Word Template Filling - User Manual

Owner: Resultmaker Research & Development
Version: 6.0.2
Revision: 2013-07-29

Lars Pedersen

Contents

1 Introduction

This document describes how it's possible to fill a Word template with data from Word data sources or Process Platform workflow variables. It describes how the overall architecture for how Word template filling works, how to create Word templates with Merge Fields, how to make Export Definitions that can map data from data sources or workflow variables to the template and how to call these from workflows.

2 Glossary

Term	Explanation
Word Data Source	Word Data Sources are assemblies capable of extracting data from a database or other data source and returning a data table to the export definition. These will take a predefined input (usually referring to a workflow variable) and will respond with a predefined output. The output will either be a single row of data corresponding to mapped Merge Fields or several rows corresponding to a Merge Field Region.
Export Definition	Export Definitions are mappings which determine what elements from the data source are to be used and which Word template is to be used. It also maps the data to the Merge Fields in the Word template. The data can come from a Word data source or from workflow variables in a workflow.
Word Mail Merge	A Merge Field is a Microsoft Word concept. It is a named field in a Word document that can be referenced by name. The fields are filled out by the Export Server Word Plugin.
Merge Field	A Merge Field is a Microsoft Word concept. It is a named field in a Word document and can be referred to by name. These are filled out by the Word Export Server plugin.
Merge Field Region	A Merge Field Region is an Aspose concept. You mark the beginning and the end of a region by naming 2 Merge Fields "TableStart:RegionName" and "TableEnd:RegionName" (Where RegionName is the name of the region). When provided with several rows of data, the region will be repeated for each row. If the start and end fields are located on the same row in a table a new row will be created for each new row of data.
Workflow data	Workflow data is the data accessible in the workflow. This is normally data which has been typed in by a user.
Export Server Word Plugin	Export Server Word Plugin is a component in the Process Platform capable of extracting data elements from a Process Platform workflow and injecting them into Word templates containing Merge Fields and Merge Field regions and producing filled word or pdf documents.

3 Architecture of elements in Word template filling

The architecture is based on workflows in the Process Platform calling the Word template filling functionality. This call includes all the data elements in the workflow as well as a reference to a so called Export Definition.

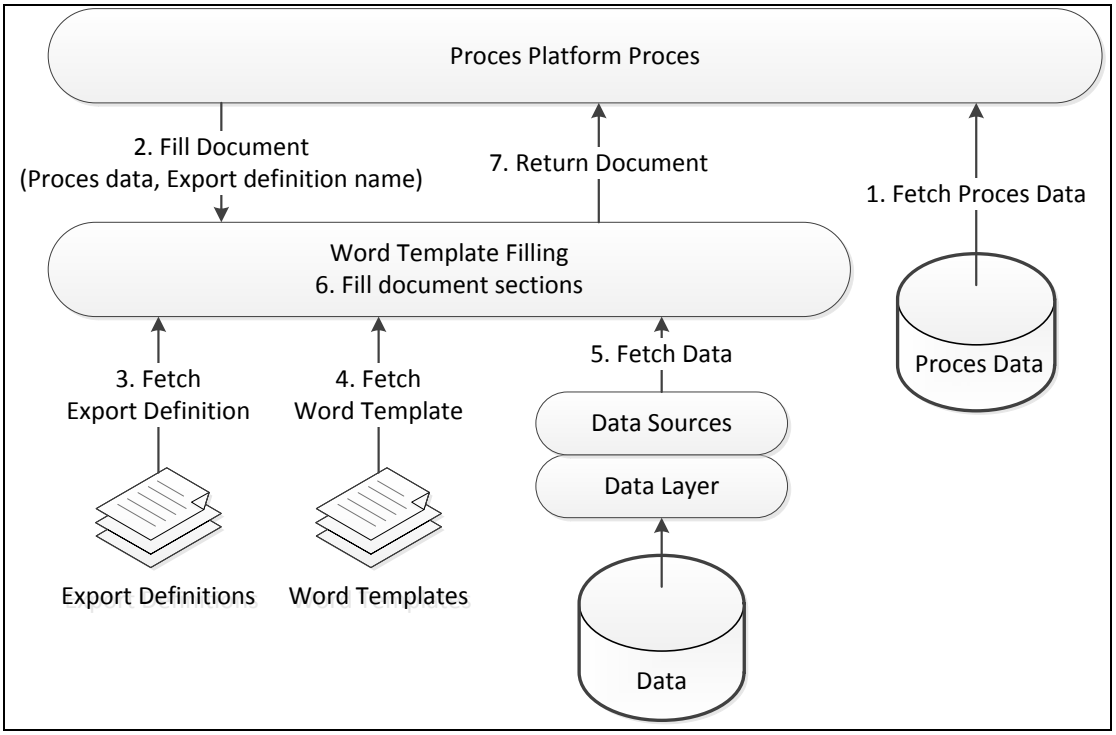


Figure 1 Architecture of Word document filling

From the Export Definition the Word template filling will be able to find the correct Word template, fetch data from data sources, merge these data into the Word template and return a word or pdf document to the workflow. The returned document is normally streamed to the end users browser of returned to a script.

4 Creation of Word templates

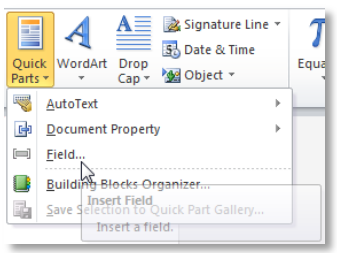
This section concerns creation of Merge Fields in Word documents.

In order to fill in data in a Word document it must contain named Merge Fields, which can be referred to from a mapping in an Export Definition. These fields are referred to as Word Mail Merge fields (Or Merge Field in Word).

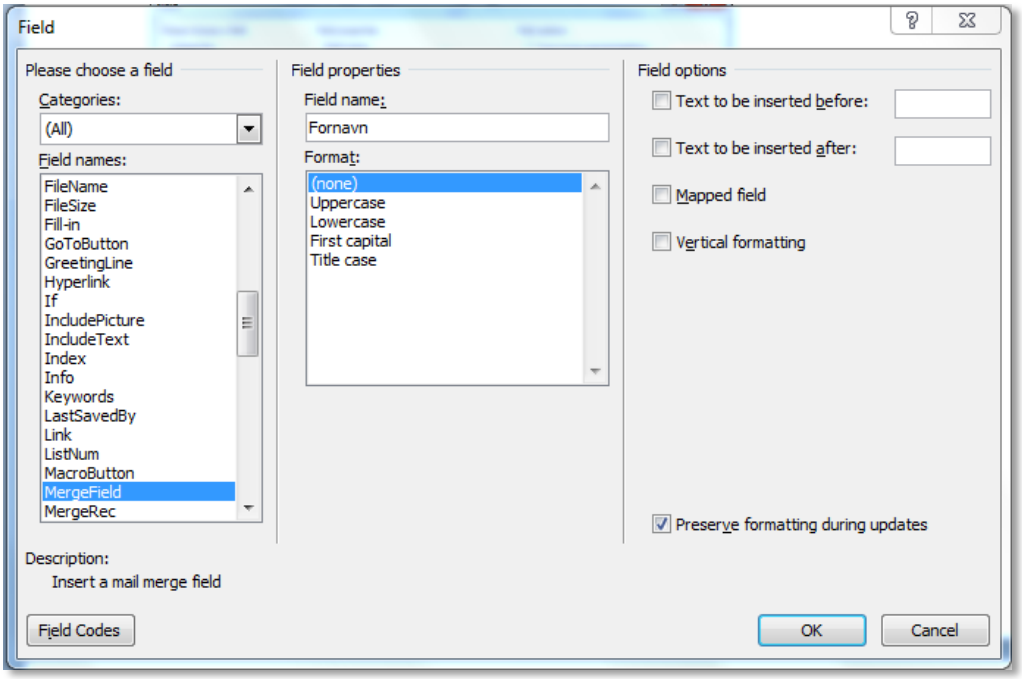
4.1 Mail Merge Fields

Open a Word document which requires Merge Fields.

To insert a Merge Field choose the “Insert” menu and select “Quick Parts”, then “Fields”.



In the following menu select ”Merge Field” in the list of ”Field names”. Then type the name of the field in the “Field name” field.



In the Word document the field is represented by the field name surrounded by <<>>.

First Name:	«FirstNameField»
Last Name:	«LastNameField»
Zipcode:	«ZipCodeField»
City:	«CityField»

4.2 Mail Merge Region

Mail Merge Regions are used to insert one or more rows of data of the same type. This could for instance be a series of loans belonging to the same person, where all the loans provide the same columns.

