

## Variables

[TDriveScanForm](#)

This help file was created with [HelpScribble](#).

# TDriveScanForm

## Unit

[Drivescan](#)

## Description

The wrapper-form used by TDriveScan.

This help file was created with [HelpScribble](#).



# TDriveScan component

[Properties](#)

[Methods](#)

[Events](#)

[Example](#)

## Inherits from

[TCompoundGroupBox](#)

## Unit

[Drivescan](#)

## Inherits from

[TCompoundGroupBox](#)


## Description


TDriveScan is a component that scans a whole device and outputs the directorynames and filenames in separate stringlists. Multiple file-filters can be set.

This help file was created with [HelpScribble](#).



## Properties

 Run-time only

 Key properties

 Dirlist

 Drive

 FileList

 FileListing

 FileType

 Filter




 Form

 Path

This help file was created with [HelpScribble](#).

## Methods

 Key methods





 [Execute](#)

 [Stop](#)

This help file was created with [HelpScribble](#).

## Events

### Key events

-  [OnDirectoryFound](#)
-  [OnDriveChanged](#)
-  [OnFileFound](#)
-  [OnReady](#)

This help file was created with [HelpScribble](#).

# Drive property

[Example](#)

**Applies to**  
[TDriveScan](#) component

**Declaration**  
`property Drive: char;`

**Description**  
Specifies the drive to scan.

This help file was created with [HelpScribble](#).

## TDriveScan example

```
procedure TForm1.DriveScan1DirectoryFound(Sender: TObject);
begin
  Caption:=DriveScan1.DirFound;
  RichEdit1.Lines.Add(DriveScan1.DirFound);
end;

procedure TForm1.DriveScan1Ready(Sender: TObject);
begin
  Label1.Caption:='Directories ('+IntToStr(DriveScan1.DirList.Count)+') :';
  Label2.Caption:='Pictures ('+IntToStr(DriveScan1.FileList.Count)+') :';
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  RichEdit1.Clear;
  RichEdit2.Clear;
  DriveScan1.Execute;
end;

procedure TForm1.FormActivate(Sender: TObject);
begin
  DriveScan1.Filter.Add('*.*');
  DriveScan1.Filter.Add('*.*');
  DriveScan1.Filter.Add('*.*');
  DriveScan1.Filter.Add('*.*');
  DriveScan1.DriveChanged(Sender);
end;

procedure TForm1.DriveScan1FileFound(Sender: TObject);
begin
  RichEdit2.Lines.Add(DriveScan1.FileFound);
end;

procedure TForm1.DriveScan1DriveChanged(Sender: TObject);
begin
  Button1.Caption:='Scan drive '+DriveScan1.Drive+'\';
end;
```

# DirList property

[Example](#)

## Applies to

[TDriveScan](#) component

## Declaration

```
property DirList: TStrings;
```

## Description

A stringlist updated with directorynames found.

This help file was created with [HelpScribble](#).

# FileListing property

[Example](#)

## Applies to

[TDriveScan](#) component

## Declaration

```
property FileListing: boolean;
```

## Description

If true the files scanned are put in Filelist.

If false only directories are scanned.

This help file was created with [HelpScribble](#).

# FileList property

[Example](#)

## Applies to

[TDriveScan](#) component

## Declaration

```
property FileList: TStrings;
```

## Description

A stringlist updated with filenames if Filelisting is true.

This help file was created with [HelpScribble](#).



# FileType property

[Example](#)

## Applies to

[TDriveScan](#) component

## Declaration

```
property FileType: TFileType;
```

## Description

<<< Description of FileType property >>>

This help file was created with [HelpScribble](#).

## FileType property example

This help file was created with [HelpScribble](#).

# Filter property

[Example](#)

## Applies to

[TDriveScan](#) component

## Declaration

```
property Filter: TStrings;
```

## Description

A stringlist that holds the strings to be searched for in filenames of files being scanned. Use wildcard (\*).

This help file was created with [HelpScribble](#).

# Form property

## Applies to

[TDriveScan](#) component

## Declaration

```
property Form: TDriveScanForm;
```

## Description

The wrapper form for TDriveScan.

Run-time only

This help file was created with [HelpScribble](#).

# Path property

[Example](#)

## Applies to

[TDriveScan](#) component

## Declaration

**property** Path: boolean;

## Description

If true the full filepath and filename are putted in Filelist.

If false only filenames are put in Filelist.

This help file was created with [HelpScribble](#).

# Execute method

[Example](#)

**Applies to**  
[TDriveScan](#) component

**Declaration**  
`procedure Execute;`

**Description**  
Call execute to start the scan.

This help file was created with [HelpScribble](#).

# Stop Method

[Example](#)

**Applies to**  
[TDriveScan](#) component

**Declaration**  
`procedure Stop;`

**Description**  
Call Stop to terminate the scan before it is finished.

This help file was created with [HelpScribble](#).

# OnDirectoryFound event

[Example](#)

## Applies to

[TDriveScan](#) component

## Declaration

```
type TOnDirectoryFoundEvent = procedure(Sender: TObject; DirFound: string); of  
object;
```

## Description

Occurs every time a new directory is found.

This help file was created with [HelpScribble](#).



# OnDriveChanged event

[Example](#)

## Applies to

[TDriveScan](#) component

## Declaration

```
type TOnDriveChangedEvent = procedure (Sender: TObject; Drive: char); of  
object;
```

## Description

Occurs every time the drive is changed.

This help file was created with [HelpScribble](#).

# OnFileFound event

[Example](#)

## Applies to

[TDriveScan](#) component

## Declaration

```
type TOnFileFoundEvent = procedure(Sender: TObject; FileFound: string); of  
object;
```

## Description

Occurs every time a file that matches the filter-setting is found.

This help file was created with [HelpScribble](#).

# OnReady event

[Example](#)

## Applies to

[TDriveScan](#) component

## Declaration

```
type TOnReadyEvent = procedure (Sender: TObject); of object;
```

## Description

Occurs when the entire scan is done.

This help file was created with [HelpScribble](#).

# TFileType type

## Unit

[Drivescan](#)

## Declaration

```
type TFileType = (faReadOnly, faHidden, faSysFile, faVolumeID, faDirectory,  
faArchive, faAnyFile);
```

## Description

<<< Description of TFileType type >>>

This help file was created with [HelpScribble](#).

# WKey unit

This unit declares the TWKey-component.  
A component that stops program execution, waiting for a key to be pressed.

## Components

[TWKey](#)

## Types

[TKeyIn](#)

This help file was created with [HelpScribble](#).



# TWKey component

[Properties](#)

[Methods](#)

## Inherits from

[TComponent](#)

## Unit


[wkey](#)


## Description

A component that stops program execution, waiting for a key to be pressed.

This help file was created with [HelpScribble](#).

## Properties

 Run-time only

 Key properties

 KeyToPress


 MessageColor


 MessagePosition

 ShowMessage

This help file was created with [HelpScribble](#).

## Methods

 Key methods

 Wait

This help file was created with [HelpScribble](#).



# KeyToPress property

## Applies to

[TWKey](#) component

## Declaration

**property** KeyToPress: [TKeyIn](#);

## Description

The key (a-z, AnyKey) that starts program execution after calling Wait-method.

This help file was created with [HelpScribble](#).

# MessageColor property

## Applies to

[TWKey](#) component

## Declaration

**property** MessageColor: TColor;

## Description

FaceColor of the message displayed if showmessage property is set to true.

This help file was created with [HelpScribble](#).

# MessagePosition property

## Applies to

[TWKey](#) component

## Declaration

**property** MessagePosition: TPosition;

## Description

Screen-position for message displayed if ShowMessage-property is set to true.

This help file was created with [HelpScribble](#).

# ShowMessage property

## Applies to

[TWKey](#) component

## Declaration

```
property ShowMessage: boolean;
```

## Description

If set to true a message is shown when Wait-method is called.

This help file was created with [HelpScribble](#).

# Wait method

## Applies to

[TWKey](#) component

## Declaration

```
procedure Wait;
```

## Description

Calling Wait stops program execution and waits for a key to be pressed.

This help file was created with [HelpScribble](#).

# TKeyIn type

## Unit

[wkey](#)

## Declaration

```
type TKeyIn = (AnyKey, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r,  
s, t, u, v, w, x, y, z);
```

## Description

Possible keys to start program execution.

This help file was created with [HelpScribble](#).

# PushBtn unit

This unit declares the TPushBtn-component.  
A push-button control.

## Components

[TPushBtn](#)

## Types

[TPBtnColor](#)

[TPBtnSize](#)

This help file was created with [HelpScribble](#).



# TPushBtn component

[Properties](#)

[Methods](#)

[Events](#)

## Inherits from

[TGraphicControl](#)

## Unit

[PushBtn](#)


## Description


A push-button control.


This help file was created with [HelpScribble](#).




## Properties

 Run-time only

 Key properties

 Color


 Size

 State

 BounceUp

This help file was created with [HelpScribble](#).

## Methods


 Key methods

[Create](#)

[Destroy](#)

This help file was created with [HelpScribble](#).

## Events

 Key events

 [OnPushBtnChanged](#)

This help file was created with [HelpScribble](#).

# Color property

## Applies to

[TPushBtn](#) component

## Declaration

```
property Color: TPBtnColor;
```

## Description

Sets the button-color. (pbRed, pbYellow, pbLime, pbBlue)

This help file was created with [HelpScribble](#).

# Size property

## Applies to

[TPushBtn](#) component

## Declaration

```
property Size: TPBtnSize;
```

## Description

Sets the button-size. (pbSmall, pbNormal, pbBig)

This help file was created with [HelpScribble](#).

# State property

## Applies to

[TPushBtn](#) component

## Declaration

```
property State: Boolean;
```

## Description

Returns True if PushBtn is pressed.

This help file was created with [HelpScribble](#).

# BounceUp property

## Applies to

[TPushBtn](#) component

## Declaration

```
property BounceUp: Boolean;
```

## Description

If set to true the pushbutton will return to released state after 0,5 sec when clicked.

This help file was created with [HelpScribble](#).

# OnPushBtnChanged event

## Applies to

[TPushBtn](#) component

## Declaration

```
type TOnPushBtnChangedEvent = procedure(Sender: TObject; State: boolean); of  
object;
```

## Description

Event that occurs when the PushBtn state is changed. Read State to determine if the button was pressed or released. True = pressed.

This help file was created with [HelpScribble](#).



# TPBtnColor type

## Unit

[PushBtn](#)

## Declaration

```
type TPBtnColor = (pbRed, pbYellow, pbLime, pbBlue);
```

## Description

Possible PushBtn-colors.

This help file was created with [HelpScribble](#).

# TPBtnSize type

## Unit

[PushBtn](#)

## Declaration

```
type TPBtnSize = (pbSmall, pbNormal, pbBig);
```

## Description

Possible PushBtn-sizes.

