

---

Guide for administrators

# officeatwork Output Method Instructions (OOMI) Guide

Version 3.7



This guide has been created using officeatwork Advanced.  
5. February 2009

LIB Consultants AG has prepared this manual with the greatest possible care so as to ensure that the information contained herein is easy to understand, accurate and reliable. Nevertheless, LIB Consultants AG is in no way liable for any issues which have any connection with this manual, including – and without restriction – its standard quality and availability for special purposes. From time to time, LIB Consultants AG will revise the software described in this manual and reserves the right to do so without prior advice to the customer. Under no circumstances is LIB Consultants AG liable for indirect, special or incidental damages resulting from the purchase or use of this manual or the information contained herein. This guarantee exclusion has no impact on the statutory rights of the user.

Copyright© 1992-2008 LIB Consultants AG, Switzerland.  
All rights reserved.

officeatwork® is a registered trademark of LIB Consultants AG  
Microsoft® Word, Microsoft® Office, Windows®, Windows 95™, Windows 98™, Windows NT®, Windows Vista and MS-DOS™ are trademarks of the Microsoft Corporation.  
Other names of companies, products or services may be trademarks or registered trademarks of the respective owners.

LIB Consultants AG  
Bundesplatz 12  
CH-6300 Zug  
Switzerland

H: 0900 566 088 (Hotline, for Switzerland only, CHF 3.50 per minute, price subject to change)

T: +41 (0)41 763 16 70

F: +41 (0)41 763 16 75

E-mail: [mail@officeatwork.com](mailto:mail@officeatwork.com)

Internet: [www.officeatwork.com](http://www.officeatwork.com)

# Table of Contents

<b>About this guide</b>	<b>5</b>
For whom is the guide intended.....	5
What is covered in this guide.....	5
Knowledge required.....	5
Typographic conventions.....	5
Feedback.....	6
General contact details.....	6
<b>Support</b>	<b>7</b>
Help.....	7
Manuals.....	7
Support.....	8
<b>Concept</b>	<b>11</b>
Introduction.....	11
Description of the steps in sequence.....	12
Basic XML Structure overview.....	12
<b>Definitions</b>	<b>15</b>
XML Definitions.....	16
Parameter definitions.....	19
ActiveFileName().....	19
ActivePath().....	20
Cancel().....	20
Changed().....	21
DocumentLanguage ().....	22
HostApplication().....	23
MasterProperties ().....	24
OutputFileName().....	26
OutputPath().....	27
WordBookmarks ().....	27
WordBuiltInDocumentProperties ().....	29
WordCustomDocumentProperties ().....	30
WordVariables ().....	31
Variables.....	32
<b>Creating OOMI-Files</b>	<b>35</b>
Creation.....	35
<b>Integrating OOMI-Files</b>	<b>37</b>
Integration into an officeatwork solution.....	37
Integration in output variations.....	37
<b>Samples</b>	<b>39</b>

Sample 1: Adobe Acrobat (Save PDF) .....	39
Sample 2: Adobe Acrobat (Send PDF).....	39
Sample 3: PDF Creator .....	40
Sample 4: BullZip PDF.....	41
Sample 5: PDF Factory Save .....	42
Sample 6: PDF Factory Send.....	43
Sample 7: FreePDF XP Save .....	44
Sample 8: FreePDF XP Send.....	45

---

**Appendix** **47**

LCID's.....	48
OutputMethodInstruction schema.....	51

---

**Index** **53**

---

---

## CHAPTER 1

# About this guide

The first chapter explains the structure of the guide and its intended users.

---

## For whom is the guide intended

This guide is intended for Administrators who want to integrate or modify any printer-driver integration into their officeatwork solution.

---

## What is covered in this guide

The guide illustrates the OOMI-files concept of officeatwork. It gives you an overview of the architecture and shows how to modify or extend the officeatwork output management with your own printer drivers through configuration of OOMI-files.

---

## Knowledge required

To understand this guide, a good general knowledge of computing as well as basic knowledge regarding XML is needed.

---

## Typographic conventions

Before reading this guide, you should be familiar with the typographic conventions used.

The following graphic descriptions highlight sections of text with particular significance.

<u>Formatting Convention</u>	<u>Type of Information</u>
Triangle ➤	Step-by-step procedure. You can follow these instructions to perform a specific task.
<b>Bold Typeface</b>	Objects needed for selection, such as menus, buttons or items in a list.

CAPITAL LETTERS

Key legends on the keyboard. For example SHIFT, CTRL or ALT.

KEY+KEY

Key combinations which must be pressed at the same time are marked with +. Examples: CTRL+P or ALT+F4.

---

## Feedback

We are very interested in your opinion. We kindly ask you to present us with any feedback you have, as well as with any other aspects of officeatwork, by writing an e-mail to [feedback@officeatwork.com](mailto:feedback@officeatwork.com).

