

```

:
( Notepad )
MDI                               Txt                               Rtf
.
OpenDialog                         MainMenu
:
: Form1

```

bdRightToLeft	BiDiMode
	Caption
fsMDIForm	FormStyle
MainForm	Name
poDesktopCenter	Position

```

:( ) ( ) : MainMenu
: ( ) FileMenu Name ( )

```

Shortcut	Name	Caption
	New	&
	OpenFile	&
	CloseApp	&

```

: ( ) Window1 Name ( )

```

Shortcut	Name	Caption
		-

```

: OpenFileDialog

```

	.rtf	DefaultExt
	: Filter	Filter
Filter Name	Filter	
Rich Text	*.rtf	
Text File	*.Txt	
All File	*.*	

```

. MyTextEditor.Dpr Main.pas

```

MainMenu : File → New → Form

.FontDialog PrintDialog SaveDialog Win32 RichText

:

bdRightToLeft	BiDiMode
fsMDIChild	FormStyle
EditForm	Name

() () () MainMenu1

:

: () FileMenu Name ()

Shortcut	Name	Caption
	New	&
	OpenFile	&
Ctrl+S	Save	
	SaveAs	
	PrintFile	
	CloseFile	
	CloseAll	&

: () EditMenu Name ()

Shortcut	Name	Caption
Ctrl+C	CopyText	
Ctrl+X	Cuttext	
Ctrl+V	PasteText	
Ctrl+A	SelectAll	

: () EditMenu Name ()

Shortcut	Name	Caption
	Left1	
	Right1	
	Center1	
		-
	WordWrap1	
	Font1	
True	RadioItem	

--

: Richedit1

	alClient	Align
	Editor	Name

: Savedialog1

	.rtf	DefaultExt
	Filter	Filter
Filter Name	Filter	
Rich Text	*.rtf	
Text File	*.Txt	

EditText.pas

```

: MainForm
TEditForm.Create(Self);

: EditForm
TEditForm.Create(Self);

: EditForm OnClose
Action := CaFree ;
CaMinimized

(
( TrichEdit ) Editor

:
. EditText Private String PathName

const
DefaultFileName = ' ';

: EditText public
Procedure Open(const AFileName: string);

Ctrl+Alt+C

: Implementation
PathName := AFileName;
    
```

http://www.alshater.net

```

Caption := ExtractFileName(AFileName);
with Editor do
begin
  Lines.LoadFromFile(PathName);
  SelStart := 0;
  Modified := False;
end;

: ExtractFileName ( AFileName)

: With Editor Do begin .... End;

Editor.Lines.LoadFromFile(PathName);
Editor.SelStart := 0;
Editor.Modified := False;

Width
Editor.selstart := 0
Editor.Modified

True

MainForm

If OpenFileDialog1.Execute Then
TeditForm.Create(Self).Open(OpenDialog1.FileName ) ;

Open Then

EditForm

MainForm.Openfile.Click ;

: EditForm OnCreate

PathName := DefaultFileName;

PathName

:

rtf TRichEdit Editor
Editor Tstrings Txt rtf
. .Txt ÷
: EditForm

var S: Tstringlist;
begin
if PathName = DefaultFileName then
  SaveAsClick(Sender)
else
begin
if uppercase(rightstr(PathName,3)) = uppercase( 'rtf' ) then
  Editor.Lines.SaveToFile(PathName) else
try

```

```

http://www.alshater.net
S:=Tstringlist.Create ;
s.AddStrings(Editor.Lines )
finally
s.SaveToFile(PathName) ;
end;
Editor.Modified := False;
end;

```

```

" " PathName
: PathName
) Editor.Lines.SaveToFile(PathName) rtf 1
'rtf' PathName
TStringList S Txt 2
Editor.Modified False Editor

```

```

: EditForm
SaveDialog1.FileName := PathName;
if SaveDialog1.Execute then
begin
PathName := SaveDialog1.FileName;

Caption := ExtractFileName(PathName);
SaveClick(Sender);
end;

```

```

PathName
SaveClick ( Sender )

```

```

: EditForm
Close;
: MainForm
Close;

```

: EditForm
MainForm.CloseApp.Click ;

```

: EditForm OnCloseQuery
Const
SWarningText = '
begin
if Editor.Modified then
begin
case MessageDlg(Format(SWarningText + #13 + '%s', [PathName]), mtConfirmation,
[mbYes, mbNo, mbCancel], 0) of
mrYes: SaveClick(Self);
mrCancel: CanClose := False;
end;
end;
end;
end;

```

```

OnCloseQuery
CanClose
False True
True ) Editor.Modified
: (
. SaveClick : Yes
: No
CanClose : Cancel
False
MessageDlg #13
Enter
:
:
Editor.CopyToClipboard;
:
Editor.CutToClipboard;
:

```

http://www.alshater.net

```
Editor.PasteFromClipboard;
```

```
:
```

```
Editor.SelectAll;
```

```
Editor
```

```
( )
```

```
:
```

```
Copytext.Enabled := editor.SelLength > 0;
```

```
Cutttext.Enabled := editor.SelLength > 0;
```

```
Pastetext.Enabled := clipboard.HasFormat(CF_text);
```

```
. EditForm
```

```
uses
```

```
ClipBrd
```

```
Editor
```

```
SelLength
```

```
:
```

```
with Sender as TMenuItem do Checked := True;
```

```
with Editor.Paragraph do
```

```
if Left1.Checked then
```

```
Alignment := taLeftJustify
```

```
else if Right1.Checked then
```

```
Alignment := taRightJustify
```

```
else if Center1.Checked then
```

```
Alignment := taCenter;
```

```
OnClick
```

```
( ) ( )
```

```
. Left1Click
```

```
Event
```

```
:
```

```
with Editor do
```

```
begin
```

```
WordWrap := not WordWrap; { toggle word wrapping }
```

```
if WordWrap then
```

```
ScrollBars := ssVertical
```

```
else
```

```
ScrollBars := ssBoth;
```

```
WordWrap1.Checked := WordWrap; { set menu item check }
```

```
end;
```

```
:
```

```
FontDialog1.Font := Editor.Font;
```

```
if FontDialog1.Execute then
```

```
Editor.SelAttributes.Assign(FontDialog1.Font);
```

```

Memo
Memo
RichEdit
SelAttributes
: MainForm
(
:
. GroupIndex 1
1 0
GroupIndex 2
.
:
MainForm.Tile
)
:
MainForm.Cascade;
.
:
WindowsMenu MainForm
( ) Window1
.
( Setup )

```