This help file was created with [HelpScribble](#).

## URLLabel unit

This unit declares the TTURLLabel-component.  
A clickable URL- or mail-Label.

### Components

[TURLLabel](#)

This help file was created with [HelpScribble](#).



# TURLLabel component

[Properties](#)

[Methods](#)

## Unit




[URLLabel](#)

## Description

A clickable URL- or mail-Label.


This help file was created with [HelpScribble](#).

## Properties

-  Run-time only
-  Key properties
-  [MailLabel](#)

This help file was created with [HelpScribble](#).

## Methods

 Key methods

[Create](#)

[Click](#)

This help file was created with [HelpScribble](#).

# MailLabel property

## Applies to

[TURRLabel](#) component

## Declaration

```
property MailLabel: boolean;
```

## Description

If set to true a 'mailto:' is added to caption, in order to launch mail.

If set to false a http:// adress is expected in the caption property.

This help file was created with [HelpScribble](#).

# RegistDays unit

[See also](#)

This unit declares the TRegistDays-component.  
A registration component.

## Components

[TRegistDays](#)

## Constants

FakeGUID

This help file was created with [HelpScribble](#).



## See also

[TRegistTimes](#)

This help file was created with [HelpScribble](#).



# TRegistDays component

[See also](#)

[Properties](#)

[Methods](#)

## Unit

[registdays](#)

## Description

A registration component.


This help file was created with [HelpScribble](#).


## See also


[TRegistTimes](#)

This help file was created with [HelpScribble](#).

## Properties

 Run-time only

 Key properties

 DaysLeft

 DaysToUse


 Password

 Registered

 RegNumber

This help file was created with [HelpScribble](#).

## Methods

 Key methods

[Create](#)

[Destroy](#)

 [CheckPassWord](#)

 [Init](#)

 [RegisterApp](#)

This help file was created with [HelpScribble](#).

# DaysLeft property

## Applies to

[TRegistDays](#) component

## Declaration

```
property DaysLeft: Integer;
```

## Description

The number of days left before registering the application is necessary.

This help file was created with [HelpScribble](#).

# DaysToUse property

{keepn}

## Applies to

[TRegistDays](#) component

## Declaration

**property** DaysToUse: Integer;

## Description

Specifies the number of days the application could be used without registering it.

This help file was created with [HelpScribble](#).

# Password property

## Applies to

[TRegistDays](#) component

## Declaration

```
property Password: string;
```

## Description

The string used when registering the application.

This help file was created with [HelpScribble](#).



# Registered property

## Applies to

[TRegistDays](#) component

## Declaration

`property Registered: Boolean;`

## Description

True if application was successfully registered.

This help file was created with [HelpScribble](#).

# RegNumber property

## Applies to

[TRegistDays](#) component

## Declaration

```
property RegNumber: Integer;
```

## Description

A number used for Key-name and Key-data entries in Registry.

Randomly (1-999999999) generated at component creation, or chosen by user.

This help file was created with [HelpScribble](#).

# CheckPassWord method

## Applies to

[TRegistDays](#) component

## Declaration

```
function CheckPassWord(Value: string): Boolean;
```

## Description

Function that checks the input string against the [PassWord](#) property.  
Returns true if equal.

This help file was created with [HelpScribble](#).

# Init method

## Applies to

[TRegistDays](#) component

## Declaration

```
function Init: Integer;
```

## Description

Function that should be called each time the application is started.

Call if from the MainForm OnCreate event.

Returns [DaysLeft](#)

This help file was created with [HelpScribble](#).

# RegisterApp method

## Applies to

[TRegistDays](#) component

## Declaration

```
procedure RegisterApp;
```

## Description

Registers the application.

Sets Key-name and Key-data equal to [RegNumber](#) in Registry.

This help file was created with [HelpScribble](#).

# RegistTimes unit

[See also](#)

This unit declares the TRegistTimes-component.  
A registration component.

## Components

[TRegistTimes](#)

## Constants

FakeGUID

This help file was created with [HelpScribble](#).

## See also

[TRegistDays](#)

This help file was created with [HelpScribble](#).



# TRegistTimes component

[See also](#)

[Properties](#)

[Methods](#)

## Unit

[registtimes](#)

## Description

A registration component.

This help file was created with [HelpScribble](#).





## See also


[TRegistDays](#)

This help file was created with [HelpScribble](#).

## Properties

 Run-time only

 Key properties

 Password

 Registered


 RegNumber

 TimesLeft

 TimesToUse

This help file was created with [HelpScribble](#).

## Methods

 Key methods

[Create](#)

[Destroy](#)

 [CheckPassWord](#)

 [Init](#)

 [RegisterApp](#)

This help file was created with [HelpScribble](#).

# Password property

## Applies to

[TRegistTimes](#) component

## Declaration

```
property Password: string;
```

## Description

The string used when registering the application.

This help file was created with [HelpScribble](#).

# Registered property

## Applies to

[TRegistTimes](#) component

## Declaration

**property** Registered: Boolean;

## Description

True if application was successfully registered.

This help file was created with [HelpScribble](#).

# RegNumber property

## Applies to

[TRegistTimes](#) component

## Declaration

```
property RegNumber: Integer;
```

## Description

A number used for Key-name and Key-data entries in Registry.

Randomly (1-999999999) generated at component creation, or chosen by user.

This help file was created with [HelpScribble](#).

# TimesLeft property

## Applies to

[TRegistTimes](#) component

## Declaration

```
property TimesLeft: Integer;
```

## Description

The number of times left before registering is necessary.

This help file was created with [HelpScribble](#).

# TimesToUse property

## Applies to

[TRegistTimes](#) component

## Declaration

```
property TimesToUse: Integer;
```

## Description

Specifies the number of times the application could be used without registering it.

This help file was created with [HelpScribble](#).



# CheckPassWord method

## Applies to

[TRegistTimes](#) component

## Declaration

```
function CheckPassWord(Value: string): Boolean;
```

## Description

Function that checks the input string against the [PassWord](#). Returns true if equal.

This help file was created with [HelpScribble](#).

# Init method

## Applies to

[TRegistTimes](#) component

## Declaration

```
function Init: integer;
```

## Description

Function that should be called each time the application is run.

Call if from the MainForm.OnCreate event.

Decrements [TimesLeft](#) and returns [TimesLeft](#).

This help file was created with [HelpScribble](#).

# RegisterApp method

## Applies to

[TRegistTimes](#) component

## Declaration

```
procedure RegisterApp;
```

## Description

Registers the application.

Sets Key-name and Key-data equal to [RegNumber](#).

This help file was created with [HelpScribble](#).

# ExecApp unit

This unit declares the TExecApp component and the TFileNameProperty-type.

## Components

[TExecApp](#)

## Types

[TFileNameProperty](#)

This help file was created with [HelpScribble](#).

# TFileNameProperty

[Methods](#)

## Unit


[ExecApp](#)

## Description

An enhancement to the Object-Inspector. When selecting a file for the FileName-property you can browse to find it.

This help file was created with [HelpScribble](#).

## Methods

 Key methods

[GetAttributes](#)

[Edit](#)

This help file was created with [HelpScribble](#).



# TExecApp component

[Properties](#)

[Methods](#)

## Unit

[ExecApp](#)




## Description

A speedbutton-type component.

When you click the button the application with the name in FileName is launched.

This help file was created with [HelpScribble](#).


## Properties

-  Run-time only
-  Key properties
-  FileName

This help file was created with [HelpScribble](#).



## Methods

 Key methods

[Create](#)

[Destroy](#)

[Click](#)

This help file was created with [HelpScribe](#).

# FileName property

## Applies to

[TExecApp](#) component

## Declaration

**property** FileName: [TFileName](#);

## Description

The complete path and filename for the application thats going to be executed.

This help file was created with [HelpScribble](#).

# mLED unit

## Components

[TmLED](#)

## Types

[TLEDColor](#)

[TLEDSize](#)

[TLEDState](#)

This help file was created with [HelpScribble](#).



# TmLED component

[Properties](#)

[Methods](#)

## Unit


[mLED](#)


## Description


A LED-imitation component with four colors and three sizes.


This help file was created with [HelpScribble](#).

## Properties

 Run-time only

 Key properties


 Color

 Size

 State

This help file was created with [HelpScribble](#).

## Methods

 Key methods

[Create](#)

[Destroy](#)

This help file was created with [HelpScribble](#).

# Color property

## Applies to

[TmLED](#) component

## Declaration

**property** Color: [TLEDColor](#);

## Description

Sets the LED-color. Four colors available: red, yellow, green and blue.

This help file was created with [HelpScribble](#).

# Size property

## Applies to

[TmLED](#) component

## Declaration

**property** Size: [TLEDSize](#);

## Description

Sets the LED-size. Three sizes available: small, normal and big.

This help file was created with [HelpScribble](#).



# State property

## Applies to

[TmLED](#) component

## Declaration

**property** State: [TLEDState](#);

## Description

Puts the LED in ON- or OFF-state.

This help file was created with [HelpScribble](#).

# TLEDColor type

## Unit

[mLED](#)

## Declaration

```
type TLEDColor = (lcYellow, lcRed, lcGreen, lcBlue);
```

This help file was created with [HelpScribble](#).

# TLEDSize type

## Unit

mLED

## Declaration

```
type TLEDSize = (lzSmall, lzNormal, lzBig);
```

This help file was created with [HelpScribble](#).

# TLEDState type

## Unit

mLED

## Declaration

```
type TLEDState = (lsON, lsOFF);
```

This help file was created with [HelpScribble](#).

# ToggleSwitch unit

## Components

[IToggleForm](#)

[IToggleSwitch](#)

## Classes

[THintStrings](#)

## Types

[IPositionType](#)

[IStepType](#)

This help file was created with [HelpScribble](#).

# TToggleForm

## Unit

[ToggleSwitch](#)

## Description

The container-Form for TToggleSwitch component.

This help file was created with [HelpScribble](#).

# THintStrings class

[Properties](#)

## Unit


[ToggleSwitch](#)


## Description

Holds the hintstrings to is to be shown in the different switchpositions.

This help file was created with [HelpScribble](#).

## Properties

 Run-time only

 Key properties

 HintPos1

 HintPos2

 HintPos3

 HintPos4

 HintPos5

 HintPos6

 HintPos7

 HintPos8

This help file was created with [HelpScribble](#).



# HintPos1 property

## Applies to

[THintStrings](#) class

## Declaration

```
property HintPos1: string;
```

## Description

The hintstring for switchposition 1.

This help file was created with [HelpScribble](#).

# HintPos2 property

## Applies to

[THintStrings](#) class

## Declaration

```
property HintPos2: string;
```

## Description

The hintstring for switchposition 2.

This help file was created with [HelpScribble](#).

## HintPos3 property

### Applies to

[THintStrings](#) class

### Declaration

```
property HintPos3: string;
```

### Description

The hintstring for switchposition 3.

