

makeDoc

Lazarus/Free Pascal project to export documentation lines with tag marker.

File: makeDoc.pas

Date: 2025-10-06

Author: [Leandro](#)

Version: 1.0.0 / 2025-10-06

Copyright App: CC01 1.0 Universal

Copyright Icons: Uicons by [Flaticon](#)

Note: Changes in this document will be discarded on next build, any changes should be made on source code documentation instead.

Details

Save formatted lines from source code into documentation file.

Read source code line-by-line and save prefixed lines by tag ??D to file.

C/C++/Free Pascal source codes lines start with tag //D and Bash lines start with tag //D.

Only those lines started by tags are exported to documentation files.

Mixed commented lines can co-exist at same source code, one for local documentation purpose and another to be exported to appropriate documentation file.

All lines are documented using Markdown format, the exported document can be read by an Markdown program reader.

[\[Top\]](#) | [\[Index\]](#)

Index

[Top](#)

[Details](#)

[Glossary](#)

[Unit](#)

[Interface](#)

[Uses](#)

[Type](#)

[Variables](#)

[Implementation](#)

[saveHeader](#) Save a formatted html header to parameter file descriptor.

[saveFooter](#) Save a formatted html footer to parameter file descriptor.

[btnAboutClick](#) Treat About button click event.

[btnCloseClick](#) Treat Close button click event.

[btnRunClick](#) Treat Run button click event.

[FormShow](#) Treat form show event.

[Bottom](#)

[\[Top\]](#) | [\[Index\]](#)

Glossary

Use Description

Constants: Memory space for read only data.

Variables: Memory space for read/write data.

Functions: fileSOURCE/Executable statement code, can be called anywhere from source code.

Parameters: Data passed to functions.

Result: Functions result after execution.

Return: Always an integer returned from function to inform success or failure.

none: Is similar as a void type, no parameter, no result or no return from function.

string: Char vector to store a group of characters.

type[]: Memory vector space to store contiguous data type.

[\[Top\]](#) | [\[Index\]](#)

Tags Programming Language Documentation

//D>: C/C++, C#, Java, JavaScript, Pascal/Object Pascal.

//D>: Bash, Zsh, Python, Perl, Ruby, Go, Swift, Kotlin, Rust

--D>: SQL, Ada, Haskell.

''D>: Visual Basic, VBScript.

%%D>: LaTeX, MATLAB.

??E>: End, stop documentation until ??B> tag.

??B>: Begin, start documentation until ??E> tag.

??M>:Macro, evaluate lines to show variables values.

[\[Top\]](#) | [\[Index\]](#)

Unit makeDoc

Interface

Uses

Name	Description
LCLType	Key codes
Classes	Abstract and concrete classes for object-oriented programming
Forms	Creating and managing user interface Forms
Controls	Common controls, such as TButton, TEdit, TMemo, TComboBox, and TListBox
Graphics	Native graphics classes like TCanvas, TFont, and TBitmap
Dialogs	Standard dialog components and functions like ShowMessage and dialogs for file opening/saving, color selection, and more
StdCtrls	Standard components like TButton, TLabel, TEdit, TMemo, TCheckBox, and TRadioButton
EditBtn	Edit box with an associated button, is found in the LCLBase unit
SysUtils	Part of the Free Pascal Run-Time Library (RTL)
RegExpr	Working with regular expressions is RegExpr

[\[Top\]](#) | [\[Index\]](#)

Type

Type	Name	Description
<i>TFormMain</i>	FormMain	Main form application
<i>TButton</i>	btnClose	Button close
<i>TButton</i>	btnAbout	Show about box
<i>TButton</i>	btnRun	Execute documentation procedure
<i>TFileNameEdit</i>	editSource	Source file edit box
<i>TFileNameEdit</i>	editDestine	Destine file edit box
<i>TLabel</i>	lblDestine	Source file static label
<i>TLabel</i>	lblSource	Destine file static label
<i>procedure</i>	btnAboutClick	Treat About button click event
<i>procedure</i>	btnCloseClick	Treat Close button click event
<i>procedure</i>	btnRunClick	Treat Run button click event
<i>procedure</i>	FormShow	Treat form show event

[\[Top\]](#) | [\[Index\]](#)

Variables

Type	Name	Description
<i>TFormMain</i>	FormMain	Main form application

[\[Top\]](#) | [\[Index\]](#)

Implementation

saveHeader()

boolean **saveHeader**(*TextFile* **fd**) : *string*
Save a formatted html header to parameter file descriptor.

Parameter:
TextFile: **fd** - Target file descripton.

Result:
string: HTML header.

Return:
boolean: **true** - Success
boolean: **false** - Error

[\[Top\]](#) | [\[Index\]](#)

saveFooter()

boolean **saveFooter**(*TextFile* **fd**) : *string*
Save a formatted html footer to parameter file descriptor.

Parameter:

*TextFile: **fd*** – Target file descripton.

Result:

string: HTML footer.

Return:

*boolean: **true*** – Success

*boolean: **false*** – Error

[\[Top\]](#) | [\[Index\]](#)

btnAboutClick()

*boolean **btnAboutClick(TObject Sender) : None***

Treat About button click event.

Parameter:

*TObject: **Sender*** – Objects and Events from main window.

Result:

TFormMain: Create and show a formatted about form window.

Return:

None

[\[Top\]](#) | [\[Index\]](#)

btnCloseClick()

*boolean **btnCloseClick(TObject Sender) : None***

Treat Close button click event.

Parameter:

*TObject: **Sender*** – Objects and Events from main window.

Result:

TFormMain: End and close main form window.

Return:

None

[\[Top\]](#) | [\[Index\]](#)

btnRunClick()

*boolean **btnRunClick(TObject Sender) : None***

Treat Run button click event.

Parameter:

*TObject: **Sender*** – Objects and Events from main window.

Result:

TFormMain: Run documentation procedure.

Return:

None

[\[Top\]](#) | [\[Index\]](#)

FormShow()

*boolean **FormShow(TObject Sender) : None***

Treat form show event.

Parameter:

*TObject: **Sender*** – Objects and Events from main window.

Result:

TFormMain: Prepare main form edit box to load and show current directory.

Return:

None

[\[Top\]](#) | [\[Index\]](#)