## General contact details

LIB Consultants AG  
Bundesplatz 12  
6300 Zug  
Switzerland

H: 0900 566 088 (Hotline, for Switzerland only, CHF 3.50 per minute, Price subject to change)

T: +41 (0)41 763 16 70

F: +41 (0)41 763 16 75

E-Mail: [mail@officeatwork.com](mailto:mail@officeatwork.com)

Internet: [www.officeatwork.com](http://www.officeatwork.com)

---

## CHAPTER 2

# Support

---

## Help

### **officeatwork**

For all questions relating to officeatwork, please contact our service desk: Telephone 0900 566 088 (CHF 3.50 per minute).

Please also note the wide range of support and service options at [www.officeatwork.com](http://www.officeatwork.com) - Support.

---

## Manuals

officeatwork manuals are divided into the following categories:

### **Manuals for Users**

- officeatwork 20-Minute Guide
- officeatwork User Manual\*
- officeatwork Smart-Template Manager Manual
- officeatwork Smart-Content Manager Manual

### **Manuals for Administrators**

- officeatwork Installation Guide
- officeatwork Master-Template Manager Manual\*\*
- officeatwork Solution Manager Manual\*\*
- officeatwork Output Method Instructions (OOMI) Guide

### **Manuals for Project Managers**

- officeatwork Project Manager Manual\*
- officeatwork Project Preparation Manual
- officeatwork Sample Record Book

### **Manuals for Developers**

- officeatwork API

\* some chapters published  
\*\* not yet published

The manuals can be downloaded from the officeatwork website  
[www.officeatwork.com](http://www.officeatwork.com) – Download

---

## Support

Support for officeatwork products can be obtained from your officeatwork project partner or from the producer itself.

The producer (LIB Consultants AG) offers the following support options:

### **Hotline 0900 566 088**

Talk directly to our qualified officeatwork specialists. For personal support we recommend that you use our online-support-tool. The tool can be used without having to install it. The link to download the tool can be found on our website at [www.officeatwork.com](http://www.officeatwork.com) -> Support.

### **Premium officeatwork support**

Send your questions by e-mail to [premiumsupport@officeatwork.com](mailto:premiumsupport@officeatwork.com) or call us at +41 (0)41 763 16 73 (LIB Consultants AG). Your questions will be answered by our officeatwork specialists with priority. If desired, your officeatwork solution will be analysed by our officeatwork specialists. If necessary, changes to your solution can be made at your request. For more information check out our website at [www.officeatwork.com](http://www.officeatwork.com) -> Support.

### **Free officeatwork E-Mail support**

Send us your questions by e-mail to [support@officeatwork.com](mailto:support@officeatwork.com). This support is free. Your questions will be answered by our support department on average within 5 to 10 working days. For more information check out our website at [www.officeatwork.com](http://www.officeatwork.com) -> Support.

### **1 Incident**

You receive telephone support per incidence at a fixed price. Per single incident you will receive telephone support until the conclusion of your incident has been reached. For more information check out our website at [www.officeatwork.com](http://www.officeatwork.com) -> Support.

### **5 Incidents**

5 incidents are the equivalent to five single incidents. With the 5-incident package you will receive five incidents for the price of four. For more information check out our website at [www.officeatwork.com](http://www.officeatwork.com) -> Support.

### **Unlimited support per month**

You are provided with unlimited telephone support for one full month at a fixed price. For more information check out our website at [www.officeatwork.com](http://www.officeatwork.com) -> Support.

### **Manuals**

You can download the most up-to-date manuals at [www.officeatwork.com](http://www.officeatwork.com) -> Download.

### **Newsletter**

officeatwork-News is "good news" with added value. officeatwork-News informs you regularly about relevant topics regarding «corporate office automation». The qualified news keeps you up-to-date and helps you to increase the practical value of Microsoft Office. For more information and registration check out our website at [www.officeatwork.com](http://www.officeatwork.com) -> Support.



## CHAPTER 3

# Concept

## Introduction

The officeatwork output management architecture is an open architecture based on XML definitions. The part that is in charge of the definition on how to operate the required printer driver is stored in separate files with the extension \*.oomi. OOMI stands for «Officeatwork Output Method Instruction». OOMI-Files are XML formatted text files and need to be registered in your officeatwork solution. Each output variant can have multiple OOMI-Files registered.

The following illustration shows the elements involved when creating an output using the officeatwork output-management.