This help file was created with [HelpScribble](#).

## HintPos4 property

### Applies to

[THintStrings](#) class

### Declaration

```
property HintPos4: string;
```

### Description

The hintstring for switchposition 4.

This help file was created with [HelpScribble](#).

## HintPos5 property

### Applies to

[THintStrings](#) class

### Declaration

```
property HintPos5: string;
```

### Description

The hintstring for switchposition 5.

This help file was created with [HelpScribble](#).

## HintPos6 property

### Applies to

[THintStrings](#) class

### Declaration

```
property HintPos6: string;
```

### Description

The hintstring for switchposition 6.

This help file was created with [HelpScribble](#).

## HintPos7 property

### Applies to

[THintStrings](#) class

### Declaration

```
property HintPos7: string;
```

### Description

The hintstring for switchposition 7.

This help file was created with [HelpScribble](#).

# HintPos8 property

## Applies to

[THintStrings](#) class

## Declaration

```
property HintPos8: string;
```

## Description

The hintstring for switchposition 8.

This help file was created with [HelpScribble](#).





# TToggleSwitch component

[Properties](#)

[Methods](#)

[Events](#)

## Unit


[ToggleSwitch](#)


## Description

A toggle-switch control.

This help file was created with [HelpScribble](#).

## Properties

 Run-time only

 Key properties

 HintList


 Position

 Steps

 TurnBack


This help file was created with [HelpScribble](#).

## Methods

 Key methods  
[Toggle](#)

This help file was created with [HelpScribe](#).

## Events

 Key events

 [OnToggle](#)

This help file was created with [HelpScribble](#).

# HintList property

## Applies to

[IToggleSwitch](#) component

## Declaration

```
property HintList: THintStrings;
```

## Description

Holds the hintstrings to be shown in the different switchpositions.

This help file was created with [HelpScribble](#).

# Position property

## Applies to

[IToggleSwitch](#) component

## Declaration

**property** Position: [TPositionType](#);

## Description

The switch-position.

This help file was created with [HelpScribble](#).

# Steps property

## Applies to

[IToggleSwitch](#) component

## Declaration

**property** Steps: [IStepType](#);

## Description

Sets number of switch-steps.

This help file was created with [HelpScribble](#).

# TurnBack property

## Applies to

[IToggleSwitch](#) component

## Declaration

```
property TurnBack: Boolean;
```

## Description

If set to true the switch change rotation-direction after reaching end-position, otherwise switch changes to first position when reaching end-position.

This help file was created with [HelpScribble](#).



# Toggle method

## Applies to

[IToggleSwitch](#) component

## Declaration

```
procedure Toggle;
```

## Description

Call Toggle to programatically toggle the switch.

This help file was created with [HelpScribble](#).

# OnToggle event

## Applies to

[IToggleSwitch](#) component

## Declaration

```
type TOnToggleEvent = procedure(Sender: TObject; Position: TPositionType); of  
object;
```

## Description

Occurs when the switch is clicked or programatically toggled.  
The new position is in Position var.

This help file was created with [HelpScribble](#).

# TPositionType type

## Unit

[ToggleSwitch](#)

## Declaration

```
type TPositionType = 1..8;
```

This help file was created with [HelpScribble](#).

# TStepType type

## Unit

[ToggleSwitch](#)

## Declaration

```
type TStepType = 2..8;
```

This help file was created with [HelpScribble](#).

# HPCounter unit

## Components

[THPCounter](#)

## Types

[TInt64](#)

This help file was created with [HelpScribble](#).



# THPCounter component

[Methods](#)

## Unit


[HPCounter](#)

## Description

A high-precision counter/timer. Retrieves time differences down to 1 microsec.

This help file was created with [HelpScribble](#).

## Methods

 Key methods

[Create](#)

[Destroy](#)

 [Read](#)

 [ReadInt](#)

 [Start](#)

This help file was created with [HelpScribble](#).

# Read method

## Applies to

[THPCounter](#) component

## Declaration

```
function Read: string;
```

## Description

Reads the counter-value as a string. Place this call just after the code you want to measure.

This help file was created with [HelpScribble](#).



# ReadInt method

## Applies to

[THPCounter](#) component

## Declaration

```
function ReadInt: TLargeInteger;
```

## Description

Reads the counter-value as an Int64. Place this call just after the code you want to measure.

This help file was created with [HelpScribble](#).

# Start method

## Applies to

[THPCounter](#) component

## Declaration

```
procedure Start;
```

## Description

Starts the counter. Place this call just before the code you want to measure.

This help file was created with [HelpScribble](#).

# TInt64 type

## Unit

[HPCounter](#)

## Declaration

```
type TInt64 = TLargeInteger;
```

This help file was created with [HelpScribble](#).

# RText unit

## Components

[TRText](#)

## Types

[TDirection](#)

This help file was created with [HelpScribble](#).



# TRText component

[Properties](#)

[Methods](#)

## Unit

[RTText](#)

## Description

A rotating-text label. Marquee-type.

This help file was created with [HelpScribble](#).

## Properties

- ▶ Run-time only
- ▶ Key properties
  - ▶ Delay
    - ▶ Direction
    - ▶ Space

This help file was created with [HelpScribble](#).

## Methods

### ▶ Key methods

- ▶ [Create](#)
- ▶ [Destroy](#)
- ▶ [Start](#)
- ▶ [Stop](#)

This help file was created with [HelpScribble](#).

# Delay property

## Applies to

[IRText](#) component

## Declaration

```
property Delay: integer;
```

## Description

Sets the delay for each movement in msec.

This help file was created with [HelpScribble](#).



# Direction property

## Applies to

[IRText](#) component

## Declaration

**property** Direction: [TDirection](#);

## Description

Sets the rotate-direction.

This help file was created with [HelpScribble](#).

# Space property

## Applies to

[IRText](#) component

## Declaration

```
property Space: integer;
```

## Description

Sets number of spaces between caption-strings.

This help file was created with [HelpScribble](#).

## Start method

### Applies to

[TRText](#) component

### Declaration

```
procedure Start;
```

### Description

Starts the rotation.

This help file was created with [HelpScribble](#).

# Stop method

## Applies to

[TRText](#) component

## Declaration

```
procedure Stop;
```

## Description

Stops the rotation.

This help file was created with [HelpScribble](#).

# TDirection type

## Unit

[RText](#)

## Declaration

```
type TDirection = (LeftToRight, RightToLeft);
```

This help file was created with [HelpScribble](#).

# hllabel unit

## Components

[THLabel](#)

## Types

[LType](#)

This help file was created with [HelpScribble](#).



# THLLLabel component

[Properties](#)

[Events](#)

## Unit

[hllabel](#)

## Description

A clickable URL- ,Mail- or Exe-Label. Changes fontcolor when mouse pass over.

This help file was created with [HelpScribble](#).

## Properties

- ▶ Run-time only
- ▶ Key properties
  - ▶ [FontColorIn](#)
    - ▶ [FontColorOut](#)
    - ▶ [LType](#)

This help file was created with [HelpScribble](#).



## Events

### ▶ Key events

- ▶ [OnMouseOut](#)
- ▶ [OnMouseOver](#)

This help file was created with [HelpScribble](#).

# FontColorIn property

## Applies to

[THLLabel](#) component

## Declaration

```
property FontColorIn: TColor;
```

## Description

FontColor when mouse-pointer is over label.

This help file was created with [HelpScribble](#).

# FontColorOut property

## Applies to

[THLLabel](#) component

## Declaration

```
property FontColorOut: TColor;
```

## Description

FontColor when mouse-pointer is out of label.

This help file was created with [HelpScribble](#).

# LType property

## Applies to

[THLLabel](#) component

## Declaration

```
property LType: TLType;
```

## Description

Four Label-types available:

**Exe-Label:** Runs an executable when clicked.

**URL-Label:** A 'http://' is added to caption, in order to launch the Web-Browser.

**Mail-Label:** A 'mailto:' is added to caption, in order to launch the mail-program.

**Normal-Label:** Non-clickable. For use when only colorchange is wanted.

# OnMouseOut event

## Applies to

[THLLabel](#) component

## Declaration

```
type TOnMouseOutEvent = procedure (Sender: TObject); of object;
```

## Description

Event raised when mouse-pointer is out of label.

This help file was created with [HelpScribble](#).

# OnMouseOver event

## Applies to

[THLLabel](#) component

## Declaration

```
type TOnMouseOverEvent = procedure (Sender: TObject); of object;
```

## Description

Event raised when mouse-pointer is over label.

This help file was created with [HelpScribble](#).

# TlType type

## Unit

[hllabel](#)

## Declaration

```
type TlType = (Normal, URL, Mail, Exe);
```

This help file was created with [HelpScribble](#).

# TimePick unit

## Components

[TTimePick](#)

## Variables

[TCForm](#)

This help file was created with [HelpScribble](#).



# TCForm

## Unit

[TimePick](#)

## Description

The container-form for TTimePick component.

This help file was created with [HelpScribble](#).



# TTimePick component

[Properties](#)

[Events](#)

## Unit

[TimePick](#)

## Description

A component for picking time-values.

This help file was created with [HelpScribble](#).

## Properties

- ▶ Run-time only
- ▶ Key properties
  - ▶ [Hour](#)
  - ▶ [Min](#)
  - ▶ [RoundTheClock](#)
  - ▶ [Time](#)

This help file was created with [HelpScribble](#).

## Events

▶ Key events



[OnTimeSet](#)

This help file was created with [HelpScribble](#).

# Hour property

## Applies to

[ITimePick](#) component

## Declaration

```
property Hour: string;
```

## Description

Read-only

This help file was created with [HelpScribble](#).

# Min property

## Applies to

[ITimePick](#) component

## Declaration

```
property Min: string;
```

## Description

Read-only

This help file was created with [HelpScribble](#).

# RoundTheClock property

## Applies to

[ITimePick](#) component

## Declaration

```
property RoundTheClock: Boolean;
```

## Description

If set to false the time-incrementation stops when reaching 23:59.

This help file was created with [HelpScribble](#).

# Time property

## Applies to

[ITimePick](#) component

## Declaration

```
property Time: string;
```

## Description

Read-only

This help file was created with [HelpScribble](#).



# OnTimeSet event

## Applies to

[TTimePick](#) component

## Declaration

```
type TOnTimeSetEvent = procedure (Sender: TObject; Time1: TTime; Time2, Hour,  
Min: string); of object;
```

## Description

Occurs when the hour- or minute-values are changed.

This help file was created with [HelpScribble](#).

## mWImage unit

A component with six different animated images to be used when your application is doing something that's worth waiting for. Wheel-, clock-, compress-, sleep-, typewriter- and pen- imagestyles.

### Components

[TmWImage](#)

### Types

[TImageType](#)

This help file was created with [HelpScribble](#).