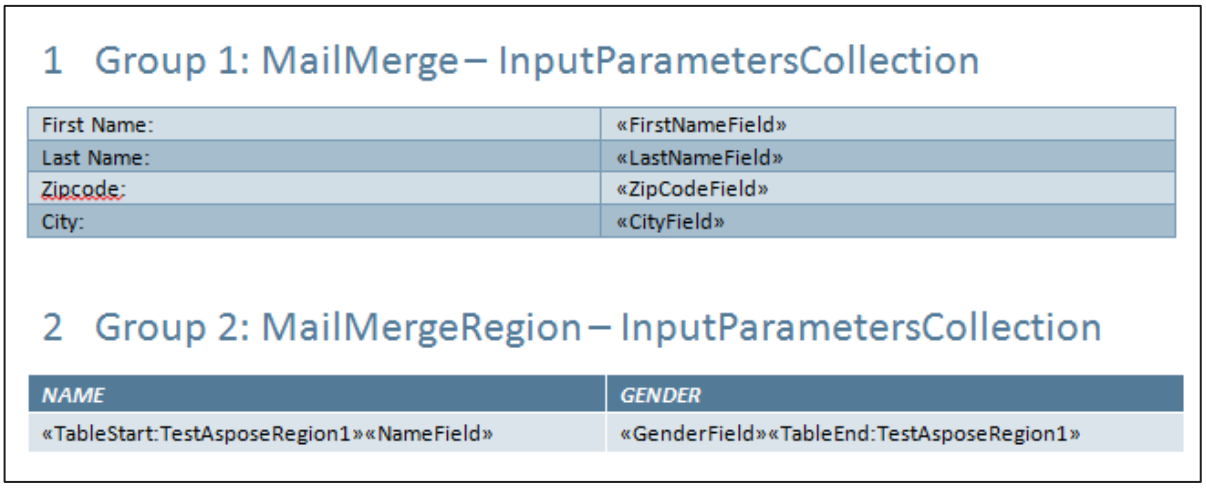
A mail merge region requires a Start and End field. To mark the beginning of a region a Merge Field named “TableStart:TableName” is inserted, where “TableName” is the name of the table. And to end the region a field named “TableEnd:TableName” is inserted. “TableName” must be the name of the region specified as “regionName in the OutputMapping tag in the Export Definition. (See 7 Appendix with examples of Export Definitions)

In between the Start and End fields one places the merge fields who correspond with the mapped fields from the Export Definition. These fields will be filled out and repeated for every row of data returned by a word data source or workflow variable table. If the fields are places on the same row of a word table, a new row will be added to the table for each row of data added.

When regions are created the following rules must always be followed:

- TableStart and TableEnd fields must reside in the same section of the word document.
- If inserted in a table, TableStart and TableEnd must be placed on the same row of the table.
- For every TableStart there must always be a TableEnd with the same region name specified.

Example of a word template with fields:



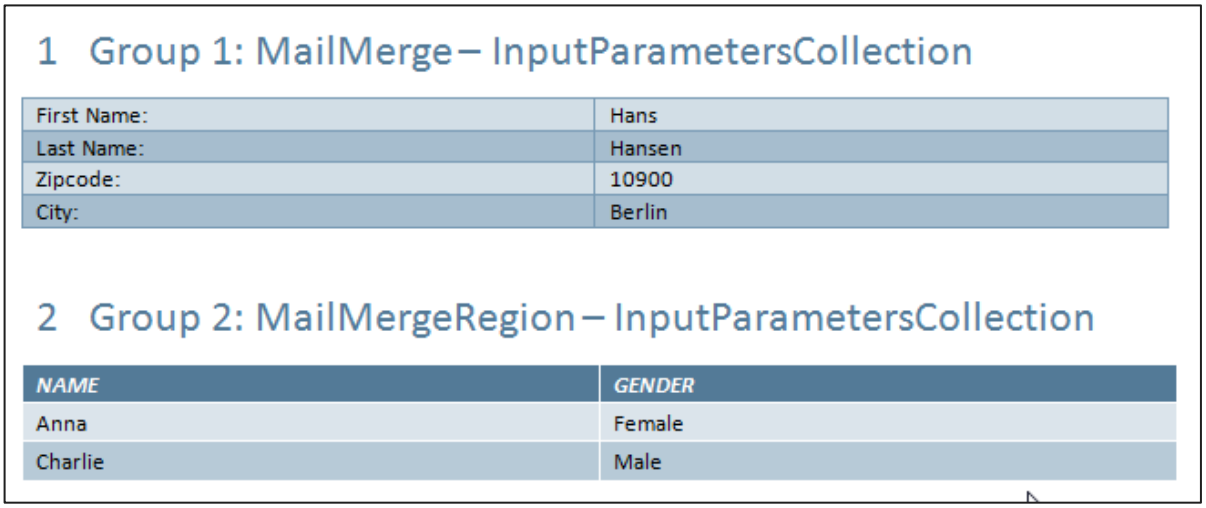
1 Group 1: MailMerge – InputParametersCollection

First Name:	«FirstNameField»
Last Name:	«LastNameField»
Zipcode:	«ZipCodeField»
City:	«CityField»

2 Group 2: MailMergeRegion – InputParametersCollection

NAME	*GENDER*
«TableStart:TestAsposeRegion1»«NameField»	«GenderField»«TableEnd:TestAsposeRegion1»

Example of a completed document:



1 Group 1: MailMerge – InputParametersCollection

First Name:	Hans
Last Name:	Hansen
Zipcode:	10900
City:	Berlin

2 Group 2: MailMergeRegion – InputParametersCollection

NAME	*GENDER*
Anna	Female
Charlie	Male

4.3 Formatting data

Data coming from workflows and data sources initially follow the ISO format. This means the dates are represented in a big endian format (Year-Month-Day). For reference the US standard I middle endian (Month-Day-Year), the UK standard I little endian (Day-Month-Year). Numbers will use a period "." as decimal separator.

4.3.1 Date formatting

Dates can be formatted by inserting the following code in the Merge Field:

\@"dd/MM/yyyy"

The complete code in the field would look as follows:

{ MERGEFIELD SendDate\@"dd/MM/yyyy" * MERGEFORMAT }

4.3.2 Number formatting

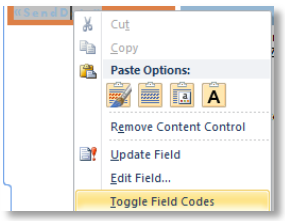
Numbers can be formatted by inserting the following code in the Merge Field:

\# #,###.00

The complete code in the field would look as follows:

{ MERGEFIELD WarantyAmount \# #,###.00 MERGEFORMAT }

To edit the formating of a field you can right click it and select “Toggle Field Codes”.



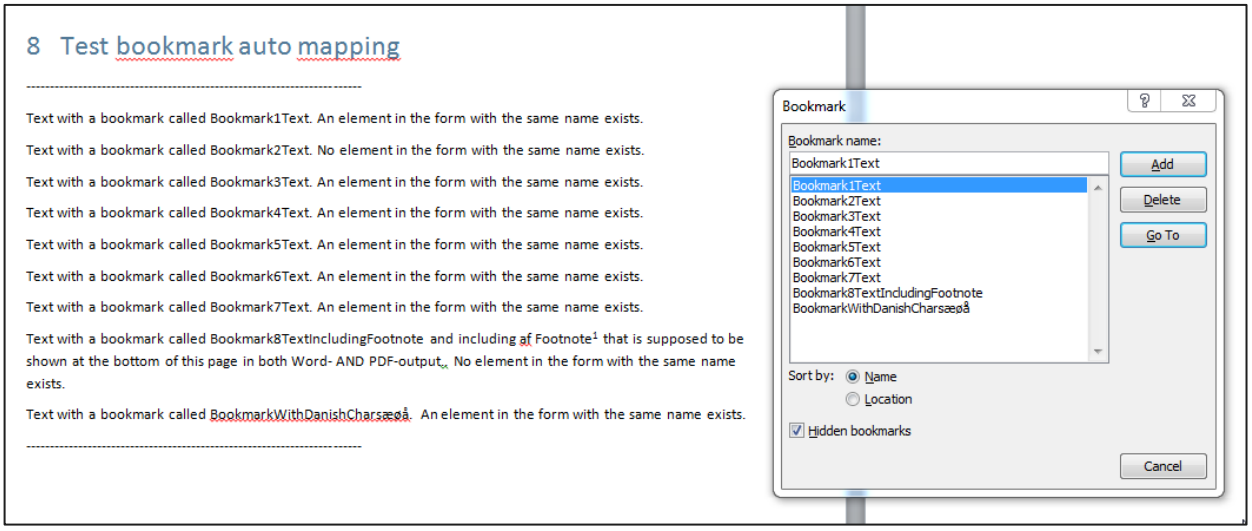
4.4 Bookmark mapping

It is possible to integrate with Word bookmarks from Process Platform data. In a Word document a section of text can be marked as a bookmark with a bookmark name. This is normally used for swift navigation within the document. From a Word Template Filling perspective the bookmarks can be overwritten by data from workflow variables.

