

MSWordDocument component

[Registration](#)

Scalabium

Scalable solutions that grow with you...

TMSWordDocument component for Delphi/C++Builder allow to read any MS Word document directly without MS Word installed. OLE automation of MS Word (or any other app from MS Office) is not required.

You may extract plain text and SummaryInformation from any doc-file very fast.

Visit the web site at <http://www.scalabium.com> to see if updated help files is available.

Registration

[By credit card](#) [By mail](#)

Why Register

Thank you for your interest in TMSWordDocument component.

Registered users will receive the latest registered version of TMSWordDocument component, free on-line support, and the source code. Any trial limitations will be removed from registered version.

Online registration

You can order a product online at:

1. PayPro: [https://store.payproglobal.com/checkout?products\[1\].\[id\]=15415](https://store.payproglobal.com/checkout?products[1].[id]=15415)

Questions & Comments

Please refer questions or comments about TMSWordDocument component to:

<http://www.scalabium.com>

<mailto:mshkolnik@scalabium.com>

TSMWordSummaryInformation object

[See also](#) [Properties](#) [Methods](#) [Example](#)

Unit

[MSWord](#)

Description

This class provide an access to parsed values of SummaryInformation that is tored in processed doc-file.

Same values you may view in Properties dialog of MS Word (when doc-file is loaded)

Author property

[See also](#)

Applies to

[TSMWordSummaryInformation](#) object

Declaration

property Author: **string**;

Description

Using this property you may read an author that created a doc-file

CharCount property

[See also](#)

Applies to

[TSMWordSummaryInformation](#) object

Declaration

property CharCount: Integer;

Description

Using this property you may read a total count of characters in doc-file

Comments property

[See also](#)

Applies to

[TSMWordSummaryInformation](#) object

Declaration

property Comments: **string**;

Description

Using this property you may read the comments that are added by author of doc-file

Created property

[See also](#)

Applies to

[TSMWordSummaryInformation](#) object

Declaration

property Created: TDateTime;

Description

Using this property you may read a date-time when doc-file is created by author

EditTime property

[See also](#)

Applies to

[TSMWordSummaryInformation](#) object

Declaration

property EditTime: Integer;

Description

Using this property you may read a total time (in minutes) that author is spent on doc-file editing

Keywords property

[See also](#)

Applies to

[TSMWordSummaryInformation](#) object

Declaration

property Keywords: **string**;

Description

Using this property you may read keywords that defined by author of doc-file

LastAuthor property

[See also](#)

Applies to

[TSMWordSummaryInformation](#) object

Declaration

property LastAuthor: **string**;

Description

Using this property you may read an author that saved a doc-file in last time

LastPrinted property

[See also](#)

Applies to

[TSMWordSummaryInformation](#) object

Declaration

property LastPrinted: TDateTime;

Description

Using this property you may read a date-time when doc-file was printed

LastSaved property

[See also](#)

Applies to

[TSMWordSummaryInformation](#) object

Declaration

property LastSaved: TDateTime;

Description

Using this property you may read a date-time when doc-file was saved

PageCount property

[See also](#)

Applies to

[TSMWordSummaryInformation](#) object

Declaration

property PageCount: Integer;

Description

Using this property you may read a total count of pages in doc-file

RevNumber property

[See also](#)

Applies to

[TSMWordSummaryInformation](#) object

Declaration

property RevNumber: **string**;

Description

Using this property you may read a revision number for doc-file

Subject property

[See also](#)

Applies to

[TSMWordSummaryInformation](#) object

Declaration

property Subject: **string**;

Description

Using this property you may read subject that defined by author of doc-file

Template property

[See also](#)

Applies to

[TSMWordSummaryInformation](#) object

Declaration

property Template: **string**;

Description

Using this property you may read a name of template-file that is used for doc-file creation

Title property

[See also](#)

Applies to

[TSMWordSummaryInformation](#) object

Declaration

property Title: **string**;

Description

Using this property you may read title for doc-file that defined by author

WordCount property

[See also](#)

Applies to

[TSMWordSummaryInformation](#) object

Declaration

property WordCount: Integer;

Description

Using this property you may read a total count of words in doc-file

ParseProperty method

Applies to

[TSMWordSummaryInformation](#) object

Declaration

```
procedure ParseProperty(propID: dWord; Value: Pointer);
```

Description

This method allow to specify any property.

For example, to change a Title property:

```
str := 'New value for Title';
```

```
yourMSWordDocument.SummaryInformation.ParseProperty($00000002, @str[1]);
```



TMSWordDocument component

[See also](#) [Properties](#) [Methods](#) [Events](#)

Unit

[MSWord](#)

Description

TMSWordDocument component is a basic component for doc-file reading and text extracting.