# TmWImage component

[Properties](#)

[Methods](#)

## Unit

[mWImage](#)

## Description

A component with six different animated images to be used when your application is doing something that's worth waiting for. Wheel-, clock-, compress-, sleep-, typewriter- and pen- imagestyles.

This help file was created with [HelpScribble](#).

## Properties

- ▶ Run-time only
- ▶ Key properties
  - ▶ [ImageType](#)
    - ▶ [Interval](#)

This help file was created with [HelpScribble](#).

## Methods

### ▶ Key methods

- ▶ [Create](#)
- ▶ [Destroy](#)
- ▶ [Start](#)
- ▶ [Stop](#)

This help file was created with [HelpScribble](#).

# ImageType property

## Applies to


[TmWImage](#) component


## Declaration


```
property ImageType: TImageType;
```

## Description


The animated image to be shown:

itWheel - 

itClock - 

itCompress - 

itSleep - 

itTypeWriter - 

itPen - 

This help file was created with [HelpScribble](#).

# Interval property

## Applies to

[TmWImage](#) component

## Declaration

```
property Interval: Integer;
```

## Description

The image-shift interval in msec.

This help file was created with [HelpScribble](#).

## Start method

### Applies to

[TmWImage](#) component

### Declaration

```
procedure Start;
```

### Description

Starts the animation.

This help file was created with [HelpScribble](#).



# Stop method

## Applies to

[TmWImage](#) component

## Declaration

```
procedure Stop;
```

## Description

Stops the animation.

This help file was created with [HelpScribble](#).

# TImageType type

## Unit

[mWImage](#)

## Declaration

```
type TImageType = (itWheel, itClock, itCompress, itSleep, itTypeWriter,  
itPen);
```

This help file was created with [HelpScribble](#).

# mShortcut unit

This component produces windows-shortcuts.

## Components

[TmShortcut](#)

## Types

[TShortcutDest](#)

This help file was created with [HelpScribble](#).



# TmShortcut component

[Properties](#)

[Methods](#)

## Unit

[mShortcut](#)

## Description

This component produces windows-shortcuts.

This help file was created with [HelpScribble](#).

## Properties

- ▶ Run-time only
- ▶ Key properties
  - ▶ Destination
    - ▶ ExeParameters
    - ▶ FullExePath
    - ▶ ShortcutCaption
    - ▶ ShortcutFolder

This help file was created with [HelpScribble](#).

## Methods

### ▶ Key methods

- ▶ [Create](#)
- ▶ [Destroy](#)
- ▶ [Execute](#)

This help file was created with [HelpScribe](#).

# Destination property

## Applies to

[TmShortcut](#) component

## Declaration

**property** Destination: TShortcutDest;

## Description

Where the shortcut should be put.

sdDeskTop - On desktop

sdStartMenu - In startmenu

sdFolder - In a specific folder

sdAll - On desktop, in startmenu, and in a specific folder.

This help file was created with [HelpScribble](#).

# ExeParameters property

## Applies to

[TmShortcut](#) component

## Declaration

```
property ExeParameters: string;
```

## Description

If you want to send inparameters to the target application, then put them in here with <space> inbetween.

This help file was created with [HelpScribble](#).



# FullExePath property

## Applies to

[TmShortcut](#) component

## Declaration

```
property FullExePath: TFileName;
```

## Description

The path + filename of the target application.

This help file was created with [HelpScribble](#).

# ShortcutCaption property

## Applies to

[TmShortcut](#) component

## Declaration

```
property ShortcutCaption: string;
```

## Description

The shortcut-name.

This help file was created with [HelpScribble](#).

# ShortcutFolder property

## Applies to

[TmShortcut](#) component

## Declaration

```
property ShortcutFolder: string;
```

## Description

The folder in which the shortcut should be put. (If Destination is sdFolder or sdAll)

This help file was created with [HelpScribble](#).

# Execute method

## Applies to

[TmShortcut](#) component

## Declaration

```
procedure Execute;
```

## Description

Creates the shortcut.

This help file was created with [HelpScribble](#).

# TShortcutDest type

## Unit

[mShortcut](#)

## Declaration

```
type TShortcutDest = (sdDeskTop, sdStartMenu, sdFolder, sdAll);
```

This help file was created with [HelpScribble](#).

## mtagproducer unit

A component that generates HTML-Formtags. To be used in a WebModule or with WebDisPatcher. The Content this component produces could easily be added to the DisPatchers ResponseContent.

### Components

[TmTagProducer](#)

### Types

[TChecked](#)

[TTagType](#)

This help file was created with [HelpScribble](#).



# TmTagProducer component

[Properties](#)

[Methods](#)

## Unit

[mtagproducer](#)

## Description

A component that generates HTML-Formtags. To be used in a WebModule or with WebDisPatcher. The Content this component produces could easily be added to the DisPatchers ResponseContent.



**TagType:** mhtCheckBox

**HtmlCode:** <INPUT TYPE="CHECKBOX" NAME="mTagProducer1" VALUE="Value1" CHECKED="CHECKED">

**TagType:** mhtDropDownList

**HtmlCode:** <SELECT NAME="mTagProducer1" SIZE="1"><OPTION>Option1</OPTION><OPTION>Option2 <OPTION></SELECT>

**TagType:** mhtFileUpload

**HtmlCode:** <INPUT TYPE="FILE" NAME="mTagProducer1" SIZE="10">

**TagType:** mhtHR

**HtmlCode:** <HR WIDTH="100" SIZE="3" COLOR="#FF0000">



**TagType:** mhtImageButton

**HtmlCode:** <INPUT TYPE="IMAGE" NAME="mTagProducer1" SRC="Image1.bmp" WIDTH="50" HEIGHT="50">

**TagType:** mhtListBox

**HtmlCode:** <SELECT NAME="mTagProducer1" SIZE="2"><OPTION>Option1</OPTION><OPTION>Option2</OPTION></SELECT>



**TagType:** mhtOptionButton

**HtmlCode:** <INPUT TYPE="RADIO" NAME="mTagProducer1" VALUE="Value1" CHECKED="CHECKED">

**TagType:** mhtPassWord

**HtmlCode:** <INPUT TYPE="PASSWORD" NAME="mTagProducer1" VALUE="Value1" SIZE="20">

**TagType:** mhtPushButton

**HtmlCode:** <INPUT TYPE="BUTTON" NAME="mTagProducer1" VALUE="Push">

**TagType:** mhtResetButton

**HtmlCode:** <INPUT TYPE="RESET" NAME="mTagProducer1" VALUE="Reset">

**TagType:** mhtSubmitButton

**HtmlCode:** <INPUT TYPE="SUBMIT" NAME="mTagProducer1" VALUE="Submit">

**TagType:** mhtTextArea

**HtmlCode:** <TEXTAREA NAME="mTagProducer1" ROWS="2" COLS="20"></TEXTAREA>

**TagType:** mhtTextBox

**HtmlCode:** <INPUT TYPE="TEXT" NAME="mTagProducer1" VALUE="Value1" SIZE="20">



## Properties

- ▶ Run-time only
- ▶ Key properties
  - ▶ Checked
    - ▶ Cols
    - ▶ Content
    - ▶ Name
    - ▶ OptionName
    - ▶ Rows
    - ▶ Size
    - ▶ SRC
    - ▶ TagType
    - ▶ Value
    - ▶ Color
    - ▶ NoOfOptions
  - ▶ Width

This help file was created with [HelpScribble](#).

## Methods

### ▶ Key methods

[Create](#)

[Destroy](#)

This help file was created with [HelpScribble](#).

# Checked property

## Applies to

[TmTagProducer](#) component

## Declaration

**property** Checked: TChecked;

## Description

HTML-param. used in TagType: OptionButton.

This help file was created with [HelpScribble](#).

# Cols property

## Applies to

[TmTagProducer](#) component

## Declaration

```
property Cols: integer;
```

## Description

HTML-param. used in TagType: TextArea.

This help file was created with [HelpScribble](#).

# Content property

## Applies to

[TmTagProducer](#) component

## Declaration

```
property Content: string;
```

## Description

The resulting HTML-tag.

This help file was created with [HelpScribble](#).

# Name property

## Applies to

[TmTagProducer](#) component

## Declaration

**property** Name: TComponentName;

## Description

Name of the component. Used in the HTML-Tag as well.

This help file was created with [HelpScribble](#).

# OptionName property

## Applies to

[TmTagProducer](#) component

## Declaration

```
property OptionName: string;
```

## Description

HTML-param. used in TagTypes: DropDownList and ListBox.

This help file was created with [HelpScribble](#).

# Rows property

## Applies to

[TmTagProducer](#) component

## Declaration

```
property Rows: integer;
```

## Description

HTML-param. used in TagType: TextArea.

This help file was created with [HelpScribble](#).



# Size property

## Applies to

[TmTagProducer](#) component

## Declaration

```
property Size: integer;
```

## Description

HTML-param. used in TagType: ListBox and DropDownList (Size always = 1).

This help file was created with [HelpScribble](#).

# SRC property

## Applies to

[TmTagProducer](#) component

## Declaration

```
property SRC: TFileName;
```

## Description

HTML-param. used in TagType: ImageButton. The ImageFileName.

This help file was created with [HelpScribble](#).

# TagType property

## Applies to

[TmTagProducer](#) component

## Declaration

**property** TagType: TTagType

## Description

Depending of value of this property, HTML are generated for:  
PushButton, CheckBox, FileUpload, ImageButton, OptionButton, ResetButton, SubmitButton, TextBox,  
TextArea, DropDownList and ListBox.

This help file was created with [HelpScribble](#).

# Value property

## Applies to

[TmTagProducer](#) component

## Declaration

```
property Value: string;
```

## Description

HTML-param. used in TagTypes: PushButton, CheckBox, ResetButton, SubmitButton and OptionButton.

This help file was created with [HelpScribble](#).

# TChecked type

## Unit

[mtagproducer](#)

## Declaration

```
type TChecked = (mchChecked, mchUnChecked);
```

This help file was created with [HelpScribble](#).

# TTagType type

## Unit

[mtagproducer](#)

## Declaration

```
type TTagType = (mhtPushButton, mhtCheckBox, mhtFileUpload, mhtImageButton,  
mhtOptionButton, mhtResetButton, mhtSubmitButton, mhtTextBox, mhtTextArea,  
mhtDropDownList, mhtListBox);
```

This help file was created with [HelpScribble](#).

# Color property

## Applies to

[TmTagProducer](#) component

## Declaration

```
property Value: string;
```

## Description

HTML-param. used in TagType: HR.

This help file was created with [HelpScribble](#).

# NoOfOptions property

## Applies to

[TmTagProducer](#) component

## Declaration

`property` Value: `integer`;

## Description

Number of options when TagType = DropDownList or ListBox.

This help file was created with [HelpScribble](#).



# Width property

## Applies to

[TmTagProducer](#) component

## Declaration

```
property Value: string;
```

## Description

HTML-param. used in TagType: HR.