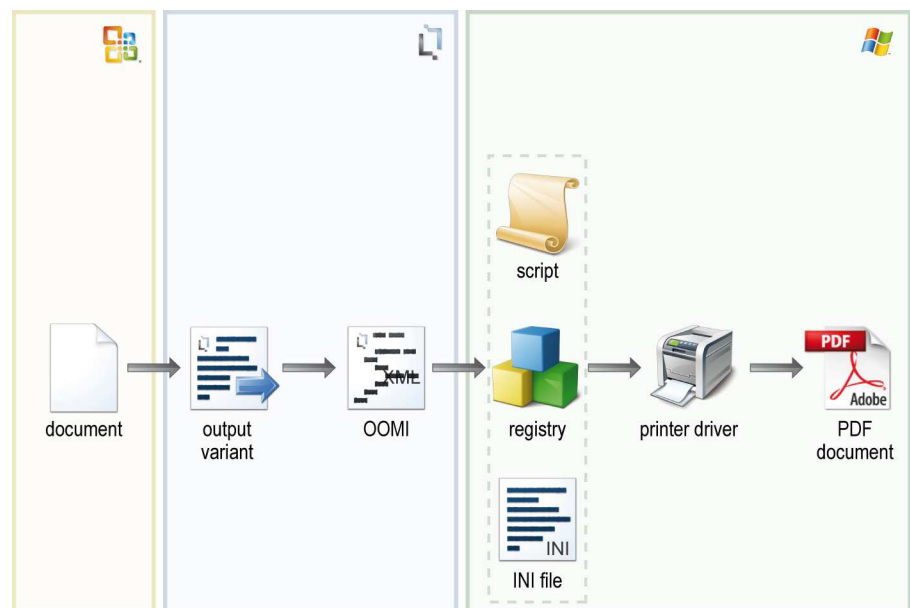


Figure 1: officeatwork OOMI-concept overview

It is for instance possible to have the output variation «Save Draft PDF» which uses an OOMI-File to create a low resolution PDF, whereas a «Save as final PDF» variation uses an OOMI-File that provides maximum resolution for all images in the PDF. You could also include the instruction in your OOMI-file to protect your PDF output with a password. It is even possible to have a variation that converts the text to a picture (only if supported by your driver) for maximum security (especially for documents like contracts).

---

## Description of the steps in sequence

When an officeatwork output variation is executed the following steps occur in sequence.

### ➤ *Steps for officatwork sending or saving variations*

- ✓ The user chooses an officeatwork sending or saving output variation.
- ✓ officeatwork changes the content of the document according to the variation.
- ✓ officeatwork checks the availability of the driver described in the oomi-file for this output variation.
- ✓ If the driver is not found, officeatwork checks the driver noted within the next OOMI-file registered for this output variation. officeatwork repeats this until it finds the desired driver on the user's PC. If no matching driver is found, the document is restored to its original state and the output process is stopped. In this case a message informs the user that the required driver could not be found for this output variation. If a driver is found, the output process is continued.
- ✓ If necessary, officeatwork switches to the currently required printer driver.
- ✓ officeatwork reads the «BeforeAction» section of the oomi-file and performs the defined instructions.
- ✓ officeatwork executes the instructions defined in the «Action» section of the oomi file.
- ✓ officeatwork executes the instructions defined in the «AfterAction» section of the oomi file.
- ✓ officeatwork restores the connected printer driver to its original state.
- ✓ officeatwork restores the document to its original state.

---

## Basic XML Structure overview

An OOMI-file is split up into three main sections, the BeforeAction, the Action and AfterAction section.

### BeforeAction

This section is used to prepare the driver before the driver is activated. This is where all driver settings are usually set with their appropriate values. At this stage the document that needs printing has already been updated and its values can be accessed and passed on to the driver. Most commonly the file name of the newly to create document is passed on here.

It is also possible to set a flag so that the changes made in the BeforeAction revert to original automatically, after the BeforeAction section has been processed.

### Action

In this section the driver is activated by printing the document within the office application.

AfterAction

This section allows you to restore the driver settings if necessary. It is also possible to receive updated information from the document.



## CHAPTER 4

# Definitions

# XML Definitions

The following sample XML File represents the complete structure of an OOMI-File:

