

Worddriver updated

Michaël Van Canneyt

July 22, 2009

Abstract

In issue 3/2004 of Toolbox, a component was presented which can be used to create personalized letters and tables in MS-Word (TWordDriver). Since then, both the component and MS-Word have evolved to new versions. In this article, the updated version of the TWordDriver component is presented.

1 Introduction

In Issue 3/2004 of Toolbox, the TWordDriver component was presented. It was a Delphi component which enabled a programmer to export the data of a dataset to MS-Word or OpenOffice. The operation of the component was quite simple: all it did was a simple search-and-replace mechanism: In a start document, anything of the form

```
{fieldname}
```

was replaced by the contents of the corresponding field in the dataset. For each record in the dataset, a copy of the start document with the placeholders replaced by the data in the dataset was saved. All copies of this file could then be merged together in a merge document.

Alternatively, the component could create a table or fill an existing table with data from the dataset: a placeholder like the one above could be replaced with a word-table, filled with the data from the dataset.

Supported MS-Word versions were MS-Word 97, 2000 and XP, plus a generic algorithm which could be used with other versions of MS-Word.

2 Changes

Since its initial appearance, the TWordDriver component has changed relatively little. The first change is that the old 'KeepOpen' property was replaced by 3 other properties:

KeepWordOpen If set to 'True', MS-Word is left running after the export finished.

KeepDocumentsOpen If set to 'True', all created Documents are left open in MS-Word.

KeepOnlyMergeOpen If set to 'True', only the Merge-Document is left open in MS-Word.

Obviously, 'KeepDocumentsOpen' can only be set to 'True' if 'KeepWordOpen' is 'True', and the same is true for 'KeepOnlyMergeOpen'.

A change under the hood is the support for the Office 2003 and Office 2007 versions of MS-Word: these versions of word are now also supported natively, i.e. the type libraries for these versions of Office have been imported. Additionally, the auto-detection mechanism will now detect these versions of MS-Word as well.

The last change is the support for replacing form-fields in the MS-Word document. Till now, placeholders had to be placed in the document, and they had to have the form

```
{ANiceField}
```

they had to be put in the text as plain text. However, MS-Word also has native support for placeholders where users must enter data: form fields. These form fields can now also be filled in by the TWordDriver component, provided they have a tag (bookmark) associated with them.

To demonstrate this, the serial letter example presented with the TWordDriver component was changed so it uses form fields. To insert a form field in MS-Word 2007, the following must be done:

1. Enable 'Developer' mode in the 'Word Options' dialog. This will place a new tab on the ribbon, named 'Developer'.
2. On the 'Developer' tab of the ribbon, press 'Design mode' in the 'Controls' group.
3. Place the cursor on the location in the document where the form field must appear.
4. Select the 'plain text' placeholder from the controls tab. A placeholder will be put in the document.
5. With the new placeholder still selected, press 'Design mode' in the 'Controls' group on the 'Developer' tab of the ribbon.
6. A dialog appears as in figure 1 on page 3. Here a small descriptive text can be entered, as well as a 'Tag'. The 'Tag' must be set to the fieldname of the field whose contents should be put in the placeholder.

In older versions of MS-Word, the 'Tag' property is named differently. There, the 'Bookmark' property of the field must be set in the 'Text form field options' dialog.

3 The code

To enable the new Form Fields feature, the new 'UseFormFields' property of TWordDriver must be set to 'True'. If it is set to 'True', then the following routine is also run when creating a serial letter:

```
procedure TWord2007Driver.ReplaceFields(Value: TStrings);
```

```
Var
```

```
  F : Word2007.FormField;
```

```
  I : Integer;
```

```
  N : String;
```

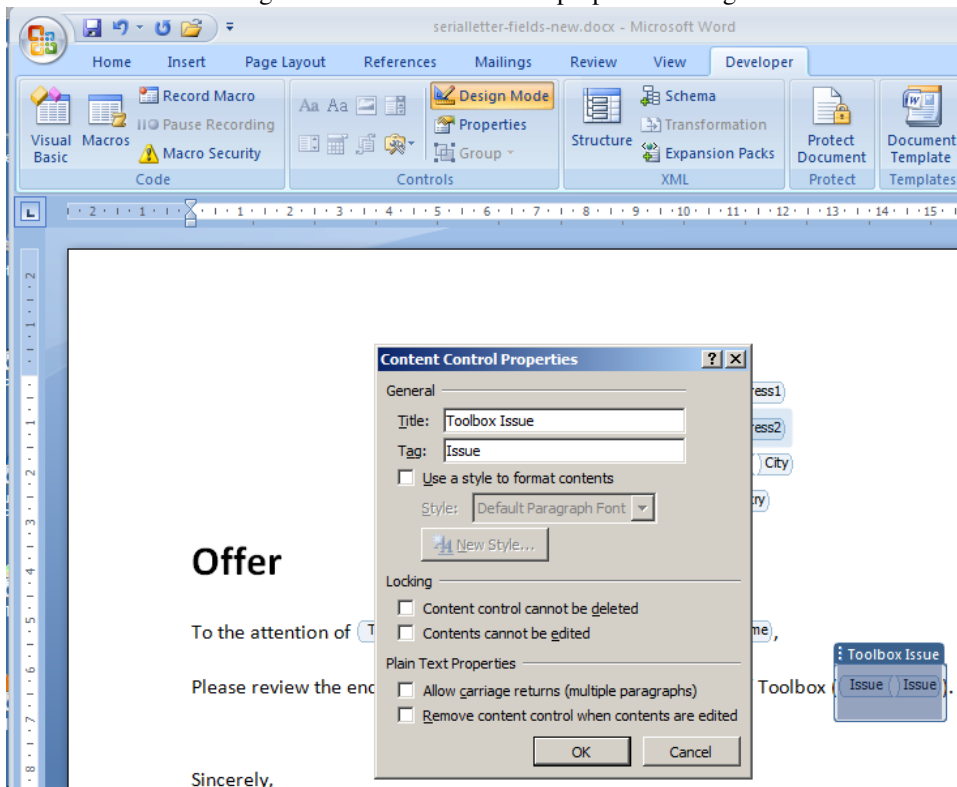
```
  O : Olevariant;
```

```
begin
```

```
  For I:=1 to FDocument.FormFields.Count do
```

```
    begin
```