This help file was created with [HelpScribble](#).

# mp3Tag unit

A component for reading and writing mp3-tags.

## Components

[Imp3Tag](#)

This help file was created with [HelpScribble](#).



# Tmp3Tag component

[Properties](#)

[Methods](#)

## Unit

[mp3Tag](#)

## Description

A component for reading and writing mp3-tags.

This help file was created with [HelpScribble](#).

## Properties

- ▶ Run-time only
- ▶ Key properties

- ▶ Album

- ▶ Artist
- ▶ Comment
- ▶ Genre
- ▶ GenreID
- ▶ Genres
- ▶ TagModified
- ▶ TagPresent
- ▶ Title
- ▶ Track
- ▶ v1Tag
- ▶ Year

This help file was created with [HelpScribble](#).

## Methods

### ▶ Key methods

- ▶ [Create](#)
- ▶ [Destroy](#)
- ▶ [DeleteTagFromFile](#)
- ▶ [LoadTag](#)
- ▶ [LoadTagFromFile](#)
- ▶ [LoadTagFromStream](#)
- ▶ [SaveTag](#)
- ▶ [SaveTagToFile](#)
- ▶ [SaveTagToStream](#)

This help file was created with [HelpScribble](#).

# Album property

## Applies to

[Imp3Tag](#) component

## Declaration

```
property Album: string;
```

## Description

The Album title.

This help file was created with [HelpScribble](#).

# Artist property

## Applies to

[Imp3Tag](#) component

## Declaration

```
property Artist: string;
```

## Description

The Artist

This help file was created with [HelpScribble](#).

# Comment property

## Applies to

[Imp3Tag](#) component

## Declaration

```
property Comment: string;
```

## Description

Any comments made.

This help file was created with [HelpScribble](#).



# Genre property

[See also](#)

## Applies to

[Timp3Tag](#) component

## Declaration

```
property Genre: string;
```

## Description

The music Genre. Could be one of the following (in GenreId-order):

Blues, Classic Rock, Country, Dance, Disco, Funk, Grunge, Hip-Hop, Jazz, Metal, New Age, Oldies, Other, Pop, R&B, Rap, Reggae, Rock, Techno, Industrial, Alternative, Ska, Death Metal, Pranks, Soundtrack, Euro-Techno, Ambient, Trip-Hop, Vocal, JazzFunk, Fusion, Trance, Classical, Instrumental, Acid, House, Game, Sound Clip, Gospel, Noise, AlternRock, Bass, Soul, Punk, Space, Meditative, Instrumental Pop, Instrumental Rock, Ethnic, Gothic, Darkwave, Techno-Industrial, Electronic, Pop-Folk, Eurodance, Dream, Southern Rock, Comedy, Cult, Gangsta, Top 40, Christian Rap, Pop/Funk, Jungle, Native American, Cabaret, New Wave, Psychedelic, Rave, Showtunes, Trailer, Lo-Fi, Tribal, Acid Punk, Acid Jazz, Polka, Retro, Musical, Rock & Roll, Hard Rock, Folk, Folk/Rock, National Folk, Swing, Fast Fusion, Bebob, Latin, Revival, Celtic, Bluegrass, Avantgarde, Gothic Rock, Progressive Rock, Psychedelic Rock, Symphonic Rock, Slow Rock, Big Band, Chorus, Easy Listening, Acoustic, Humour, Speech, Chanson, Opera, Chamber Music, Sonata, Symphony, Booty Bass, Primus, Porn Groove, Satire, Slow Jam, Club, Tango, Samba, Folklore, Ballad, Power Ballad, Rhythmic Soul, Freestyle, Duet, Punk Rock, Drum Solo, Acapella, Euro-House, Dance Hall, Goa, Drum & Bass, Club-House, Hardcore, Terror, Indie, BritPop, Negerpunk, Polsk Punk, Beat, Christian Gangs, Heavy Metal, Black Metal, Crossover, Contemporary Ch?, Cristian Rock, Merengue, Salsa, Thrash Metal, Anime, JPop, Synthpop

## See also

- ▶ [Genre](#)
- ▶ [GenreID](#)
- ▶ [Genres](#)

This help file was created with [HelpScribble](#).

# GenreID property

[See also](#)

## Applies to

[Tmp3Tag](#) component

## Declaration

```
property GenreID: Integer;
```

## Description

The GenreID. An integer-value between 0 and 157.

This help file was created with [HelpScribble](#).

# Genres property

[See also](#)

## Applies to

[Imp3Tag](#) component

## Declaration

**property** Genres: TStringList;

## Description

A stringlist with all possible music-genres. Read-Only.

This help file was created with [HelpScribble](#).

# TagModified property

## Applies to

[Imp3Tag](#) component

## Declaration

```
property TagModified: Boolean;
```

## Description

Set to true if Tag changed.

Read-only

This help file was created with [HelpScribble](#).

# TagPresent property

## Applies to

[Imp3Tag](#) component

## Declaration

**property** TagPresent: Boolean;

## Description

True if loaded tag seems to be OK.

This help file was created with [HelpScribble](#).

# Title property

## Applies to

[Imp3Tag](#) component

## Declaration

```
property Title: string;
```

## Description

The song Title.

This help file was created with [HelpScribble](#).

# Track property

## Applies to

[Imp3Tag](#) component

## Declaration

**property** Track: Integer;

## Description

The Track number.

This help file was created with [HelpScribble](#).



# v1Tag property

## Applies to

[Imp3Tag](#) component

## Declaration

```
property v1Tag: Boolean;
```

## Description

True if ID3v1, false if ID3v1.1

This help file was created with [HelpScribble](#).

# Year property

## Applies to

[Imp3Tag](#) component

## Declaration

```
property Year: string;
```

## Description

Year as a 4-char string.

This help file was created with [HelpScribble](#).

# DeleteTagFromFile method

## Applies to

[Imp3Tag](#) component

## Declaration

```
procedure DeleteTagFromFile(const aFileName: string);
```

## Description

Delete tag from file (if it exists).

This help file was created with [HelpScribble](#).

# LoadTag method

## Applies to

[Imp3Tag](#) component

## Declaration

```
procedure LoadTag(const ABuf; ABufSz: Integer);
```

## Description

Load tag from buffer (max ABufSz).

This help file was created with [HelpScribble](#).

# LoadTagFromFile method

[See also](#)

[Example](#)

## Applies to

[Imp3Tag](#) component

## Declaration

```
procedure LoadTagFromFile(const aFileName: string);
```

## Description

Load ID3Tag from file AFileName.

This help file was created with [HelpScribble](#).

## See also

- ▶ [SaveTagToFile](#)

This help file was created with [HelpScribble](#).

## LoadTagFromFile method example

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
  if OpenFileDialog1.Execute then  
    begin  
      mp3Tag1.LoadTagFromFile(OpenDialog1.FileName);  
      Edit1.Text:= mp3Tag1.Title;  
      Edit2.Text:= mp3Tag1.Artist;  
      Edit3.Text:= mp3Tag1.Album;  
      Edit4.Text:= IntToStr(mp3Tag1.Track);  
      Edit5.Text:= mp3Tag1.Year;  
      Edit6.Text:= mp3Tag1.Genre;  
      Edit7.Text:= mp3Tag1.Comment;  
      MediaPlayer1.FileName:= OpenFileDialog1.FileName;  
      MediaPlayer1.Open;  
    end;  
end;
```

This help file was created with [HelpScribble](#).

# LoadTagFromStream method

## Applies to

[Imp3Tag](#) component

## Declaration

```
procedure LoadTagFromStream(AStream: TStream; AStreamSz: Integer);
```

## Description

Find ID3tag in AStream (max AStreamSz bytes from current position) and Load it. If success then returns offset to tag, else -1.

This help file was created with [HelpScribble](#).



# SaveTag method

## Applies to

[Imp3Tag](#) component

## Declaration

```
procedure SaveTag(var ABuf; ABufSz: Integer);
```

## Description

Write tag into ABuf (max ABufSz bytes), clear Modified.

This help file was created with [HelpScribble](#).

# SaveTagToFile method

[See also](#)

[Example](#)

## Applies to

[Imp3Tag](#) component

## Declaration

```
procedure SaveTagToFile(const aFileName: string);
```

## Description

Add/Update tag in file AFileName, clear Modified.

This help file was created with [HelpScribble](#).

## See also

- ▶ [LoadTagFromFile](#)

This help file was created with [HelpScribble](#).

## SaveTagToFile method example

```
procedure TForm1.Button2Click(Sender: TObject);  
begin  
  if MediaPlayer1.FileName = "" then  
    begin  
      MessageDlg("There"s no mp3-file loaded !", mtWarning, [mbOK], 0);  
      Exit;  
    end;  
    mp3Tag1.Title:= Edit1.Text;  
    mp3Tag1.Artist:= Edit2.Text;  
    mp3Tag1.Album:= Edit3.Text;  
    mp3Tag1.Track:= StrToInt(Edit4.Text);  
    mp3Tag1.Year:= Edit5.Text;  
    mp3Tag1.Genre:= Edit6.Text;  
    mp3Tag1.Comment:= Edit7.Text;  
    MediaPlayer1.Close;  
    mp3Tag1.SaveTagToFile(OpenDialog1.FileName);  
    MediaPlayer1.FileName:= OpenDialog1.FileName;  
    MediaPlayer1.Open;  
end;
```

This help file was created with [HelpScribble](#).

# SaveTagToStream method

## Applies to

[Imp3Tag](#) component

## Declaration

**procedure** SaveTagToStream(AStream: TStream);

## Description

Write tag ito current position of AStream, clear Modified.

This help file was created with [HelpScribble](#).

## mRegApp unit

This registration-component let you decide how many days your application is to be used, before the user have to register. Keeps track of clock-manipulations, and is farely hard to break.  
Requires Delphi 6.

### Components

[TmRegApp](#)

### Types

[TDType](#)

### Constants

RegKey

This help file was created with [HelpScribble](#).



# TmRegApp component

[Properties](#)

[Methods](#)

## Unit

[mRegApp](#)

## Description

This registration-component let you decide how many days your application is to be used, before the user have to register. Keeps track of clock-manipulations, and is farely hard to break.

Requires Delphi 6.

This help file was created with [HelpScribble](#).

## Properties

- ▶ Run-time only
- ▶ Key properties
  - ▶ CodeKey
    - ▶ DaysToUse
    - ▶ FakeGUID
    - ▶ P

This help file was created with [HelpScribble](#).



## Methods

### ▶ Key methods

[Create](#)

[Destroy](#)

### ▶ [CheckPassWord](#)

▶ [ExpireDate](#)

▶ [Init](#)

▶ [InstallationDate](#)

▶ [LastUseDate](#)

▶ [MoveBack](#)

▶ [ClearReg](#)

This help file was created with [HelpScribble](#).

# CodeKey property

## Applies to

[TmRegApp](#) component

## Declaration