## Sample XML File:

```
<?xml version="1.0" encoding="UTF-8"?>
<OutputMethodInstruction xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <BeforeAction RestoreSettingsAfterAction="0 | -1">
    <Registry>
      <Keys>
        <Key HKEY="HKEY_CLASSES_ROOT | HKEY_CURRENT_USER | HKEY_LOCAL_MACHINE"
Path="Software\PDFDriver\Program" >
          <Values>
            <Value Name="RunProgramAfterSaving" Value="0"/>
            <Value Name="UseAutosave" Value="1"/>
            <Value Name="AutosaveDirectory" Value="%Desktop%"/>
            <Value Name="AutosaveFilename" Value=""/>
            <Value Name="StandardTitle" Value=""/>
            <Value Name="StandardSubject" Value=""/>
          </Values>
        </Key>
      </Keys>
    </Registry>
    <IniFiles>
      <IniFile Path="%AppData%\PDFDriver" Filename="settings.ini">
        <Sections>
          <Section Name="PDFDriver">
            <Entries>
              <Entry Name="Output" Value="%Desktop%\pdf"/>
              <Entry Name="ConfirmOverwrite" Value="no"/>
              <Entry Name="ShowSettings" Value="never"/>
              <Entry Name="ShowPDF" Value="yes"/>
              <Entry Name="Title" Value=""/>
              <Entry Name="Subject" Value=""/>
            </Entries>
          </Section>
        </Sections>
      </IniFile>
    </IniFiles>
    <VBScripts>
      <RunVBScript ResultType="File | Printout" Location="%Scripts%\Script.vbs" AllowUI="0 | -1"
UseSafeSubset="0 | -1" Function="FunctionName">
        <ParameterDefinitions>
          <HostApplication/>
          <ActiveFileName/>
          <ActivePath/>
          <DocumentLanguage/>
          <Changed/>
          <MasterProperties>
            <MasterProperty IDName="Signature1">
              <Field Name="Name"/>
              <Field Name="Function"/>
            </MasterProperty>
            <MasterProperty IDName="CustomField">
              <Field Name="External"/>
              <Field Name="No"/>
              <Field Name="ShowTimes"/>
            </MasterProperty>
          </MasterProperties>
          <WordCustomDocumentProperties>
            <WordCustomDocumentProperty Name="TestProperty"/>
          </WordCustomDocumentProperties>
          <WordBuiltInDocumentProperties>
            <WordBuiltInDocumentProperty Name="Author"/>
          </WordBuiltInDocumentProperties>
          <WordVariables>
            <WordVariable Name="TestVariable"/>
          </WordVariables>
          <WordBookmarks>
            <WordBookmark Name="TestBookmark"/>
          </WordBookmarks>
        </ParameterDefinitions>
      </RunVBScript>
    </VBScripts>
    <VBAMacros>
      <RunVBAMacro ResultType="File | Printout" Function="Module.MacroName">
        <ParameterDefinitions>
          <HostApplication/>
          <ActiveFileName/>
          <ActivePath/>
          <DocumentLanguage/>
          <Changed/>
        </ParameterDefinitions>
      </RunVBAMacro>
    </VBAMacros>
  </BeforeAction>
</OutputMethodInstruction>
```

```

    <MasterProperties>
      <MasterProperty IDName="Signature1">
        <Field Name="Name" />
        <Field Name="Function" />
      </MasterProperty>
      <MasterProperty IDName="CustomField">
        <Field Name="External" />
        <Field Name="No" />
        <Field Name="ShowTimes" />
      </MasterProperty>
    </MasterProperties>
  <WordCustomDocumentProperties>
    <WordCustomDocumentProperty Name="TestProperty" />
  </WordCustomDocumentProperties>
  <WordBuiltInDocumentProperties>
    <WordBuiltInDocumentProperty Name="Author" />
  </WordBuiltInDocumentProperties>
  <WordVariables>
    <WordVariable Name="TestVariable" />
  </WordVariables>
  <WordBookmarks>
    <WordBookmark Name="TestBookmark" />
  </WordBookmarks>
</ParameterDefinitions>
</RunVBAMacro>
</VBAMacro>
</BeforeAction>
<Action>
  <Print ResultFilename="%Desktop%\%.pdf" PrinterName="Printer driver name as displayed in the Microsoft Office Print Dialog" />
  <Save ResultFilename="%Desktop%\%.pdf" Filetype="any file type available from Microsoft Office" />
  <RunVBScript ResultType="File | Printout" Location="%Scripts%\Script.vbs" AllowUI="0 | -1" UseSafeSubset="0 | -1" Function="FunctionName">
    <ParameterDefinitions>
      <HostApplication />
      <ActiveFileName />
      <ActivePath />
      <DocumentLanguage />
      <Changed />
      <MasterProperties>
        <MasterProperty IDName="Signature1">
          <Field Name="Name" />
          <Field Name="Function" />
        </MasterProperty>
        <MasterProperty IDName="CustomField">
          <Field Name="External" />
          <Field Name="No" />
          <Field Name="ShowTimes" />
        </MasterProperty>
      </MasterProperties>
      <WordCustomDocumentProperties>
        <WordCustomDocumentProperty Name="TestProperty" />
      </WordCustomDocumentProperties>
      <WordBuiltInDocumentProperties>
        <WordBuiltInDocumentProperty Name="Author" />
      </WordBuiltInDocumentProperties>
      <WordVariables>
        <WordVariable Name="TestVariable" />
      </WordVariables>
      <WordBookmarks>
        <WordBookmark Name="TestBookmark" />
      </WordBookmarks>
    </ParameterDefinitions>
  </RunVBScript>
</RunVBAMacro ResultType="File | Printout" Function="Module.MacroName">
  <ParameterDefinitions>
    <HostApplication />
    <ActiveFileName />
    <ActivePath />
    <DocumentLanguage />
    <Changed />
    <MasterProperties>
      <MasterProperty IDName="Signature1">
        <Field Name="Name" />
        <Field Name="Function" />
      </MasterProperty>
      <MasterProperty IDName="CustomField">
        <Field Name="External" />
        <Field Name="No" />
        <Field Name="ShowTimes" />
      </MasterProperty>
    </MasterProperties>
    <WordCustomDocumentProperties>
      <WordCustomDocumentProperty Name="TestProperty" />
    </WordCustomDocumentProperties>
    <WordBuiltInDocumentProperties>
      <WordBuiltInDocumentProperty Name="Author" />
    </WordBuiltInDocumentProperties>
  </ParameterDefinitions>