The following rules apply:

- When a bookmark exists in a word document and there is no workflow variable with the same name as the bookmark, nothing happens
- When a bookmark exists in a word document and there is a workflow variable with the same name as the bookmark, the bookmark is overwritten with the value of the workflow variable. Both in the situation where the workflow variable contains a text and in the situation where it contains an empty string.
    - In the latter situation the bookmark is as such deleted.

To insert a bookmark in a document, mark a section of text and choose Insert, Bookmark.



Give the bookmark the name of the workflow variable that is to be overwritten and select Add.

There is nothing to specify in the Export Definition for the Bookmark mapping – see Creating Export Definitions for details about Export Definitions.

Bookmark naming is case insensitive. That is a variable with the name “BOOKMark1Text” will overwrite a bookmark with the name “Bookmark1Text”.

5 Creating Export Definitions

An Export Definition is a mapping file defining which data are to be exported from workflow variables or DataSources as well as which Merge Fields these data will be filled in to.

Read the “DevGuide Process Platform - Word Mail Merge Export Server Plugin” document for a detailed description of how export definitions work.

Export definitions are XML files that define the following:

- Which word template to be used.
- The format of the output – word or pdf.
- Which Word Data Sources data is to be fetched from.
- What workflow variables contain the input for the Word Data Sources.
- Which workflow variables are to fill out word templates directly.
- And what elements from the data sources are to be used to fill in the word templates Merge Fields.

5.1 The elements of an Export Definition

In this section we will go over the elements of an export definition.

```
<?xml version="1.0" encoding="utf-8"?>
<WordMailMerge xmlns="http://www.resultmaker.com/Exports/WordMailMerge">
  <Output format="docx" />
  <Input>
    <DocumentTemplate name="SoftwareTest_WordMailMerge.docx" />
    <Documents>
      <!-- Ex 1 : MailMerge - InputParameterCollection -->
      <Document type="fixedtext" documentName="docs://MailMerge">
        <DataSource type="InputParamatersCollection">
          <OutputMapping>
            <Map datasourcename="TestProcess:FirstName" wordmergefieldname="FirstNameField"></Map>
            <Map datasourcename="TestProcess:LastName" wordmergefieldname="LastNameField"></Map>
            <Map datasourcename="TestProcess:ZipCode" wordmergefieldname="ZipCodeField"></Map>
            <Map datasourcename="TestProcess:City" wordmergefieldname="CityField"></Map>
          </OutputMapping>
        </DataSource>
      </Document>
  </Documents>
</Input>
</WordMailMerge>
```

5.1.1 Output format

Ouput specifies the output format: Legal value sare: doc and docx
Examples:

```
<Output format="docx" />
or
<Output format="doc" />
or
<Output format="pdf" />
```

5.1.2 DocumentTemplate

DocumentTemplate specifies the name of the Word document which is to be used as a template for the filled document.
Example:

```
  <DocumentTemplate name="SoftwareTest_WordMailMerge.docx" />
```

5.1.3 Document

Document specifies the type of mapping being defined.

type, is used to internally specify which type of document is being used. This should always be the *fixedtext* type.

documentName, can either hold the value ”docs://MailMerge” or ”docs://MailMergeRegion”. The attribute is used to indicate if the template filling is of the type “simple mail merge” or “mail merge region”.

Example:

```
<Document type="fixedtext" documentName="docs://MailMergeRegion">
or
<Document type="fixedtext" documentName="docs://MailMerge">
```

5.1.4 DataSource

That DataSource element determines what source to take the data from. Currently two types exist:

- InputParametersCollection
- Assembly

The InputParametersCollection data source type instructs the module to use the workflow variables in the current workflows for the word filling.

The Assembly data source type indicates the assembly containing data. The document “DevGuide Process Platform - Word Mail Merge Export Server Plugin.docx” contains a guide for creating these data sources. In this example we’re using word data sources called “OneRowTestDataSource_CountActiveBitsInLong” and “MultiRowTestDataSource_SplitStringByCaps”.

Examples:

```
<DataSource type="InputParamatersCollection">

or

<DataSource type="Assembly" name="TestDataSource.OneRowTestDataSource_CountActiveBitsInLong,
TestDataSource">
```

5.1.5 MethodParameters - Parameter

The ”Parameter” element under ”MethodParameters” typically maps a workflow variable to the input of the data source called. (There are a few exceptions mentioned in the “DevGuide Process Platform - Word Mail Merge Export Server Plugin.docx”)

The attribute “name” contains the name of the parameter needed for the “OneRowTestDataSource_CountActiveBitsInLong” word data source.

The attribute “variableName” is the name of the workflow variable containing the input for the data source.

Example:

```
<MethodParameters>
  <Parameter name="LongNumber" variableName="TestProcess:LongNumber"></Parameter>
</MethodParameters>
```

5.1.6 OutputMapping

The OutputMapping section contains the Map elements that map the workflow variable name to the Mail Merge Field name.

5.1.6.1 Region Mapping

For MailMergeRegions a regionName attribute must be specified. For MailMerge a regionName attribute is not required.

Example:

```
<DataSource type="Assembly"
name="TestDataSource.MultiRowTestDataSource_SplitStringByCaps, TestDataSource">
   …
  <OutputMapping regionName="TestAsposeRegion2">
</DataSource>
```

The regionName must match the name used to mark the TableStart and TableEnd fields in the word document. For this example those fields would be named «TableStart:TestAsposeRegion2» and «TableStart:TestAsposeRegion2».

5.1.6.2 AutoMapping

For MailMerge an autoMapping attribute can be specified. This will try to automatically match Word Mail Merge fields with workflow variables by name. Therefore explicit mappings are only necessary when the word fields and workflow variables have different names.

Example:

```
<DataSource type="InputParamatersCollection">
   <OutputMapping autoMapping="true"></OutputMapping>
</DataSource>
```

5.1.7 Map

The Map elements inside the OutputMapping element determine what fields from the data source are to be mapped to which Merge Fields in the word document.

Example:

```
<Map datasourcename="FirstName" wordmergefieldname="FirstName" />
<Map datasourcename="LastName" wordmergefieldname="LastName" />
```

In this case the FirstName field from the data source is filled into the Merge Field named FirstName in the word template.

It is possible to use different names. In the example below the Country value from the data source is mapped to the Merge Field names Land.

```
  <Map datasourcename="Country" wordmergefieldname="Land"></Map>
```

See further examples of Export Definitions I section “7 Appendix with examples of Export Definition”.

Mapping to a field that does not exist in the Word Template will not cause an error.

5.2 Data Export Definition Tool

Resultmaker Data Export Definition Tool is an application which is part of Process Platform Tools 6. The application's purpose is to create, edit or delete Export Definitions in the Process Platform database.

The application can be started from the "Start" menu, by selecting "All Programs->Resultmaker Process Platform 6->Process Platform Tools 6".