FileName property

Applies to

[TMSWordDocument](#) component

Declaration

```
property FileName: string;
```

Description

FileName is the name of the doc-file that must be loaded and parsed by the TMSWordDocument component.

PlainText property

[Example](#)

Applies to

[TMSWordDocument](#) component

Declaration

```
property PlainText: TStrings;
```

Description

Using this run-time property you may read an extracted text from doc-file.

SummaryInformation property

[See also](#)

Applies to

[TMSWordDocument](#) component

Declaration

property SummaryInformation: TSMWordSummaryInformation;

Description

Using this compound property you may read any value from Summary Information that is stored in MS Word document

Execute method

[See also](#) [Example](#)

Applies to

[TMSWordDocument](#) component

Declaration

```
procedure Execute;
```

Description

An Execute method start a process specified doc-file (see FileName property). All extracted text is available in PlainText property.

OnAfterExecute event

[See also](#) [Example](#)

Applies to

[TMSWordDocument](#) component

Declaration

property OnAfterExecute: TNotifyEvent;

Description

This event will be called after each read/load process. So you may use it if you want to activate some own actions which must be executed after read.

The Sender parameter is a TMSWordDocument component which finished a text loading.

OnBeforeExecute event

[See also](#) [Example](#)

Applies to

[TMSWordDocument](#) component

Declaration

property OnBeforeExecute: TNotifyEvent;

Description

This event will be called before each read/load process is activated. So here you can activate some own actions which must be executed before doc-file reading.

The Sender parameter is a TMSWordDocument component which will start a text loading.

Registration by credit card

Visa/Discover/MasterCard/AmericanExpress 50 EUR for full package with sources

For technical support or comments about this program, you may contact Mike Shkolnik at: <mailto:mshkolnik@scalabium.com>

For your convenience we have contracted another companies (registrators), PayPro to process any orders you may wish to place with your PayPal, Visa, Discover, MasterCard or other credit cards.

Registrators can be easily contacted **for orders only** via any of the following methods:

ONLINE ORDERS

You may register SMWord suite via an online order form by pointing your browser to:

1. PayPro: [https://store.payproglobal.com/checkout?products\[1\].\[id\]=15415](https://store.payproglobal.com/checkout?products[1].[id]=15415)

Registration by mail order

Select **Print Topic...** from the **File** menu to print this form.

Item: TMSWordDocument component for Delphi/C++Builder

Price: EUR50 with sources or EUR35 without sources

Please register my copy of TMSWordDocument component
for Delphi/C++Builder,
I am sending a check or money order for \$_____

Name:_____

Company:_____

Address1:_____

Address2:_____

City:_____

State:_____ **Zip:**_____

Country:_____

Phone:_____ optional

Email:_____

Please send completed form with payment to:

Mike Shkolnik

ul.Pragskaya 4, kv.39

Kiev, 02090

Ukraine

See also

[TMSWordDocument](#)

Properties

- [Author](#)
- [CharCount](#)
- [Comments](#)
- [Created](#)
- [EditTime](#)
- [Keywords](#)
- [LastAuthor](#)
- [LastPrinted](#)
- [LastSaved](#)
- [PageCount](#)
- [RevNumber](#)
- [Subject](#)
- [Template](#)
- [Title](#)
- [WordCount](#)

- **Published properties**

Methods

- [ParseProperty.](#)

[FileTimeToElapsedTime](#)

[FileTimeToDateTime](#)