```

```

    <WordVariables>
      <WordVariable Name="TestVariable"/>
    </WordVariables>
    <WordBookmarks>
      <WordBookmark Name="TestBookmark"/>
    </WordBookmarks>
  </ParameterDefinitions>
</RunVBAMacro>
</Action>
<AfterAction>
  <Registry>
    <Keys>
      <Key HKEY="HKEY_CLASSES_ROOT | HKEY_CURRENT_USER | HKEY_LOCAL_MACHINE"
Path="Software\PDFDriver\Program" >
      <Values>
        <Value Name="RunProgramAfterSaving" Value="0"/>
        <Value Name="UseAutosave" Value="1"/>
        <Value Name="AutosaveDirectory" Value="%Desktop%"/>
        <Value Name="AutosaveFilename" Value=""/>
        <Value Name="StandardTitle" Value=""/>
        <Value Name="StandardSubject" Value=""/>
      </Values>
    </Key>
  </Keys>
</Registry>
<IniFiles>
  <IniFile Path="%AppData%\PDFDriver" Filename="settings.ini">
    <Sections>
      <Section Name="PDFDriver">
        <Entries>
          <Entry Name="Output" Value="%Desktop%\ .pdf"/>
          <Entry Name="ConfirmOverwrite" Value="no"/>
          <Entry Name="ShowSettings" Value="never"/>
          <Entry Name="ShowPDF" Value="yes"/>
          <Entry Name="Title" Value=""/>
          <Entry Name="Subject" Value=""/>
        </Entries>
      </Section>
    </Sections>
  </IniFile>
</IniFiles>
<VBScripts>
  <RunVBScript ResultType="File | Printout" Location="%Scripts%\Script.vbs" AllowUI="0 | -1"
UseSafeSubset="0 | -1" Function="FunctionName">
    <ParameterDefinitions>
      <HostApplication/>
      <ActiveFileName/>
      <ActivePath/>
      <DocumentLanguage/>
      <Changed/>
      <MasterProperties>
        <MasterProperty IDName="Signature1">
          <Field Name="Name" />
          <Field Name="Function" />
        </MasterProperty>
        <MasterProperty IDName="CustomField">
          <Field Name="External" />
          <Field Name="No" />
          <Field Name="ShowTimes" />
        </MasterProperty>
      </MasterProperties>
      <WordCustomDocumentProperties>
        <WordCustomDocumentProperty Name="TestProperty"/>
      </WordCustomDocumentProperties>
      <WordBuiltInDocumentProperties>
        <WordBuiltInDocumentProperty Name="Author" />
      </WordBuiltInDocumentProperties>
      <WordVariables>
        <WordVariable Name="TestVariable"/>
      </WordVariables>
      <WordBookmarks>
        <WordBookmark Name="TestBookmark"/>
      </WordBookmarks>
    </ParameterDefinitions>
  </RunVBScript>
</VBScripts>
<VBAMacros>
  <RunVBAMacro ResultType="File | Printout" Function="Module.MacroName">
    <ParameterDefinitions>
      <HostApplication/>
      <ActiveFileName/>
      <ActivePath/>
      <DocumentLanguage/>
      <Changed/>
      <MasterProperties>
        <MasterProperty IDName="Signature1">
          <Field Name="Name" />
          <Field Name="Function" />
        </MasterProperty>
      </MasterProperties>
    </ParameterDefinitions>
  </RunVBAMacro>
</VBAMacros>

```

```

    </MasterProperty>
    <MasterProperty IDName="CustomField">
      <Field Name="External" />
      <Field Name="No" />
      <Field Name="ShowTimes" />
    </MasterProperty>
  </MasterProperties>
  <WordCustomDocumentProperties>
    <WordCustomDocumentProperty Name="TestProperty" />
  </WordCustomDocumentProperties>
  <WordBuiltInDocumentProperties>
    <WordBuiltInDocumentProperty Name="Author" />
  </WordBuiltInDocumentProperties>
  <WordVariables>
    <WordVariable Name="TestVariable" />
  </WordVariables>
  <WordBookmarks>
    <WordBookmark Name="TestBookmark" />
  </WordBookmarks>
</ParameterDefinitions>
</RunVBAMacro>
</VBAMacros>
</AfterAction>
</OutputMethodInstruction>

```

---

## Parameter definitons

The «ParameterDefinition» tag within the OOMI XML gives you access to document values including officatwork specific variables. Most values can be read and written. To read a value you must define a request statement that will then be returned to you with the requested values. To write a value you must define a set statement.