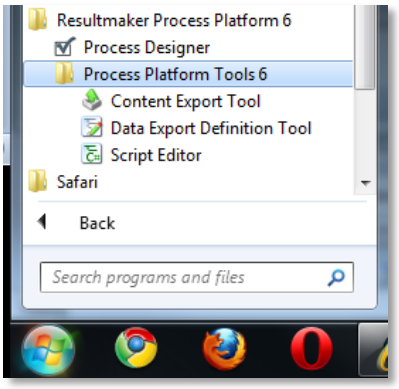
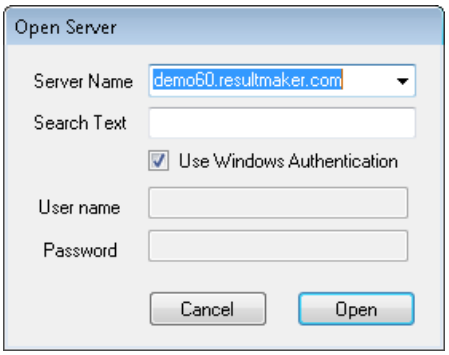
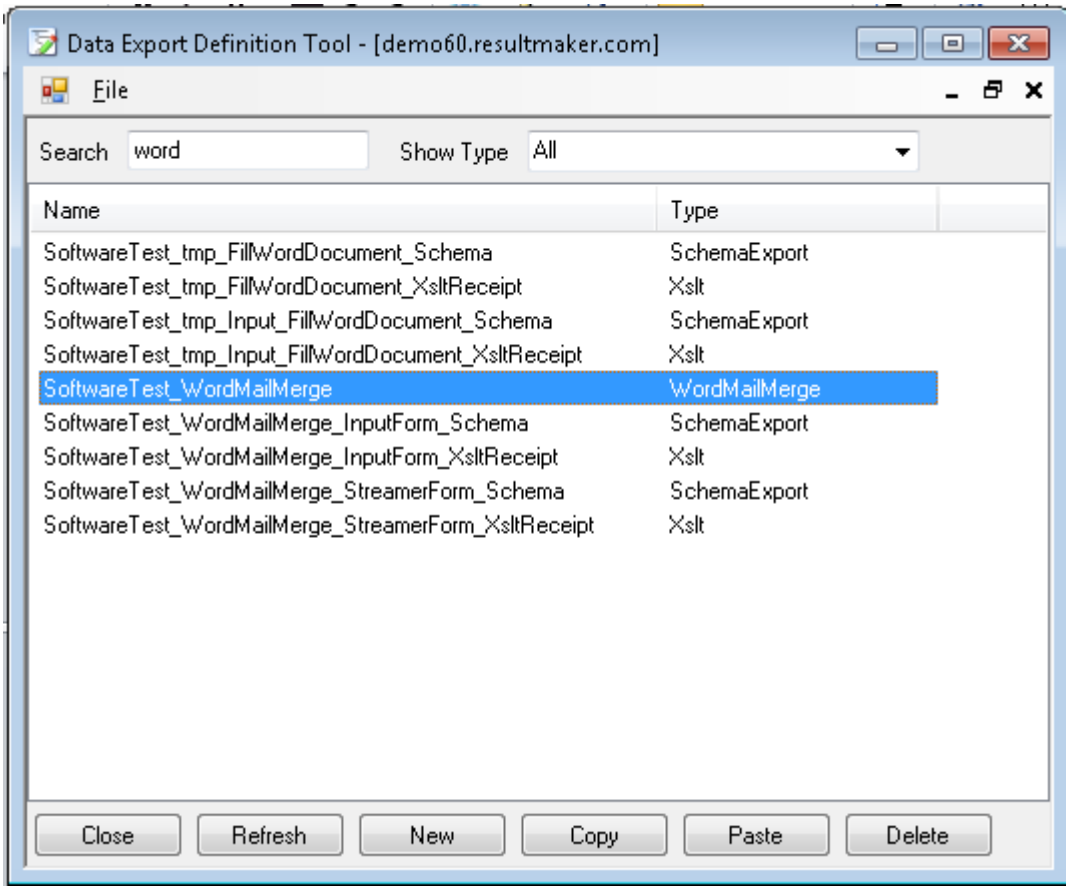


Figure 2 Start Data Export Definition Tool

Once the application is open one needs to establish a connection to the application server. This is done by selecting File->Open and supplying a Server name. Choose to "Use Windows Authentication" if you are using a Process Platform server on your local network and click Open, otherwise contact Resultmaker. If a Search Text is entered at this point it will limit the each scope of later searches (This scope can only be changed by reopening the server connection). For most purposes it may be easier not to enter a Search Text at this point.



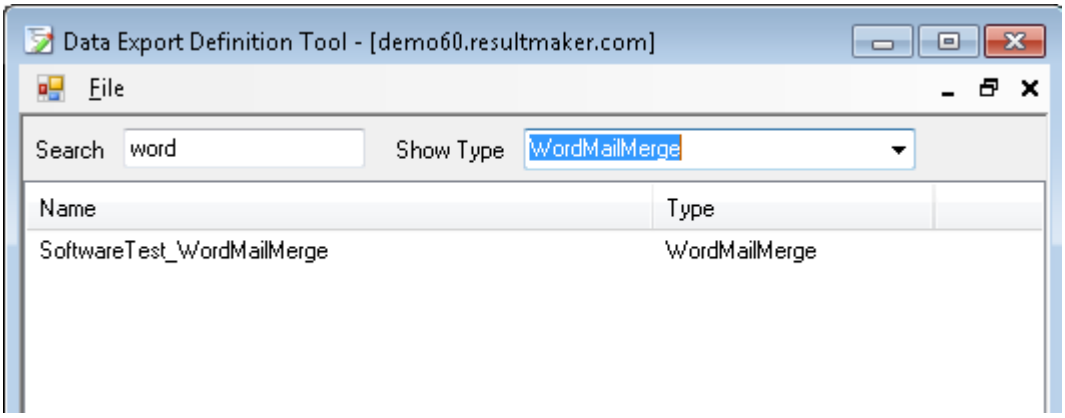
Hereafter a list of Export Definitions from the server is displayed.



In this window you have the following options:

Close	Close the search window – not the entire application.
Refresh	Refresh the list – Export Definitions that others may have saved in the meantime will then be displayed in the list.
New	Open a new window, where a new Export Definition may be edited and created.
Copy	Copy the Export Definitions selected in the list to the Clip Board. This can be used to copy export definitions in between servers.
Paste	Insert Export Definitions from the Clip Board on a server.
Delete	Deletes any selected Export Definitions from the list.

It's possible to filter by type, for instance WordMailMegre and the name of the Export Definition.



Once the desired Export Definition is found double click it to open it.

Here it's possible to edit the XML which the Export Definition is composed of and save your changes on the server. When creating a new Export Definition you need to supply a Name and a Type. The type should be WordMailMerge.