```
property CodeKey: string;
```

## Description

Key used for coding/decoding. Change this to anything you like.

This help file was created with [HelpScribble](#).

# DaysToUse property

## Applies to

[TmRegApp](#) component

## Declaration

```
property DaysToUse: Integer;
```

## Description

The number of days before registering is needed.

This help file was created with [HelpScribble](#).

# FakeGUID property

## Applies to

[TmRegApp](#) component

## Declaration

```
property FakeGUID: string;
```

## Description

A false GUID used as key in Registry.

You should change this, otherwise people with knowledge of this component could find it.

This help file was created with [HelpScribble](#).

# P property

## Applies to

[TmRegApp](#) component

## Declaration

```
property P: string;
```

## Description

The password. Automatically coded. Otherwise it could be read in the resource part of the exe-file.

Set password at designtime and remember it. A dialog showing the coded value will show up.

This help file was created with [HelpScribble](#).

# CheckPassWord method

## Applies to

[TmRegApp](#) component

## Declaration

```
function CheckPassWord(PassWord: string): boolean;
```

## Description

Returns true if password is correct (same as in Registry).  
If so the Registry-values: 'DaysToUse', 'Expire', 'LastUse' will be deleted from Registry. The value: 'Registered' will be set true.

This help file was created with [HelpScribble](#).

# ExpireDate method

## Applies to

[TmRegApp](#) component

## Declaration

```
function ExpireDate: string;
```

## Description

Returns Expiredate as a string.

This help file was created with [HelpScribble](#).

# Init method

## Applies to

[TmRegApp](#) component

## Declaration

```
function Init: Integer;
```

## Description

This method should be called each time the application starts executing.

Returns the days left before registrering is needed.

If the app. is already registered -1 is returned.

This help file was created with [HelpScribble](#).



# InstallationDate method

## Applies to

[TmRegApp](#) component

## Declaration

```
function InstallationDate: string;
```

## Description

Returns Installationdate as a string.

This help file was created with [HelpScribble](#).

# LastUseDate method

## Applies to

[TmRegApp](#) component

## Declaration

```
function LastUseDate: string;
```

## Description

Returns the date of last use as a string.

This help file was created with [HelpScribble](#).

# MoveBack method

## Applies to

[TmRegApp](#) component

## Declaration

```
function MoveBack: boolean;
```

## Description

Call this function to see if the user has manipulated the clock.

If true, the user has moved the clock back more than one day compared with the last use date. When calling the Init function 0 will be returned, and will continue to do so until registered.

This help file was created with [HelpScribble](#).

# ClearReg method

## Applies to

[TmRegApp](#) component

## Declaration

```
procedure ClearReg;
```

## Description

Deletes all Registry-entries made.  
(Deletes the FakeGUID-key from Registry.)

This help file was created with [HelpScribble](#).

# TDType type

## Unit

[mRegApp](#)

## Declaration

```
type TDType = (dtInstall, dtLastUse, dtExpire);
```

This help file was created with [HelpScribble](#).

## mDataSetTableProducer unit

This is a TDataSetTableProducer with a pagecontrol-extension. Set the MaxRows property to the number of lines you want to show for each webpage, and the pagecontrol gives you the opportunity to choose the First-Previous-Next or Last page. Just one web-action needed. Uses the Get-method.

### Components

[TmDataSetTableProducer](#)

### Types

[TPosType](#)

This help file was created with [HelpScribble](#).



## TmDataSetTableProducer component

[Properties](#)

[Methods](#)

[Events](#)

[Example](#)

### Unit

[mDataSetTableProducer](#)

### Description

This is a TDataSetTableProducer with a pagecontrol-extension. Set the MaxRows property to the number of lines you want to show for each webpage, and the pagecontrol gives you the opportunity to choose the First-Previous-Next or Last page. Just one web-action needed. Uses the Get-method.

This help file was created with [HelpScribble](#).

## Properties

- ▶ Run-time only
- ▶ Key properties
  - [Caption](#)
  - [CaptionAlignment](#)
  - [Columns](#)
  - [DataSet](#)
  - [Footer](#)
  - [Header](#)
  - [MaxRows](#)
  - ▶ [PageControl](#)
    - ▶ [PathInfo](#)
    - ▶ [Position](#)
    - [RowAttributes](#)
  - [TableAttributes](#)

This help file was created with [HelpScribble](#).



## Methods

### ▶ Key methods

[Create](#)

[Destroy](#)

[Content](#)

### ▶ [GetRecIndex](#)

▶ [RollDataSet](#)

This help file was created with [HelpScribble](#).

## Events

### ▶ Key events

[OnCreateContent](#)

[OnFormatCell](#)

[OnGetTableCaption](#)

This help file was created with [HelpScribe](#).

## TmDataSetTableProducer - Example

Shows the component being used in a webaction procedure.

```
procedure TWebModule1.WebModule1WebActionItem1Action(Sender: TObject;  
  Request: TWebRequest; Response: TWebResponse; var Handled: Boolean);  
begin  
  mDataSetTableProducer1.PathInfo:= '/gettable';  
  
  Query1.Close;  
  Query1.SQL.Text:= 'select * from animals';  
  Query1.Open;  
  
  // Get the ReclIndex from Request and position cursor to ReclIndex  
  mDataSetTableProducer1.RollDataSet(mDataSetTableProducer1.GetReclIndex(Request));  
  
  Response.Content:= mDataSetTableProducer1.Content;  
end;
```

This help file was created with [HelpScribble](#).

# PageControl property

## Applies to

[TmDataSetTableProducer](#) component

## Declaration

**property** PageControl: boolean;

## Description

Enables/disables the pagecontrol.

This help file was created with [HelpScribble](#).

# PathInfo property

## Applies to

[TmDataSetTableProducer](#) component

## Declaration

```
property PathInfo: string;
```

## Description

Indicates the path portion of the Uniform Resource Identifier (URI) for which the action item handles requests.

This help file was created with [HelpScribble](#).

# Position property

## Applies to

[TmDataSetTableProducer](#) component

## Declaration

**property** Position: TPosType;

## Description

The pagecontrol position. Over or under the dbtable.

This help file was created with [HelpScribble](#).

# GetRecIndex method

## Applies to

[TmDataSetTableProducer](#) component

## Declaration

```
function GetRecIndex(WebRequest: TWebRequest): integer;
```

## Description

Reads the last recordindex from the calling webpage.

This help file was created with [HelpScribble](#).

# RollDataSet method

## Applies to

[TmDataSetTableProducer](#) component

## Declaration

```
procedure RollDataSet(RecIndex: Integer);
```

## Description

Sets the dataset cursor position to ReclIndex.

This help file was created with [HelpScribble](#).



# TPosType type

## Unit

[mDataSetTableProducer](#)

## Declaration

```
type TPosType = (ipTop, ipBottom);
```

This help file was created with [HelpScribble](#).



## mGetNewVer unit

With this component in your application, your users would always be updated with the latest program release. When you have a new version coming up, just update a file (infile-syntax), on your webserver with the new version number. When the user starts your application, this component will look in that file. If updating is needed, the user will be asked to download the new version.

### Components

[TmGetNewVer](#)

### Types

[PTransBuffer](#)

[TOnGetURLEvent](#)

[TTransBuffer](#)



# TmGetNewVer component

[Properties](#)

[Methods](#)

[Events](#)

## Unit

[mGetNewVer](#)

## Description

With this component in your application, your users would always be updated with the latest program release. When you have a new version coming up, just update a file (infile-syntax), on your webserver with the new version number. When the user starts your application, this component will look in that file. If updating is needed, the user will be asked to download the new version.

This help file was created with [HelpScribble](#).

## Properties

- ▶ Run-time only
- ▶ Key properties
  - ▶ [DownloadDlg](#)
    - ▶ [FailureDlg](#)
    - ▶ [SiteName](#)
    - ▶ [TempFile](#)
    - ▶ [URLCurrVerFile](#)

This help file was created with [HelpScribble](#).

## Methods

- ▶ Key methods
  - [Create](#)
  - [Destroy](#)
  - ▶ [Execute](#)

This help file was created with [HelpScribble](#).

## Events

- ▶ Key events
  - ▶ [OnGetURL](#)

This help file was created with [HelpScribble](#).

# DownloadDlg property

## Applies to

[TmGetNewVer](#) component

## Declaration

**property** DownloadDlg: Boolean;

## Description

If true a download-dialog is shown to user.

This help file was created with [HelpScribble](#).

# FailureDlg property

## Applies to

[TmGetNewVer](#) component

## Declaration

**property** FailureDlg: Boolean;

## Description

If true and if there is difficulties downloading version-information, a dialog is shown to user informing that no version information could be retrieved.

This help file was created with [HelpScribble](#).



# SiteName property

## Applies to

[TmGetNewVer](#) component

## Declaration

```
property SiteName: string;
```

## Description

Name of the site. Used in the download-dialog.

This help file was created with [HelpScribble](#).

# TempFile property

## Applies to

[TmGetNewVer](#) component

## Declaration

**property** TempFile: TFileName;

## Description

A file for temporarily storage of the current version file.

This help file was created with [HelpScribble](#).

# URLCurrVerFile property

## Applies to

[TmGetNewVer](#) component

## Declaration

```
property URLCurrVerFile: string;
```

## Description

The URL to the file where the current version information is located.

This file should be of type IniFile, with two sections **Currentversions** and **DownloadURL**.

The file syntax is

[Currentversions]

<application title> = <version number>

.....

[DownloadURL]

<application title> = <The URL to the application>

.....

FileExample:

[Currentversions]

compmaker=3.7.0.0

winfiles=1.1.0.0

[DownloadURL]

compmaker=http://213.65.224.213/dl/app/compmaker.zip

winfiles=http://213.65.224.213/dl/app/winfiles.zip

This help file was created with [HelpScribble](#).

# Execute method

## Applies to

[TmGetNewVer](#) component

## Declaration

```
procedure Execute;
```

## Description

Starts downloading the version-info.

This help file was created with [HelpScribble](#).

# OnGetURL event

## Applies to

[TmGetNewVer](#) component

## Declaration

**property** OnGetURL: [TOnGetURLEvent](#);

## Description

Event fired when the versioninfo has been read.

This help file was created with [HelpScribble](#).

# PTransBuffer type

## Unit

[mGetNewVer](#)

## Declaration

```
type PTransBuffer = ^TTransBuffer;
```

This help file was created with [HelpScribble](#).

# TOnGetURLEvent type

## Unit

[mGetNewVer](#)

## Declaration

```
type TOnGetURLEvent = procedure(Sender: TObject; CurrentVersion, DownloadURL:  
string); of object;
```

This help file was created with [HelpScribble](#).

# TTransBuffer type

## Unit

[mGetNewVer](#)

## Declaration