- **Published methods**

TSMWordSummaryInformation class example

For example, you have a TListView component (lvSummary) with two columns and you want to display in this listview a SummaryInformation for loaded doc-file:

```
with lvSummary.Items.Add do
begin
Caption := 'FileName';
SubItems.Add(MSWordDocument.FileName)
end;

with lvSummary.Items.Add do
begin
Caption := 'Title';
SubItems.Add(MSWordDocument.SummaryInformation.Title)
end;
with lvSummary.Items.Add do
begin
Caption := 'Subject';
SubItems.Add(MSWordDocument.SummaryInformation.Subject)
end;
with lvSummary.Items.Add do
begin
Caption := 'Author';
SubItems.Add(MSWordDocument.SummaryInformation.Author)
end;

with lvSummary.Items.Add do
begin
Caption := 'Keywords';
SubItems.Add(MSWordDocument.SummaryInformation.Keywords)
end;
with lvSummary.Items.Add do
begin
Caption := 'Comments';
SubItems.Add(MSWordDocument.SummaryInformation.Comments)
end;
with lvSummary.Items.Add do
begin
Caption := 'Template';
SubItems.Add(MSWordDocument.SummaryInformation.Template)
end;

with lvSummary.Items.Add do
begin
Caption := 'LastAuthor';
SubItems.Add(MSWordDocument.SummaryInformation.LastAuthor)
end;
with lvSummary.Items.Add do
begin
```



```

Caption := 'Created';
SubItems.Add(DateToStr(MSWordDocument.SummaryInformation.Created))
end;
with lvSummary.Items.Add do
begin
Caption := 'LastSaved';
SubItems.Add(DateToStr(MSWordDocument.SummaryInformation.LastSaved))
end;
with lvSummary.Items.Add do
begin
Caption := 'LastPrinted';
SubItems.Add(DateToStr(MSWordDocument.SummaryInformation.LastPrinted))
end;
with lvSummary.Items.Add do
begin
Caption := 'RevNumber';
SubItems.Add(MSWordDocument.SummaryInformation.RevNumber)
end;
with lvSummary.Items.Add do
begin
Caption := 'EditTime';
SubItems.Add(IntToStr(MSWordDocument.SummaryInformation.EditTime))
end;

with lvSummary.Items.Add do
begin
Caption := 'PageCount';
SubItems.Add(IntToStr(MSWordDocument.SummaryInformation.PageCount))
end;
with lvSummary.Items.Add do
begin
Caption := 'CharCount';
SubItems.Add(IntToStr(MSWordDocument.SummaryInformation.CharCount))
end;
with lvSummary.Items.Add do
begin
Caption := 'WordCount';
SubItems.Add(IntToStr(MSWordDocument.SummaryInformation.WordCount))
end;

```

MSWord unit

In this unit you will find a declaration of basic TMSWordDocument component and TSMWordSummaryInformation class.

Components

[TMSWordDocument](#)

Objects

[TSMWordSummaryInformation](#)

See also

[CharCount](#)

[Comments](#)

[Created](#)

[EditTime](#)

[Keywords](#)

[LastAuthor](#)

[LastPrinted](#)

[LastSaved](#)

[PageCount](#)

[RevNumber](#)

[Subject](#)

[Template](#)

[Title](#)

[WordCount](#)

See also

[Author](#)

[Comments](#)

[Created](#)

[EditTime](#)

[Keywords](#)

[LastAuthor](#)

[LastPrinted](#)

[LastSaved](#)

[PageCount](#)

[RevNumber](#)

[Subject](#)

[Template](#)

[Title](#)

[WordCount](#)

See also

[Author](#)

[CharCount](#)

[Created](#)

[EditTime](#)

[Keywords](#)

[LastAuthor](#)

[LastPrinted](#)

[LastSaved](#)

[PageCount](#)

[RevNumber](#)

[Subject](#)

[Template](#)

[Title](#)

[WordCount](#)

See also

[Author](#)

[CharCount](#)

[Comments](#)

[EditTime](#)

[Keywords](#)

[LastAuthor](#)

[LastPrinted](#)

[LastSaved](#)

[PageCount](#)

[RevNumber](#)

[Subject](#)

[Template](#)

[Title](#)

[WordCount](#)

See also

[Author](#)

[CharCount](#)

[Comments](#)

[Created](#)

[Keywords](#)

[LastAuthor](#)

[LastPrinted](#)

[LastSaved](#)

[PageCount](#)

[RevNumber](#)

[Subject](#)

[Template](#)

[Title](#)

[WordCount](#)

See also

[Author](#)

[CharCount](#)

[Comments](#)

[Created](#)

[EditTime](#)

[LastAuthor](#)

[LastPrinted](#)

[LastSaved](#)

[PageCount](#)

[RevNumber](#)

[Subject](#)

[Template](#)

[Title](#)

[WordCount](#)

See also

[Author](#)

[CharCount](#)

[Comments](#)

[Created](#)

[EditTime](#)

[Keywords](#)

[LastPrinted](#)

[LastSaved](#)

[PageCount](#)

[RevNumber](#)

[Subject](#)

[Template](#)

[Title](#)

[WordCount](#)

See also

[Author](#)

[CharCount](#)

[Comments](#)

[Created](#)

[EditTime](#)

[Keywords](#)

[LastAuthor](#)

[LastSaved](#)

[PageCount](#)

[RevNumber](#)

[Subject](#)

[Template](#)

[Title](#)

[WordCount](#)

See also

[Author](#)

[CharCount](#)

[Comments](#)

[Created](#)

[EditTime](#)

[Keywords](#)

[LastAuthor](#)