### ActiveFileName()

This parameter returns the filename of the active document. The file can have the extensions \*.doc, \*.ows or \*.owt, depending on the host application.

---

This parameter is read-only.

---

#### Request Syntax

```
<ActiveFileName />
```

#### Return Syntax

```
<ActiveFileName Value=" " />
```

#### Parameter

The parameter «ActiveFileName» has the following attributes:

Name	Description
Value	String. Filename of active file

### Request Sample

```
<ParameterDefinitions>  
  <ActiveFileName/>  
</ParameterDefinitions>
```

### Return Sample

```
<ParameterDefinitions>  
  <ActiveFileName Value="Exampledocument.ows" />  
</ParameterDefinitions>
```

## ActivePath()

This parameter returns the path of the active document.

---

This parameter is read-only.

---

### Request Syntax

```
<ActivePath/>
```

### Return Syntax

```
<ActivePath Value="" />
```

### Parameter

The parameter «ActivePath» has the following attributes:

Name	Description
Value	String. Path of active file

### Request Sample

```
<ParameterDefinitions>  
  <ActivePath/>  
</ParameterDefinitions>
```

### Return Sample

```
<ParameterDefinitions>  
  <ActivePath Value="Q:\officeatwork\SmartTemplates" />  
</ParameterDefinitions>
```

## Cancel()

This parameter indicates whether officeatwork shall continue the output process or not.

If the value of the parameter «cancel» is returned with the value «-1» from the script, then officeatwork discontinues the output process. This mechanism can be used whenever the processed document should not be printed, sent or saved. This is useful whenever a document based on some status or values is not valid for creating any output.

---

This parameter is read-only.

In case of a cancellation of an output-process based on this cancel parameter, officeatwork stops the output process without notifying the user. Should you want to inform the user about the cancellation, you must alert the user within your scripts or Macros.

---

### Request Syntax

```
<Cancel />
```

### Return Syntax

```
<Cancel Value="-1" />
```

### Parameter

The parameter «Cancel» has the following attributes:

Name	Description
Value	Long. <b>0</b> , continue the output process <b>-1</b> , cancel the output process Default value is <b>0</b>

### Return Sample

```
<ParameterDefinitions>
  <Cancel Value="-1" />
</ParameterDefinitions>
```

## Changed()

This parameter indicates whether the document has been modified since it was last saved.

In combination with for instance a DMS solution, it may be necessary to know if the document has changed since it was last saved. In this case the changed parameter can be used to forbid any output from documents that have been changed.

---

This parameter is read-only.

---

### Request Syntax

```
<Changed/>
```

### Return Syntax

```
<Changed Value="0" />
```

### Parameter

The parameter «Changed» has the following attributes:

Name	Description
Value	Long. <b>0</b> , the document has not been modified since it was opened <b>-1</b> , the document has been modified since it was opened

### Request Sample

```
<ParameterDefinitions>  
  <Changed/>  
</ParameterDefinitions>
```

### Return Sample

```
<ParameterDefinitions>  
  <Changed Value="-1" />  
</ParameterDefinitions>
```

## DocumentLanguage ()

This parameter returns the document language in form of an LCID. A detailed list of available Language IDs (LCID) can be found in the appendix.

---

This parameter is read-only.

---

### Request Syntax

```
<DocumentLanguage/>
```

### Return Syntax

```
<DocumentLanguage Value=" " />
```

**Parameter**

The parameter «DocumentLanguage» has the following attributes:

<b>Name</b>	<b>Description</b>
Value	Long. The documents LCID

**Request Sample**

```
<ParameterDefinitions>
  <DocumentLanguage/>
</ParameterDefinitions>
```

**Return Sample**

```
<ParameterDefinitions>
  <DocumentLanguage Value="2055" />
</ParameterDefinitions>
```

## HostApplication()

This parameter returns the application name of the host application.

This parameter is useful to make sure your output profiles are only executed in the office applications.

---

This parameter is read-only.