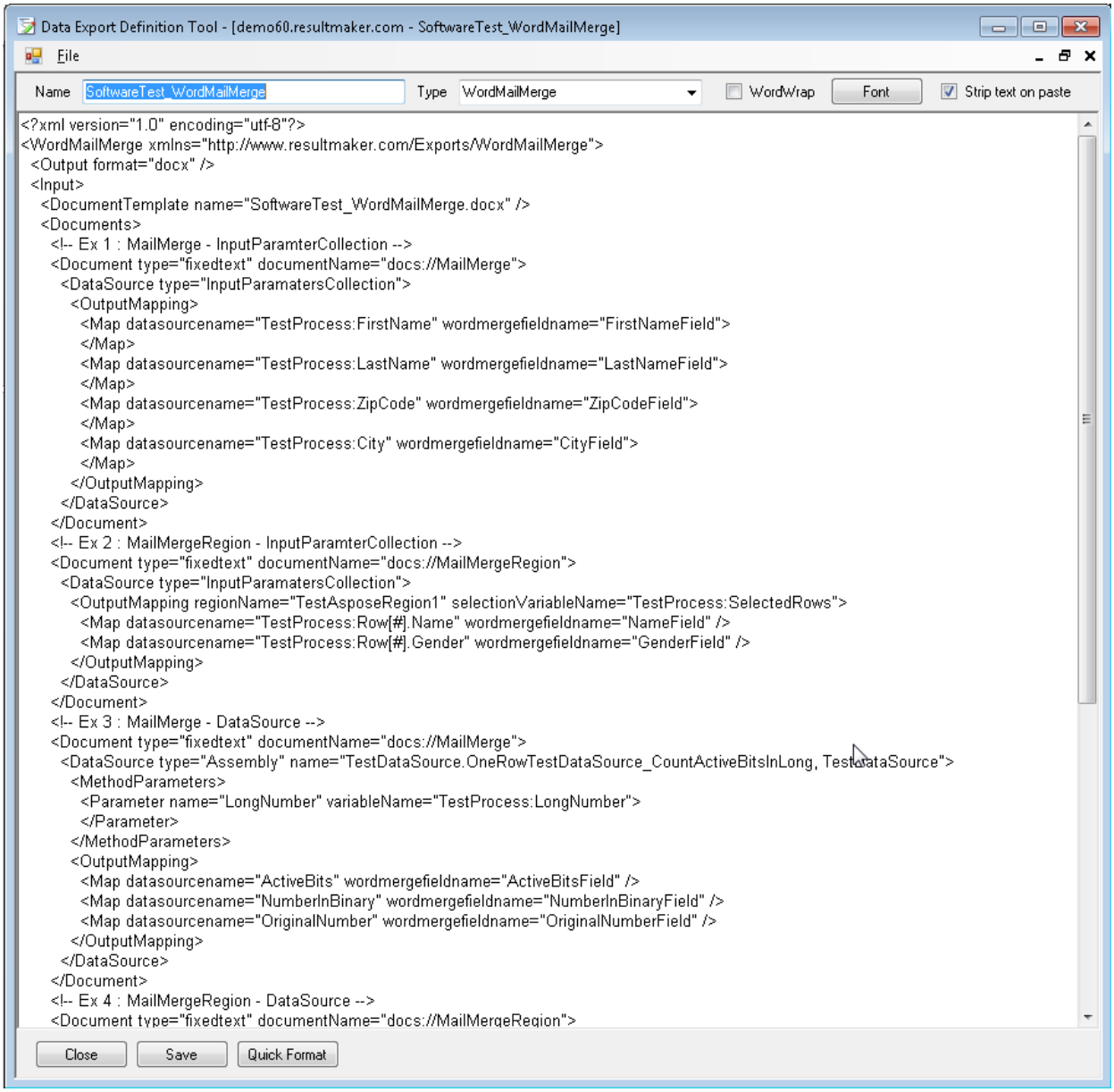
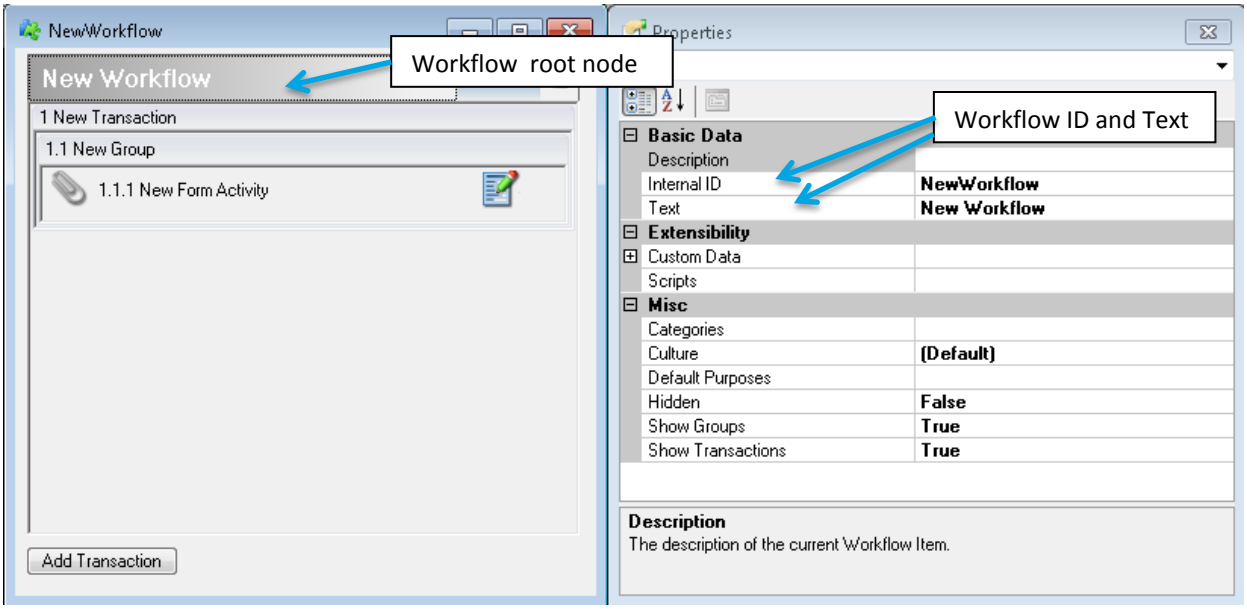


Figure 3 Export Definition opened in the Data Export Definition Tool

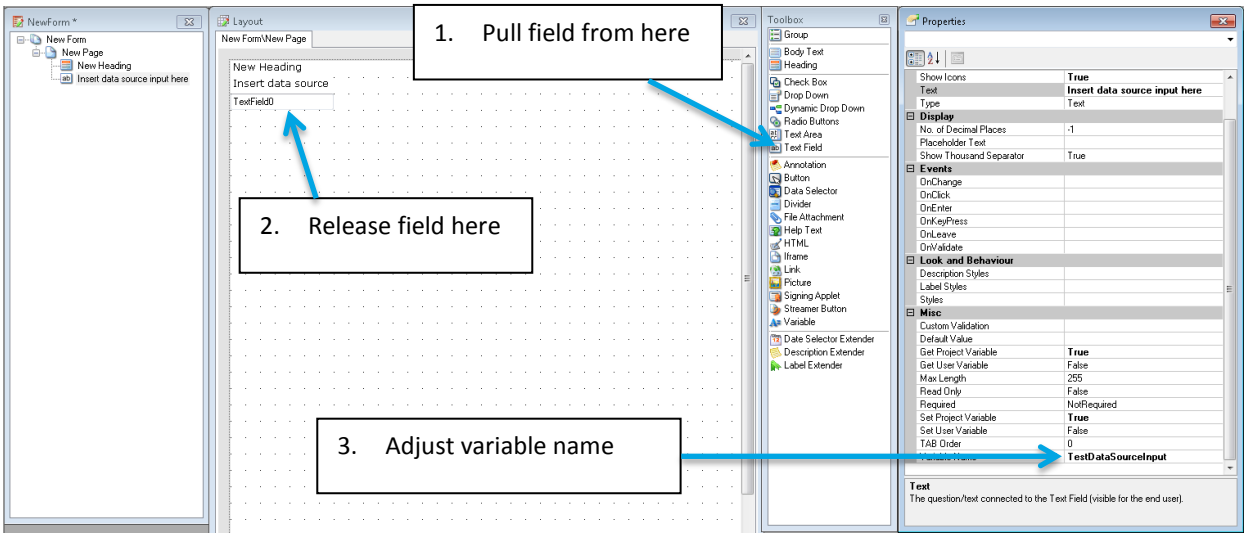
6 Calling Word template filling from a workflow

Word template filling is usually called from a workflow. This section describes how to create a simple form capable of taking some data from a data source to fill in a Word template via an Export Definition.

- Create a new workflow in the Process Designer. CTRL+N



- Give the workflow a proper ID and Text. This is done on the workflow root node.
- Create a new form to collect input data for the data source. File -> New -> Form.
- Unfold the node ”New Form” and double click ”New Page” to open the form layout



.

- Pull a ”Text Field” from the Toolbox to the page in the Layout editor.
- Change the “Variable Name” in the properties window to TestDataSourceInput
- Change the text to ”Insert data source input here”

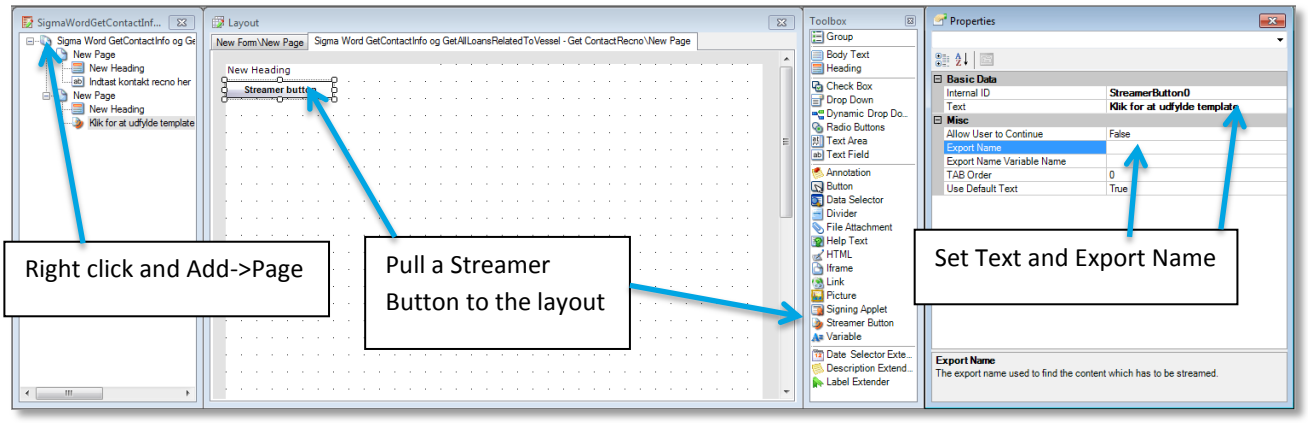
The Variable Name should match the name specified as input for the data source in the export definition. This will be specified via the Parameter element, where “name” is the name of the input for the data source and “variableName” is the name of the workflow variable:

```
<Parameter name="DataSourceInput" value="" variableName="TestDataSourceInput"></Parameter>
```