```
type TTransBuffer = array[1..4] of smallint;
```

This help file was created with [HelpScribble](#).



## mThumbs unit

A component that produces thumbnail images from jpeg image files. Two different image modes: Crop and Fill. Handles multiple sourcefiles. Fileprefix-, fillcolor-, height-, width-, zoom-property. OnImageCreate event.

### Components

[TmThumbs](#)

### Types

[TMode](#)

[TOnThumbCreatedEvent](#)

[TZoom](#)

This help file was created with [HelpScribble](#).



# TmThumbs component

[Properties](#)

[Methods](#)

[Events](#)

## Unit

[mThumbs](#)

## Description

A component that produces thumbnail images from jpeg image files. Two different image modes: Crop and Fill. Handles multiple sourcefiles. Fileprefix-, fillcolor-, height-, width-, zoom-property. OnImageCreate event.

This help file was created with [HelpScribble](#).

## Properties

- ▶ Run-time only
- ▶ Key properties
  - ▶ FilePrefix
    - ▶ FillColor
    - ▶ Height
    - ▶ InFiles
    - ▶ Mode
    - ▶ OutputDir
    - ▶ Width
    - ▶ ZoomOut

This help file was created with [HelpScribble](#).

## Methods

### ▶ Key methods

[Create](#)

[Destroy](#)

▶ [CreateThumbnails](#)

This help file was created with [HelpScribble](#).

## Events

### ▶ Key events

#### ▶ [OnThumbCreated](#)

This help file was created with [HelpScribble](#).

# FilePrefix property

## Applies to

[ImThumbs](#) component

## Declaration

```
property FilePrefix: string;
```

## Description

Fileprefix for the thumbnail files.

This help file was created with [HelpScribble](#).

# FillColor property

## Applies to

[TmThumbs](#) component

## Declaration

```
property FillColor: TColor;
```

## Description

The fillcolor used when Mode = thFill.

This help file was created with [HelpScribble](#).

# Height property

## Applies to

[TmThumbs](#) component

## Declaration

```
property Height: integer;
```

## Description

The thumbnail height in pixels.

This help file was created with [HelpScribble](#).



# InFiles property

## Applies to

[TmThumbs](#) component

## Declaration

**property** InFiles: TStrings;

## Description

A stringlist with full path filenames to the source images.

This help file was created with [HelpScribble](#).

# Mode property

## Applies to

[TmThumbs](#) component

## Declaration

**property** Mode: [TMode](#);

## Description

thCrop: The thumbnails height and width is calculated from the ZoomOut value.

Height and Width does nothing and no FillColor is needed.

thFill: Height and Width determines the thumbnail size,

FillColor is used to fill remaining area, if nescesary.

This help file was created with [HelpScribble](#).

# OutputDir property

## Applies to

[ImThumbs](#) component

## Declaration

```
property OutputDir: string;
```

## Description

The directory where the thumbnailfiles is to be stored.

This help file was created with [HelpScribble](#).

# Width property

## Applies to

[TmThumbs](#) component

## Declaration

```
property Width: integer;
```

## Description

The thumbnail width.

This help file was created with [HelpScribble](#).

# ZoomOut property

## Applies to

[ImThumbs](#) component

## Declaration

**property** ZoomOut: [TZoom](#);

## Description

The thumbnail size in percent of the source image size.

This help file was created with [HelpScribble](#).

# CreateThumbnails method

## Applies to

[TmThumbs](#) component

## Declaration

**procedure** CreateThumbnails;

## Description

Creates the thumbnail(-s).

This help file was created with [HelpScribble](#).

# OnThumbCreated event

## Applies to

[ImThumbs](#) component

## Declaration

**property** OnThumbCreated: [TOnThumbCreatedEvent](#);

## Description

Fired when a thumbnail file is created.

Delivers: The thumbimage itself, the thumb filename, the imagecount and the total imagecount for the session.

This help file was created with [HelpScribble](#).

# TMode type

## Unit

[mThumbs](#)

## Declaration

```
type TMode = (thFill, thCrop);
```

This help file was created with [HelpScribble](#).



# TOnThumbCreatedEvent type

## Unit

[mThumbs](#)

## Declaration

```
type TOnThumbCreatedEvent = procedure (FileName: string; Count, Total: integer;  
Image: TJpegImage); of object;
```

This help file was created with [HelpScribble](#).

# TZoom type

## Unit

[mThumbs](#)

## Declaration

```
type TZoom = 1..100;
```

This help file was created with [HelpScribble](#).

# MASUtils Code Library

## Routines

### String handling

[CutOutStrs](#)

[Like function](#)

[HexToInt](#)

[CheckIfHex](#)

[IsDigit](#)

[IsLower](#)

[IsUpper](#)

[Proper](#)

[ToLower](#)

[ToUpper](#)

[CodeSting](#)

[DecodeString](#)

[LoadFile](#)

[SaveFile](#)

### Files

[GetAssociation](#)

[FindFile](#)

[GetCreationDate](#)

[GetVersionInfo](#)

[GetLastWriteDate](#)

[GetSize](#)

[RecycleFile](#)

### Time

[NowToMDPDate](#)

### Internet

[EMailValid](#)

### Graphics

[ConvertToJpg](#)

**Mats Asplund/MAs Prod.**

This help file was created with [HelpScribble](#).

# HexToInt function

## Unit

[masutils](#)

## Declaration

```
function HexToInt(HexStr: string): Integer;
```

## Description

Converts the hexadecimal value (0-9,A-F) in HexStr to an integer.  
Returns -1 if conversion didn't succeed.

This help file was created with [HelpScribble](#).

# CutOutStrs procedure

## Unit

[masutils](#)

## Declaration

```
procedure CutOutStrs(Str, DelCh: string; var SStrs: TStringList);
```

## Description

Use CutOutStrs to extract substrings from a string using delimiters.

Str is the string itself. DelCh is the delimiter-character.

The result is returned in SStrs.

This help file was created with [HelpScribble](#).

# FindFile function

## Unit

[masutils](#)

## Declaration

```
function FindFile(const filespec: TFileName): TStringList;
```

## Description

Looks for files specified in filespec. Scans subdirectorys as well.

## Example

Find all .exe-files and write them to Memo1.

```
procedure TForm1.Button1Click(Sender: TObject);  
var List: TStringList;  
begin  
    List:= TStringList.Create;  
    List:= FindFile('* .exe');  
    Memo1.Text:= List.Text;  
    List.Free;  
end;
```

This help file was created with [HelpScribble](#).

# GetAssociation function

## Unit

[masutils](#)

## Declaration

```
function GetAssociation(const FileName: string): string;
```

## Description

This function retrieves the full pathname for the application associated with FileName.

This help file was created with [HelpScribble](#).

# Like function

## Unit

[masutils](#)

## Declaration

```
function Like(AString, Pattern: string; CaseSensitive: boolean): boolean;
```

## Description

This function compares a string with a pattern, and returns true if there is a match.

Use '?' for one "don't care" - character, and '\*' for one or more "don't care" - characters.

## Example

Like('Delphi', '?e?p\*', false) will return true.

This help file was created with [HelpScribble](#).



# IsDigit function

## Unit

[masutils](#)

## Declaration

```
function IsDigit(ch: char): boolean;
```

## Description

Returns true if ch is a digit ('0'-'9').

This help file was created with [HelpScribble](#).

# IsLower function

## Unit

[masutils](#)

## Declaration

```
function IsLower(ch: char): boolean;
```

## Description

Returns true if ch is a lowercase character.

This help file was created with [HelpScribble](#).

# IsUpper function

## Unit

[masutils](#)

## Declaration

```
function IsUpper(ch: char): boolean;
```

## Description

Returns true if ch is a uppercase character.

This help file was created with [HelpScribble](#).

# Proper function

## Unit

[masutils](#)

## Declaration

```
function Proper(const s: string): string;
```

## Description

Capitalizes first letter of every word in s.

This help file was created with [HelpScribble](#).

# ToLower function

## Unit

[masutils](#)

## Declaration

```
function ToLower(ch: char): char;
```

## Description

Changes a character to a lowercase letter.

This help file was created with [HelpScribble](#).

# ToUpper function

## Unit

[masutils](#)

## Declaration

```
function ToUpper(ch: char): char;
```

## Description

Changes a character to an uppercase letter.

This help file was created with [HelpScribble](#).

# CheckIfHex function

## Unit

[masutils](#)

## Declaration

```
function CheckIfHex(Str: string; ShowMsg: boolean): boolean;
```

## Description

The function returns true if all characters in Str is hexadecimal ('0'..'9', 'A'..'F', 'a'..'f'). If ShowMsg is true a message will show, indicating which character-positions that is wrong.

This help file was created with [HelpScribble](#).

# GetCreationDate function

## Unit

[masutils](#)

## Declaration

```
function GetCreationDate(FileName: string): TDateTime;
```

## Description

This function returns the creationdate of a file.

This help file was created with [HelpScribble](#).



# GetVersionInfo procedure

## Unit

[masutils](#)

## Declaration

```
procedure GetVersionInfo(AVersionList: TStrings);
```

## Description

This procedure retrieves the application's versioninfo.

Reading AVersionList returns the following:

AVersionList[0] - FileVersion

AVersionList[1] - LegalCopyright

AVersionList[2] - ProductName

AVersionList[3] - ProductVersion

This help file was created with [HelpScribble](#).

# GetLastWriteDate function

## Unit

[masutils](#)

## Declaration

```
function GetLastWriteDate(FileName: string): TDateTime;
```

## Description

This function returns last write date of a file.

This help file was created with [HelpScribble](#).

# NowToMDPDate function

## Unit

[masutils](#)

## Declaration

```
function NowToMDPDate: string;
```

## Description

This function returns today's date in the form: "<month> <day>, <year>".  
Ex. Sep 21, 2001

This help file was created with [HelpScribble](#).

# CodeString function

[See also](#)

## Unit

[masutils](#)

## Declaration

```
function CodeString(UnCodedString: string): string;
```

## Description

Returns a coded string. The coded string can be read with the DecodeString function.

This help file was created with [HelpScribble](#).

## See also

[DecodeString](#)

This help file was created with [HelpScribble](#).

# DecodeString function

[See also](#)

## Unit

[masutils](#)

## Declaration

```
function DecodeString(CodedString: string): string;
```

## Description

Decodes a string that was coded with the CodeString function.

This help file was created with [HelpScribble](#).

## See also

[CodeString](#)

This help file was created with [HelpScribble](#).

# EEmailValid function

## Unit

[masutils](#)

## Declaration

```
function EEmailValid(EMail: string): boolean;
```

## Description

Returns true if the EEmailAddress is valid.

This help file was created with [HelpScribble](#).



# GetSize function

## Unit

[masutils](#)

## Declaration

```
function GetSize(FileName: string): cardinal;
```

## Description

Returns the filesize in bytes.

This help file was created with [HelpScribble](#).

# LoadFile function

[See also](#)

## Unit

[masutils](#)

## Declaration

```
function LoadFile(const FileName: TFileName): string;
```

## Description

Loads a file content into a string.

This help file was created with [HelpScribble](#).

## See also

[SaveFile](#)

This help file was created with [HelpScribble](#).

# RecycleFile function

## Unit

[masutils](#)

## Declaration

```
function RecycleFile(FileName: string): Boolean;
```

## Description

Sends a file to the Recycle bin. Returns true if succeeded.

This help file was created with [HelpScribble](#).

# ConvertToJpeg procedure

## Unit

[masutils](#)

## Declaration

**procedure** ConvertToJpeg (FileNameBitmap: TFileName);

## Description

Converts a bitmap-file to a Jpg imagefile.

Ex.