---

**Request Syntax**

```
<HostApplication/>
```

**Return Syntax**

```
<HostApplication Value=" " />
```

**Parameter**

The parameter «HostApplication» has the following attributes:

<b>Name</b>	<b>Description</b>
Value	String. Name of the host application Possible values are «DocumentWizard» or «SmartTemplateManager»

### Request Sample

```
<ParameterDefinitions>
  <HostApplication/>
</ParameterDefinitions>
```

### Return Sample

```
<ParameterDefinitions>
  <HostApplication Value="DocumentWizard"/>
</ParameterDefinitions>
```

## MasterProperties ()

With this parameter all fields of any officeatwork MasterProperty can be read and written.

### Request Syntax

```
<MasterProperties>
  <MasterProperty IDName=" ">
    <Field Name=" "/>
  </MasterProperty>
</MasterProperties>
```

### Set Syntax

```
<MasterProperties>
  <MasterProperty IDName=" " Where=" " Is=" ">
    <Field Name=" " Value=" "/>
  </MasterProperty>
</MasterProperties>
```

### Return Syntax

```
<MasterProperties>
  <MasterProperty IDName=" ">
    <Field Name=" " Value=" "/>
  </MasterProperty>
</MasterProperties>
```

### Parameter

The element «MasterProperties» has the following element:

Name	Description
MasterProperty	Element. Element representing a specific MasterProperty. If you wish to work with two different MasterProperties, you need to add multiple MasterProperty elements to your MasterProperties element.

The element «MasterProperty» has the following attributes:

<b>Name</b>	<b>Description</b>
IDName	String. Value of the field IDName
Where	String. Name of field to be used to match value of Is attribute. Only available in the Set Syntax.
Is	String. Value used in combination with the Where attributes to find a specific record within a MasterProperty. Only available in the Set Syntax.

The element «MasterProperty» has the following sub-elements:

<b>Name</b>	<b>Description</b>
Field	Element. Element representing a specific field on a MasterProperty.

The element «Field» has the following attributes:

<b>Name</b>	<b>Description</b>
Name	String. Name of the Field.
Value	String. Corresponding field value.

### Request Sample

```
<ParameterDefinitions>
  <MasterProperties>
    <MasterProperty IDName="CustomFields">
      <Field Name="External" />
    </MasterProperty>
    <MasterProperty IDName="Signature1">
      <Field Name="Name" />
      <Field Name="Function" />
    </MasterProperty>
  </MasterProperties>
</ParameterDefinitions>
```

### Set Sample

```
<ParameterDefinitions>
  <MasterProperties>
    <MasterProperty IDName="Signature1" Where="EmployeeID" Is="AT874KK">
      <Field Name="Function" Value="CEO" />
    </MasterProperty>
  </MasterProperties>
</ParameterDefinitions>
```

### Return Sample

```
<ParameterDefinitions>
  <MasterProperties>
    <MasterProperty IDName="CustomFields">
      <Field Name="External" Value="Wert A" />
    </MasterProperty>
    <MasterProperty IDName="Signature1">
      <Field Name="Name" Value="Max von Arx" />
      <Field Name="Function" Value="Angestellter" />
    </MasterProperty>
  </MasterProperties>
</ParameterDefinitions>
```

## OutputFileName()

This parameter returns the filename that shall be used within a send or save output variant.

---

The filename extension must match with the requirements of the involved driver.

---

### Request Syntax

```
<OutputFileName />
```

### Return Syntax

```
<OutputFileName Value=" " />
```

### Parameter

The element «OutputFileName» has the following attributes:

Name	Description
Value	String. Name of the file that will be used for the output variant. In most cases this will be the name of the PDF File that will be created in this output variant.

**Request Sample**

```
<ParameterDefinitions>
  <OutputFileName/>
</ParameterDefinitions>
```

**Return Sample**

```
<ParameterDefinitions>
  <OutputFileName Value="MyPDFDocument.pdf" />
</ParameterDefinitions>
```

## OutputPath()

This parameter defines the path used for the output variant. It will be used in a send or save output variant for the file to be sent or saved.

**Request Syntax**

```
<OutputPath/>
```

**Return Syntax**

```
<OutputPath Value=" " />
```

**Parameter**

The element «OutputPath» has the following attributes:

<b>Name</b>	<b>Description</b>
Value	String. Path used to save the file. In most cases this will be the path of the PDF file.

**Request Sample**

```
<ParameterDefinitions>
  <OutputPath/>
</ParameterDefinitions>
```

**Return Sample**

```
<ParameterDefinitions>
  <OutputPath Value="C:\myFiles" />
</ParameterDefinitions>
```

## WordBookmarks ()

This parameter allows you to read and write values of word bookmarks.

---

If bookmarks are linked to officeatwork values, whatever you set the bookmark to will be replaced with what officeatwork fills the bookmarks with. So make sure the bookmarks you are using are not used by officeatwork.