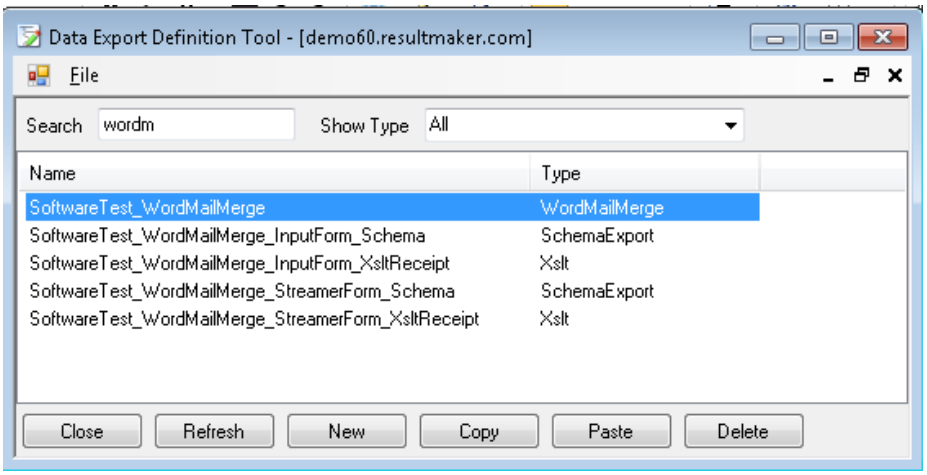
Next we insert a button called ”Streamer button”.

- Right click ”New Form” and select Add->Page

- Double click on the ”New Page”
- Pull a Streamer Button from the Toolbox to the Layout.



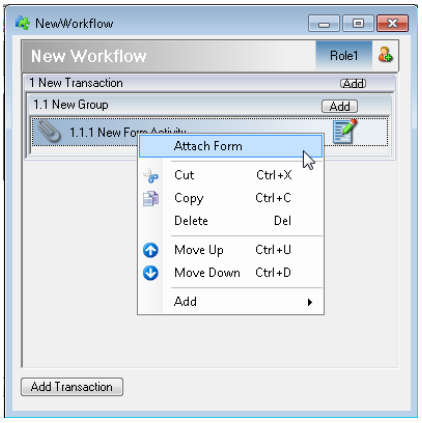
- Under Properties set Text to ”Click to fill out the template”
- Under Properties set Export Name to the name of the Export Definition you want executed when the button is pushed.
- The name can be found in the Data Export Definition Tool



- Click on New Form in the left window and correct the Properties for the form.
- Give the workflow a proper ID and Text. One you can remember.
- Save the form on the application server with CTRL+S

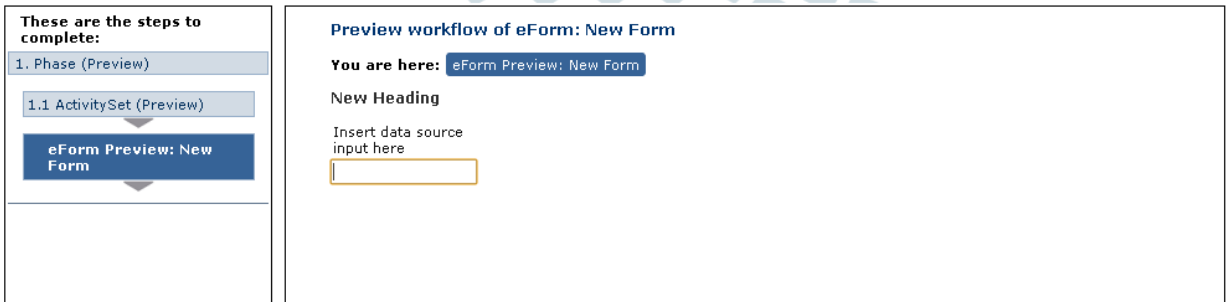
The form is now saved on the server and needs to be added to the workflow.

- Right click the activity ”New Form Activity” and select Attach Form

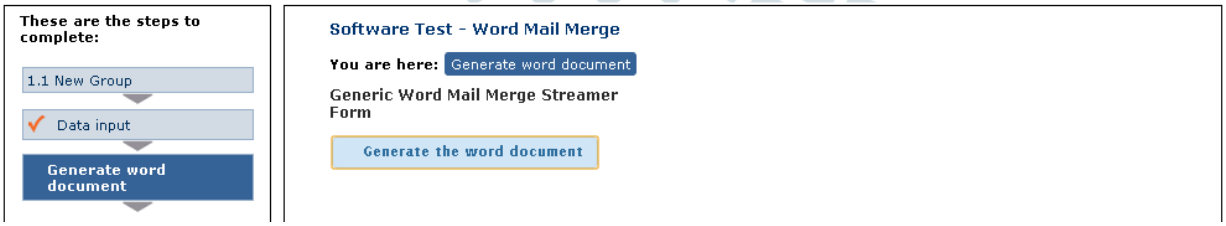


- Search for the newly created form, click on it and choose Select.
- Save the workflow and test it with F5.

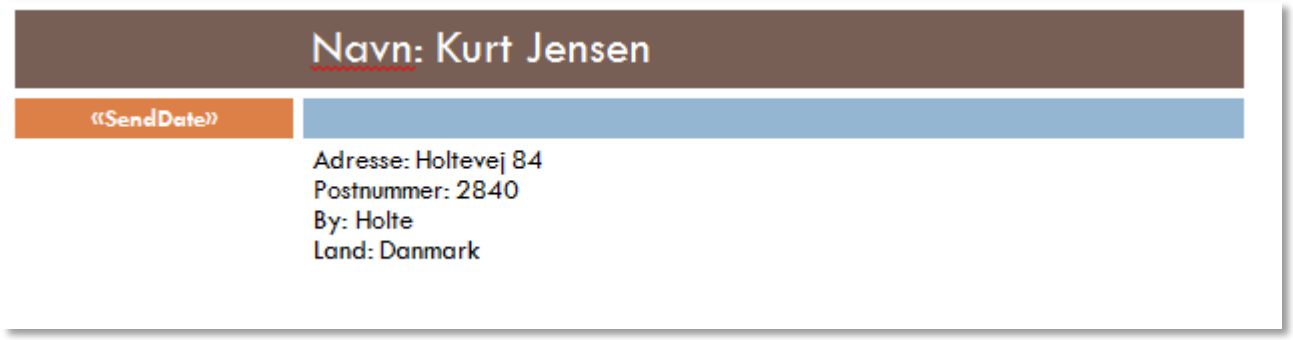
(For a good example see SoftwareTest_WordMailMerge on demo60.resultmaker.com/rmfrontend)



- Type a valid input for your data source into the field and click Continue.



Click on the streamer button to open a new window with the completed word document.



7 Appendix with examples of Export Definitions

7.1 Simple mail merge - Mapping of workflow variables word Merge Fields

```
<?xml version="1.0" encoding="utf-8"?>
<WordMailMerge xmlns="http://www.resultmaker.com/Exports/WordMailMerge">
  <!-- Output specifies the output format. Legal values are: doc, docx and pdf-->
  <Output format="docx" />
  <Input>
    <!-- DocumentTemplate specifies the name of the word document which is to be used as a
template for the final document-->
    <DocumentTemplate name="SoftwareTest_WordMailMerge.docx" />
    <Documents>
     <!-- Document specifies which type of filling is being used. MailMerge or MailMergeRegion.
This is specified in the documentName atttribute. The Type is always fixedtext.-->
      <!-- Ex 1 : MailMerge - InputParameterCollection -->
      <Document type="fixedtext" documentName="docs://MailMerge">
        <DataSource type="InputParamatersCollection">
          <OutputMapping>
            <!-- Map in OutputMapping defines the mapping between the Merge Fields in the word
               template and the output from the data source. -->
            <Map datasourcename="TestProcess:FirstName" wordmergefieldname="FirstNameField"></Map>
            <Map datasourcename="TestProcess:LastName" wordmergefieldname="LastNameField"></Map>
            <Map datasourcename="TestProcess:ZipCode" wordmergefieldname="ZipCodeField"></Map>
            <Map datasourcename="TestProcess:City" wordmergefieldname="CityField"></Map>
          </OutputMapping>
        </DataSource>
      </Document>
  </Documents>
</Input>
</WordMailMerge>
```