Figure 1: The content control properties dialog



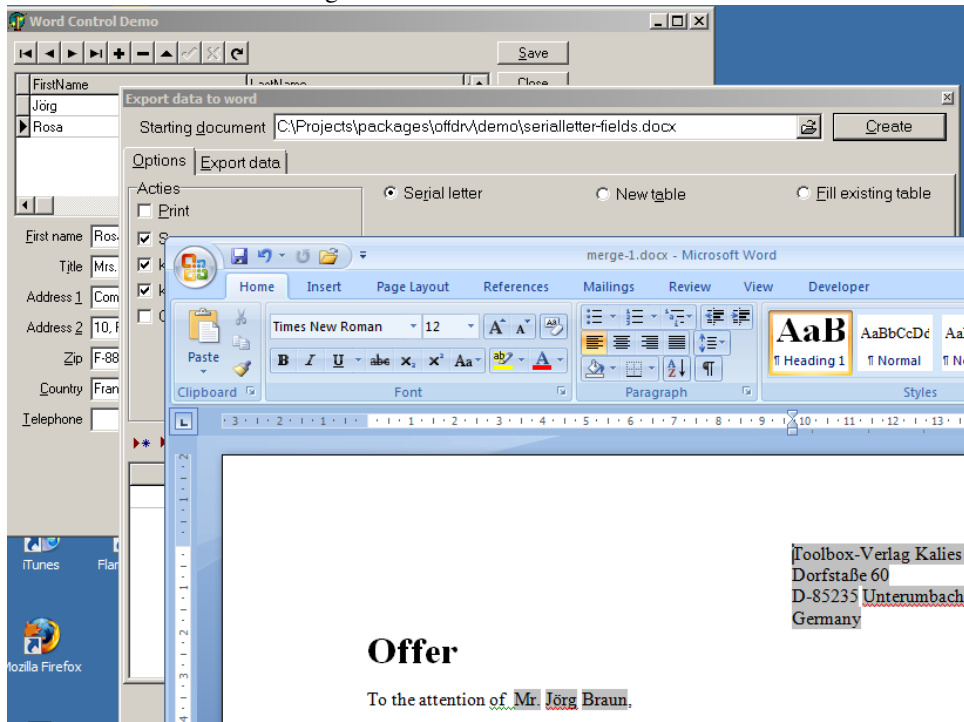
```
O:=I;
F:=FDocument.FormFields.Item(O);
N:=F.Name;
If (Value.IndexOfName(N)<>-1) then
    F.Result:=Value.Values[N];
end;
end;
```

For each of the formfields in the doducment, the loop checks whether a value is available in the list of values If so, the value is retrieved, and filled in the form field (Setting the 'Result' property sets the value of the form field).

A quite simple routine which adds a powerful feature to the TWordDriver, as it allows to re-use existing documents - many administrations have pre-formatted sample letters with such text fields - for mailing lists. The WordDriver component also has a method to get all the names available for substitution in the document. This method should be adapted so it also returns the list of names of Form Fields if the use of form fields is enabled. The following method retrieves the names of the various form fields in the Word document:

```
procedure TWord2007Driver.GetFormFieldNames(List: TStrings);
Var
    F : Word2007.FormField;
    I : Integer;
    N : String;
    O : Olevariant;
```

Figure 2: MS-Word 2007 in action



```
begin
  For I:=1 to FDocument.FormFields.Count do
    begin
      O:=I;
      F:=FDocument.FormFields.Item(O);
      N:=F.Name;
      List.Add(N);
    end;
  end;
```

As can be seen, this is quite similar to the 'ReplaceFields' method.

The customization dialog that comes with the TWordDriver component has been modified so it is aware of these new properties, and the sample serial letter has been reworked using form fields (it is named serialletter-fields-new.docx). When used, the result will look something like figure 2 on page 4. The sources of the type libraries for MS-Word 2003 and 2007 are included on the DVD accompanying this issue, as well as the updated version of the TWordDriver component. All code was tested with Delphi 2007.