```
ConvertToJpg('c:\Image.bmp');
```

The resulting file will be named: c:\Image.jpg

This help file was created with [HelpScribble](#).

# SaveFile procedure

[See also](#)

## Unit

[masutils](#)

## Declaration

```
procedure SaveFile(const FileName: TFileName; const Content: string);
```

## Description

Saves a string (Content) to a file (FileName).

This help file was created with [HelpScribble](#).

## See also

[LoadFile](#)

This help file was created with [HelpScribble](#).

## mFileScan unit

A component that scans a whole device or part of it, and outputs the fullpath filenames, corresponding to the searchfilter settings. Multiple filters can be set.

### Components

[TmFileScan](#)

### Types

[TOnFileFoundEvent](#)

[TOnReadyEvent](#)

This help file was created with [HelpScribble](#).





# TmFileScan component

[Properties](#)

[Methods](#)

[Events](#)

## Unit

[mFileScan](#)

## Description

A component that scans a whole device or part of it, and outputs the fullpath filenames, corresponding to the searchfilter settings. Multiple filters can be set.

This help file was created with [HelpScribble](#).

## Properties

- ▶ Run-time only
- ▶ Key properties
  - ▶ [FilePath](#)
    - ▶ [SearchFilter](#)
    - ▶ [SubDirectories](#)

This help file was created with [HelpScribble](#).

## Methods

### ▶ Key methods

[Create](#)

[Destroy](#)

▶ [Start](#)

▶ [Stop](#)

This help file was created with [HelpScribble](#).

## Events

### ▶ Key events

#### ▶ [OnFileFound](#)

#### ▶ [OnReady](#)

This help file was created with [HelpScribble](#).

# FilePath property

## Applies to

[TmFileScan](#) component

## Declaration

```
property FilePath: string;
```

## Description

Which filepath to scan.

This help file was created with [HelpScribble](#).

# SearchFilter property

## Applies to

[TmFileScan](#) component

## Declaration

```
property SearchFilter: TStrings;
```

## Description

The filesearch routes. Use wildcard. Multiple filters can be used.

## Example:

```
mFileScan1.FilePath:= 'c:\';  
mFileScan1.SubDirectories:= true;  
mFileScan1.SearchFilter.Add('a*.exe');
```

This will search for all exe-files on drive c:, beginning with the letter 'a'.

This help file was created with [HelpScribble](#).

# SubDirectories property

## Applies to

[TmFileScan](#) component

## Declaration

```
property SubDirectories: Boolean;
```

## Description

If true, subdirectories will be scanned.

This help file was created with [HelpScribble](#).

## Start method

### Applies to

[TmFileScan](#) component

### Declaration

```
procedure Start;
```

### Description

Starts the filescan.

This help file was created with [HelpScribble](#).



# Stop method

## Applies to

[TmFileScan](#) component

## Declaration

```
procedure Stop;
```

## Description

Stops the filescan.

This help file was created with [HelpScribble](#).

# OnFileFound event

## Applies to

[TmFileScan](#) component

## Declaration

**property** OnFileFound: [TOnFileFoundEvent](#);

## Description

Event fired each time a file is found. The FileName is returned.

This help file was created with [HelpScribble](#).

# OnReady event

## Applies to

[TmFileScan](#) component

## Declaration

**property** OnReady: [TOnReadyEvent](#);

## Description

Event fired when the entire scan is ready. Returns the searchresult in a StringList.

This help file was created with [HelpScribble](#).

# TOnFileFoundEvent type

## Unit

[mFileScan](#)

## Declaration

```
type TOnFileFoundEvent = procedure (FileName: string); of object;
```

This help file was created with [HelpScribble](#).

# TOnReadyEvent type

## Unit

[mFileScan](#)

## Declaration

```
type TOnReadyEvent = procedure (SearchResult: TStringList); of object;
```

This help file was created with [HelpScribble](#).

## AppDlg unit

TAppDlg is a component that uses an about-form for displaying application-info to user. AppName, Version and Date is fetched automatic. Site- and EMail-labels are clickable links. Uses [TURLLabel](#) and [TMDP](#).

### Components

[TAppDlg](#)

### Types

[PTransBuffer](#)

[TTransBuffer](#)

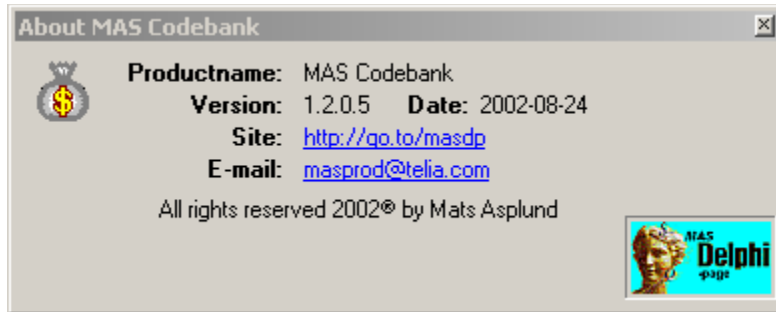
This help file was created with [HelpScribble](#).



# TAppDlg component

[Properties](#)

[Methods](#)



## Unit

[AppDlg](#)

## Description

TAppDlg is a component that uses an about-form for displaying application-info to user. AppName, Version and Date is fetched automatic. Site- and EMail-labels are clickable links.

Uses [TURLLabel](#) and [TMDP](#).

This help file was created with [HelpScribble](#).

## Properties

- ▶ Run-time only
- ▶ Key properties
  - ▶ [Email](#)
    - ▶ [Info](#)
    - ▶ [Productname](#)
    - ▶ [Site](#)
    - ▶ [Version](#)

This help file was created with [HelpScribble](#).



## Methods

- ▶ Key methods
  - [Create](#)
  - [Destroy](#)
  - ▶ [Execute](#)

This help file was created with [HelpScribe](#).

# Email property

## Applies to

[TAppDlg](#) component

## Declaration

```
property Email: string;
```

This help file was created with [HelpScribble](#).

# Info property

## Applies to

[TAppDlg](#) component

## Declaration

```
property Info: string;
```

This help file was created with [HelpScribble](#).

# Productname property

## Applies to

[TAppDlg](#) component

## Declaration

```
property Productname: string;
```

This help file was created with [HelpScribble](#).

# Site property

## Applies to

[TAppDlg](#) component

## Declaration

```
property Site: string;
```

This help file was created with [HelpScribble](#).

# Version property

## Applies to

[TAppDlg](#) component

## Declaration

```
property Version: string;
```

This help file was created with [HelpScribble](#).

# Execute method

## Applies to

[TAppDlg](#) component

## Declaration

```
function Execute: boolean;
```

## Description

Starts the Dialog.

This help file was created with [HelpScribble](#).

# PTransBuffer type

## Unit

[AppDlg](#)

## Declaration

```
type PTransBuffer = ^TTransBuffer;
```

This help file was created with [HelpScribble](#).



# TTransBuffer type

## Unit

[AppDlg](#)

## Declaration

```
type TTransBuffer = array[1..4] of smallint;
```

This help file was created with [HelpScribble](#).

# MDP unit



## Components

TMDP

## Variables

CompForm

This help file was created with [HelpScribble](#).



# TMDP component

[Methods](#)



**Unit**

[MDP](#)

This help file was created with [HelpScribble](#).

## Methods

### ▶ Key methods

[Create](#)

[Destroy](#)

▶ [Start](#)

▶ [Stop](#)

This help file was created with [HelpScribble](#).

# Start method

**Applies to**  
[TMDP](#) component

**Declaration**  
`procedure Start;`

This help file was created with [HelpScribble](#).

# Stop method

## Applies to

[TMDP](#) component

## Declaration

```
procedure Stop;
```

This help file was created with [HelpScribble](#).

## HelpScribble

HelpScribble is a help authoring tool written by Jan Goyvaerts. This help file was created with the unregistered version of HelpScribble, which is why you can read this ad. Once the author of this help file is so honest to register the shareware he uses, you will not see this ad again in his help files.

**Recompiling the help project with the registered version is all it takes to get rid of this ad.**

HelpScribble is a stand-alone help authoring tool. It does *not* require an expensive word processor. (Only a help compiler as Microsoft likes keeping the .hlp format secret. Not my fault.)

Here are some of HelpScribble's features:

- The Setup program will *properly* install and uninstall HelpScribble and all of its components, including registry keys.
- Create, edit and navigate through topics right in the main window. No need to mess with heaps of dialog boxes.
- All topics are listed in a grid in the main window so you won't lose track in big help projects. You can even set bookmarks.
- Use the built-in Browse Sequence Editor to easily create browse sequences.
- Use the built-in Window Editor to change the look of your help window and create secondary windows.
- Use the built-in Contents Editor to create Windows 95-style contents files. Works *a lot* better than Microsoft's HCW.
- No need to mess with Microsoft's SHED: use the built-in SHG Editor to create hotspot bitmaps. Draw your hotspots on the bitmap and pick the topic to link to from the list.
- With the built-in Macro Editor you can easily compose WinHelp macros whenever needed. It will tell you what the correct parameters are and provide information on them.
- If you have a problem, just consult the online help. The help file was completely created with HelpScribble, of course.
- HelpScribble is shareware. However, the unregistered version is *not* crippled in any way. It will only add a small note to your help topics to encourage you to be honest and to register the shareware you use.

These options are very interesting for Delphi and C++Builder developers:

- If you are a component writer, use the Delphi Parser to build an outline help file for your component. Just fill in the spaces and you are done. HelpScribble can also extract the comments from your source file and use them as the default descriptions.
- If you are an application writer, HelpScribble provides you with a property editor for the HelpContext property. You can select the topic you need from a list of topic titles or simply instruct to create a new topic. No need to remember obscure numbers.
- The property editor also provides a tree view of all the components on your form and their HelpContext properties. This works very intuitively. (Much nicer than those help tools that simply mess with your .dfm files.)
- HelpScribble can perform syntax highlighting on any Delphi source code in your help file.

HelpScribble is shareware, so feel free to grab your copy today from my web site at <http://www.ping.be/jg/>