7.2 Mailmerge region

```
<?xml version="1.0" encoding="utf-8"?>
<WordMailMerge xmlns="http://www.resultmaker.com/Exports/WordMailMerge">
  <!-- Output specifies the output format. Legal values are: doc, docx and pdf-->
  <Output format="docx" />
  <Input>
    <!--DocumentTemplate specifies the name of the word document which is to be used as a
template for the final document-->
    <DocumentTemplate name="SoftwareTest_WordMailMerge.docx" />
    <Documents>
     <!-- Document specifies which type of filling is being used. MailMerge or MailMergeRegion.
This is specified in the documentName atttribute. The Type is always fixedtext.-->
      <!-- Ex 2 : MailMergeRegion - InputParameterCollection -->
      <Document type="fixedtext" documentName="docs://MailMergeRegion">
        <DataSource type="InputParamatersCollection">
          <OutputMapping regionName="TestAsposeRegion1" selectionVariableName="TestProcess:SelectedRows">
            <!-- Map in OutputMapping defines the mapping between the Merge Fields in the word
               template and the output from the data source. -->
            <Map datasourcename="TestProcess:Row[#].Name" wordmergefieldname="NameField" />
            <Map datasourcename="TestProcess:Row[#].Gender" wordmergefieldname="GenderField" />
          </OutputMapping>
        </DataSource>
      </Document>
  </Documents>
</Input>
</WordMailMerge>
```

7.3 Mapping workflow variables following a table structure to an “Mail Merge Region” in Word

The attribute *selectionVariableName* in the *OutputMapping* element contains a comma separated list of numbers which represent the selected rows in a table of workflow variables. (This list does not necessarily have to contain numbers, but this would usually be the scenario at hand)

in the *datasourcename* attribute of the mapping represents the location where the values from *selectionVariableName* will be inserted.

For each value in the list all the mapped workflow variables will be inserted with the value in question replacing the # sign.

The # mark may only be inserted at one point in each workflow variable.

```
<?xml version="1.0" encoding="utf-8"?>
<WordMailMerge xmlns="http://www.resultmaker.com/Exports/WordMailMerge">
  <!-- Output specifies the output format. Legal values are: doc, docx and pdf-->
  <Output format="docx" />
  <Input>
    <!--DocumentTemplate specifies the name of the word document which is to be used as a
template for the final document-->
    <DocumentTemplate name="SoftwareTest_WordMailMerge.docx" />
    <Documents>
     <!-- Document specifies which type of filling is being used. MailMerge or MailMergeRegion.
This is specified in the documentName atttribute. The Type is always fixedtext.-->
      <!-- Ex 2 : MailMergeRegion - InputParameterCollection -->
      <Document type="fixedtext" documentName="docs://MailMergeRegion">
        <DataSource type="InputParamatersCollection">
          <OutputMapping regionName="TestAsposeRegion1" selectionVariableName="TestProcess:SelectedRows">
            <!-- Map in OutputMapping defines the mapping between the Merge Fields in the word
               template and the output from the data source. -->
            <Map datasourcename="TestProcess:Row[#].Name" wordmergefieldname="NameField" />
            <Map datasourcename="TestProcess:Row[#].Gender" wordmergefieldname="GenderField" />
          </OutputMapping>
        </DataSource>
      </Document>
  </Documents>
</Input>
</WordMailMerge>
```

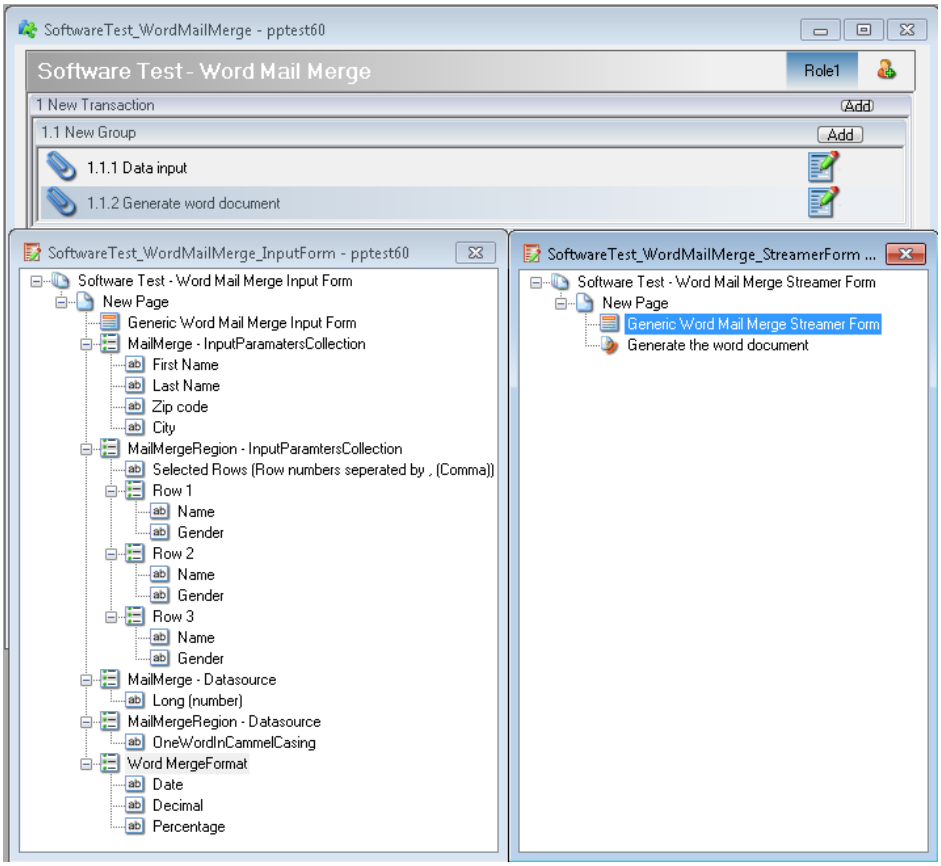
8 Word Template filling example

A good example called SoftwareTest_WordMailMerge is located on http://demo60.resultmaker.com/RMFrontEnd/default.aspx?id=SoftwareTest_WordMailMerge and should be fully functional.

Take a look at it and try it.

8.1 Workflow

The workflow consists of 2 simple forms. The first one contains the inputs for the export definition and the second one contains the streamer button calling the export definition (SoftwareTest_WordMailMerge).

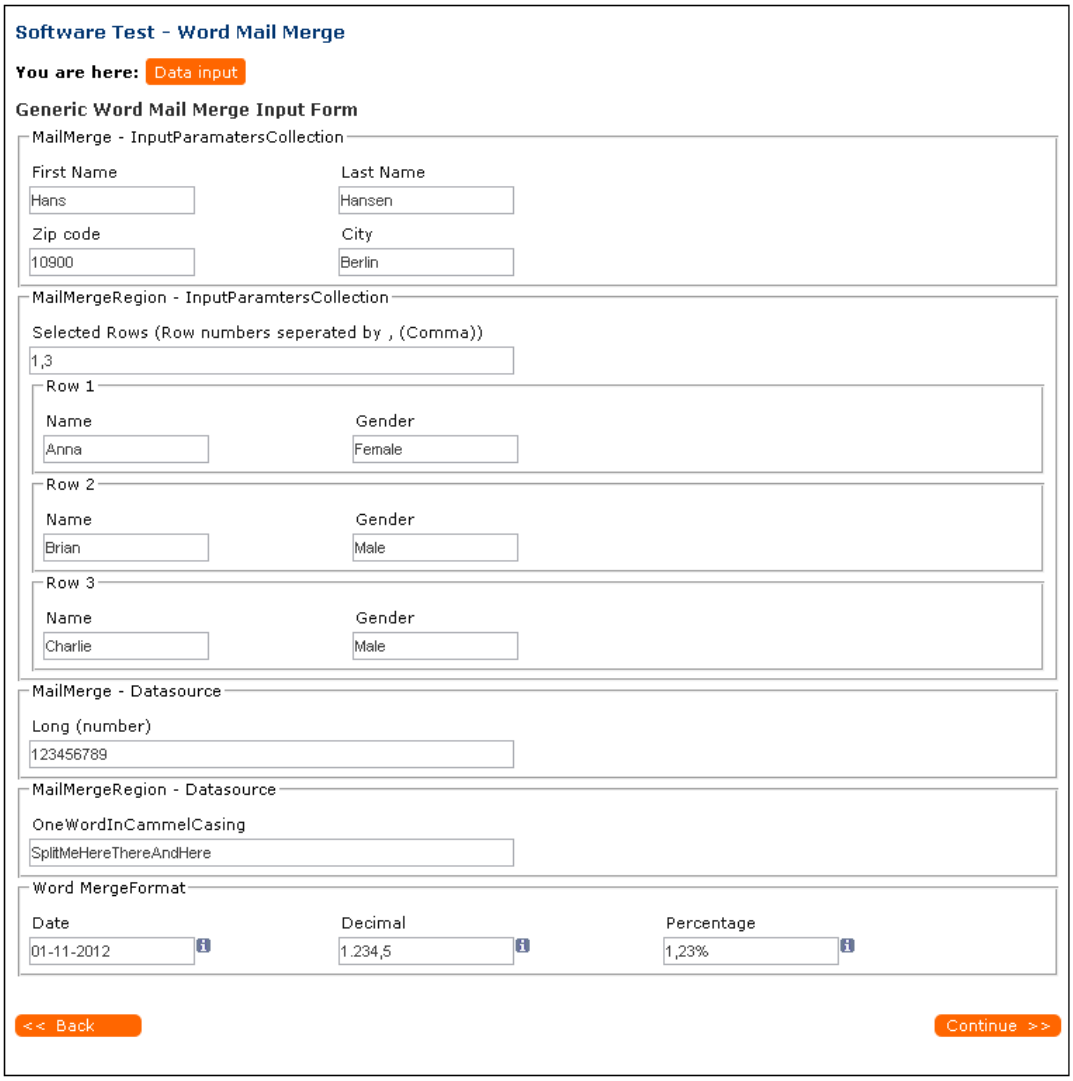


Figur 1 Workflow and forms shown in the Process Designer

The fields on the input page contain default values suited to illlustrate how the 4 different merge methods work.