[LastPrinted](#)

[PageCount](#)

[RevNumber](#)

[Subject](#)

[Template](#)

[Title](#)

[WordCount](#)

See also

[Author](#)

[CharCount](#)

[Comments](#)

[Created](#)

[EditTime](#)

[Keywords](#)

[LastAuthor](#)

[LastPrinted](#)

[LastSaved](#)

[RevNumber](#)

[Subject](#)

[Template](#)

[Title](#)

[WordCount](#)

See also

[Author](#)

[CharCount](#)

[Comments](#)

[Created](#)

[EditTime](#)

[Keywords](#)

[LastAuthor](#)

[LastPrinted](#)

[LastSaved](#)

[PageCount](#)

[Subject](#)

[Template](#)

[Title](#)

[WordCount](#)

See also

[Author](#)

[CharCount](#)

[Comments](#)

[Created](#)

[EditTime](#)

[Keywords](#)

[LastAuthor](#)

[LastPrinted](#)

[LastSaved](#)

[PageCount](#)

[RevNumber](#)

[Template](#)

[Title](#)

[WordCount](#)

See also

[Author](#)

[CharCount](#)

[Comments](#)

[Created](#)

[EditTime](#)

[Keywords](#)

[LastAuthor](#)

[LastPrinted](#)

[LastSaved](#)

[PageCount](#)

[RevNumber](#)

[Subject](#)

[Title](#)

[WordCount](#)

See also

[Author](#)

[CharCount](#)

[Comments](#)

[Created](#)

[EditTime](#)

[Keywords](#)

[LastAuthor](#)

[LastPrinted](#)

[LastSaved](#)

[PageCount](#)

[RevNumber](#)

[Subject](#)

[Template](#)

[WordCount](#)

See also

[Author](#)

[CharCount](#)

[Comments](#)

[Created](#)

[EditTime](#)

[Keywords](#)

[LastAuthor](#)

[LastPrinted](#)

[LastSaved](#)

[PageCount](#)

[RevNumber](#)

[Subject](#)

[Template](#)

[Title](#)

See also

[TSMWordSummaryInformation](#)

Properties

- [ConvertSysCharacters](#)

- [FileName](#)

- ■ [PlainText](#)

- [SummaryInformation](#)

- **Read-only** ■ **Published properties**

Methods

~~Create~~{linkDelphi=Create_Method}

~~Destroy~~{linkDelphi=Destroy_Method}

- [Execute](#)

- **Published methods**

Events

- [OnAfterExecute](#)
- [OnBeforeExecute](#)
- **Published events**

PlainText property example

TMSWordDocument component have an Execute method. Call this method when you specified a correct document to FileName property.

After that you may read parsed text using PlainText property.

For example, to extract a text from C:\My Documents\Data_2002.doc file and load it in Memo1 component:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  MSWordDocument.FileName := 'C:\My Documents\Data_2002.doc';
  MSWordDocument.Execute;
  Memo1.Lines.Assign(MSWordDocument.PlainText);
end;
```

See also

[TSMWordSummaryInformation](#)

See also

[FileName](#)

[PlainText](#)

[ConvertSysCharacters](#)

See also

[Execute](#)

[OnBeforeExecute](#)

OnAfterExecute event example

```
begin
{show a text in status bar}
StatusBar.Text := 'Text loaded.';
end;
```

See also

[Execute](#)

[OnAfterExecute](#)

OnBeforeExecute event example

```
begin  
{show a text in status bar}  
StatusBar.Text := 'Loading...';  
end;
```

FileTimeToElapsedTime

Applies to

[TSMWordSummaryInformation](#) object

Declaration

```
function FileTimeToElapsedTime(FileTime: TFileTime): Integer;
```

Description

This protected method allow to convert a date from TFileTime format to minutes.

FileTimeToDateTime

Applies to

[TSMWordSummaryInformation](#) object

Declaration

```
function FileTimeToDateTime(FileTime: TFileTime): TDateTime;
```

Description

This protected method allow to convert a date from TFileTime format to TDateTime format.

ConvertSysCharacters property

Applies to

[TMSWordDocument](#) component

Declaration

property ConvertSysCharacters: Boolean;

Description

Extracted plain text can contain some system characters. For example, Chr(13) is a new line, Chr(15) is a new page etc

If you'll set ConvertSysCharacters property in True, all such characters will be removed and replaced with correct correspondent character (for example, Chr(13) will be replaced by Chr(13)+Chr(10))

If you don't want such intellectual behaviour for TMSWordDocument, just set ConvertSysCharacters property in False and you'll receive an original text contents