There are 4 groups, one for each kind of merge:

- MailMerge – InputParamatersCollection
- MailMergeRegion – InputParamatersCollection
- MailMerge – Datasource
- MailMergeRegion – Datasource



Figur 2 The data input page in the Process Platform

8.2 Export Definition

The Export Definition SoftwareTest_WordMailMerge contains a section for each of the 4 previously mentioned merge methods. The Export Definition uses a Data Sources TestDataSource.dll that must be placed in the bin folder of the Export Server in order for the template filling to work.

```
<?xml version="1.0" encoding="utf-8"?>
<WordMailMerge xmlns="http://www.resultmaker.com/Exports/WordMailMerge">
  <Output format="docx" />
  <Input>
    <DocumentTemplate name="SoftwareTest_WordMailMerge.docx" />
    <Documents>
      <!-- Ex 1 : MailMerge - InputParameterCollection -->
      <Document type="fixedtext" documentName="docs://MailMerge">
        <DataSource type="InputParamatersCollection">
          <OutputMapping>
            <Map datasourcename="TestProcess:FirstName" wordmergefieldname="FirstNameField"></Map>
            <Map datasourcename="TestProcess:LastName" wordmergefieldname="LastNameField"></Map>
            <Map datasourcename="TestProcess:ZipCode" wordmergefieldname="ZipCodeField"></Map>
            <Map datasourcename="TestProcess:City" wordmergefieldname="CityField"></Map>
          </OutputMapping>
        </DataSource>
      </Document>
      <!-- Ex 2 : MailMergeRegion - InputParameterCollection -->
      <Document type="fixedtext" documentName="docs://MailMergeRegion">
        <DataSource type="InputParamatersCollection">
```

```
          <OutputMapping regionName="TestAsposeRegion1"
selectionVariableName="TestProcess:SelectedRows">
            <Map datasourcename="TestProcess:Row[#].Name" wordmergefieldname="NameField" />
            <Map datasourcename="TestProcess:Row[#].Gender" wordmergefieldname="GenderField" />
          </OutputMapping>
        </DataSource>
      </Document>
      <!-- Ex 3 : MailMerge - DataSource -->
      <Document type="fixedtext" documentName="docs://MailMerge">
        <DataSource type="Assembly" name="TestDataSource.OneRowTestDataSource_CountActiveBitsInLong,
TestDataSource">
          <MethodParameters>
            <Parameter name="LongNumber" variableName="TestProcess:LongNumber">
            </Parameter>
          </MethodParameters>
          <OutputMapping>
            <Map datasourcename="ActiveBits" wordmergefieldname="ActiveBitsField" />
            <Map datasourcename="NumberInBinary" wordmergefieldname="NumberInBinaryField" />
            <Map datasourcename="OriginalNumber" wordmergefieldname="OriginalNumberField" />
          </OutputMapping>
        </DataSource>
      </Document>
      <!-- Ex 4 : MailMergeRegion - DataSource -->
      <Document type="fixedtext" documentName="docs://MailMergeRegion">
        <DataSource type="Assembly" name="TestDataSource.MultiRowTestDataSource_SplitStringByCaps,
TestDataSource">
          <MethodParameters>
            <Parameter name="CammelCaseString" variableName="TestProcess:CammelCaseString"></Parameter>
          </MethodParameters>
          <OutputMapping regionName="TestAsposeRegion2">
            <Map datasourcename="PieceNumber" wordmergefieldname="PieceNumberField" />
            <Map datasourcename="PieceValue" wordmergefieldname="PieceValueField" />
            <Map datasourcename="OriginalValue" wordmergefieldname="OriginalValueField" />
          </OutputMapping>
        </DataSource>
      </Document>
      <Document type="fixedtext" documentName="docs://MailMerge">
        <DataSource type="InputParamatersCollection">
          <OutputMapping>
            <Map datasourcename="TestProcess:Date" wordmergefieldname="TestDateField" />
            <Map datasourcename="TestProcess:Decimal" wordmergefieldname="TestDecimalField" />
            <Map datasourcename="TestProcess:Percentage" wordmergefieldname="TestPercentField" />
          </OutputMapping>
        </DataSource>
      </Document>
      <Document type="fixedtext" documentName="docs://MailMerge">
        <DataSource type="InputParamatersCollection">
          <OutputMapping autoMapping="false">
            <Map datasourcename="Project.Text" wordmergefieldname="Project.Text" />
          </OutputMapping>
        </DataSource>
      </Document>
    </Documents>
  </Input>
</WordMailMerge>
```

8.3 Word Template

Below is shown the content of the Word template used by the export definition GenericTestDataSourceDocument.

Group 1: MailMerge – InputParametersCollection

First Name:	«FirstNameField»
Last Name:	«LastNameField»
Zipcode:	«ZipCodeField»
City:	«CityField»

Group 2: MailMergeRegion – InputParametersCollection

Name	*Gender*
«TableStart:TestAsposeRegion1»«NameField»	«GenderField»«TableEnd:TestAsposeRegion1»

Group 3: MailMerge – DataSource

Active bits (Number of bits with the value 1):	«ActiveBitsField»
Binary representation of the number:	«NumberInBinaryField»
Original representation of the number:	«OriginalNumberField»

Group 4: MailMergeRegion – DataSource

NUMBER	*SPLIT STRING PIECE VALUE*	*ORIGINAL STRING*
«TableStart:TestAsposeRegion2»«PieceNumberField»	«PieceValueField»	«OriginalValueField»«TableEnd:TestAsposeRegion2»

8.4 Filled in Word Template

Once filled in (Using the test workflow called SoftwareTest_WordMailMerge with the default values) the document would look as follows:

Group 1: MailMerge – InputParametersCollection

First Name:	Hans
Last Name:	Hansen
Zipcode:	10900
City:	Berlin

Group 2: MailMergeRegion – InputParametersCollection

Name	*Gender*
Anna	Female
Charlie	Male

Group 3: MailMerge – DataSource

Active bits (Number of bits with the value 1):	16
Binary representation of the number:	11101011011110011010010101
Original representation of the number:	123456789

Group 4: MailMergeRegion – DataSource

NUMBER	*SPLIT STRING PIECE VALUE*	*ORIGINAL STRING*
1	Split	SplitMeHereThereAndHere
2	Me	SplitMeHereThereAndHere
3	Here	SplitMeHereThereAndHere
4	There	SplitMeHereThereAndHere
5	And	SplitMeHereThereAndHere
6	Here	SplitMeHereThereAndHere

Group 5: Test MergeFormat

Date (d. MMMM yyyy):	1. november 2012
Decimal (#,##0.00):	1.234,50
Percentage (MergeField inside a FormulaField):	1,23

Group 6: Test InputParametersCollection auto mapping

First Name (with a colon):	Hans
Project text (with a dot):	Software Test - Word Mail Merge
Gender (with a dot and brackets):	Male
Logged in (plain):	1

Group 7: Test InputParametersCollection multiple fields with the same name

Project text:	Software Test - Word Mail Merge
Project text again:	Software Test - Word Mail Merge

8.5 Assembly Data Sources

Data sources of type Assembly are used to fetch data outside the Process Platform. This is not the typical use of Export Server plugins and is not recommended to use. Instead ensure that the data are placed in the workflow variables and use the `InputParamatersCollection.`

The Export Definition also uses the following two 2 data sources. Se the document “DevGuide Process Platform - Word Mail Merge Export Server Plugin” that describes how to develop a Data Source.