---

### Request Syntax

```
<WordBookmarks>
  <WordBookmark Name=" " />
</WordBookmarks>
```

### Set and Return Syntax

```
<WordBookmarks>
  <WordBookmark Name=" " Value=" " />
</WordBookmarks>
```

### Parameter

The WordBookmarks element has the following sub-elements:

Name	Description
WordBookmark	Element. Each WordBookmark element represents a Word bookmark.

The element «WordBookmark» has the following attributes:

Name	Description
Name	String. Name of bookmark
Value	String. Bookmark value

### Request Sample

```
<ParameterDefinitions>
  <WordBookmarks>
    <WordBookmark Name="DMS_Datum" />
    <WordBookmark Name="DMS_Status" />
  </WordBookmarks>
</ParameterDefinitions>
```

### Set and Return Sample

```
<ParameterDefinitions>
  <WordBookmarks>
    <WordBookmark Name="DMS_Datum" Value="05.04.2007" />
    <WordBookmark Name="DMS_Status" Value="Draft" />
  </WordBookmarks>
</ParameterDefinitions>
```

## WordBuiltInDocumentProperties ()

This parameter allows you to read and write values of Word BuiltInDocumentProperties.

---

If Word BuiltInDocumentProperties are linked to officeatwork values, whatever you set the Word BuiltInDocumentProperties to will be replaced with what officeatwork fills the Word BuiltInDocumentProperties with. So make sure the Word BuiltInDocumentProperties you are using are not used by officeatwork.

---

### Request Syntax

```
<WordBuiltInDocumentProperties>
  <WordBuiltInDocumentProperty Name=" " />
</WordBuiltInDocumentProperties>
```

### Return Syntax

```
<WordBuiltInDocumentProperties>
  <WordBuiltInDocumentProperty Name=" " Value=" " />
</WordBuiltInDocumentProperties>
```

### Parameter

The element «WordBuiltInDocumentProperties» has the following sub-elements:

Name	Description
WordBuiltInDocumentProperty	Element. Each element represents a Word BuiltInDocumentProperty.

The element «WordBuiltInDocumentProperty» has the following attributes:

Name	Description
Name	String. Name of Word BuiltInDocumentProperty.
Value	String. Value of Word BuiltInDocumentProperty

### Request Sample

```
<ParameterDefinitions>
  <WordBuiltInDocumentProperties>
    <WordBuiltInDocumentProperty Name="Author" />
    <WordBuiltInDocumentProperty Name="Company" />
  </WordBuiltInDocumentProperties>
</ParameterDefinitions>
```

### Return Sample

```
<ParameterDefinitions>
  <WordBuiltInDocumentProperties>
    <WordBuiltInDocumentProperty Name="Author" Value="Name des Autors" />
    <WordBuiltInDocumentProperty Name="Company" Value="Firmenname" />
  </WordBuiltInDocumentProperties>
</ParameterDefinitions>
```

## WordCustomDocumentProperties ()

This parameter allows you to read and write values of Word CustomDocumentProperties.

---

If Word CustomDocumentProperties are linked to officeatwork values, whatever you set the Word CustomDocumentProperties to will be replaced with what officeatwork fills the Word CustomDocumentProperties with. So make sure the Word BuiltInDocumentProperties you are using are not used by officeatwork.

---

### Request Syntax

```
<WordCustomDocumentProperties>  
  <WordCustomDocumentProperty Name=" " />  
</WordCustomDocumentProperties>
```

### Return Syntax

```
<WordCustomDocumentProperties>  
  <WordCustomDocumentProperty Name=" " Value=" " />  
</WordCustomDocumentProperties>
```

### Parameter

The element «WordCustomDocumentProperties» has the following sub-elements:

Name	Description
WordCustomDocumentProperty	Element. Each element represents a Word CustomDocumentProperty

The element «WordCustomDocumentProperty» has the following attributes:

Name	Description
Name	String. Name of Word CustomDocumentProperty
Value	String. Value of Word CustomDocumentProperty

### Request Sample

```
<ParameterDefinitions>
  <WordCustomDocumentProperties>
    <WordCustomDocumentProperty Name="DMS_Datum" />
    <WordCustomDocumentProperty Name="DMS_Status" />
  </WordCustomDocumentProperties>
</ParameterDefinitions>
```

### Return Sample

```
<ParameterDefinitions>
  <WordCustomDocumentProperties>
    <WordCustomDocumentProperty Name="DMS_Datum" Value="5. April 2007" />
    <WordCustomDocumentProperty Name="DMS_Status" Value="Draft" />
  </WordCustomDocumentProperties>
</ParameterDefinitions>
```

## WordVariables ()

This parameter allows you to read and write values of Word Variables.

---

If Word variables are linked to officeatwork values, whatever you set the Word variables to will be replaced with what officeatwork fills the Word variables with. So make sure the Word variables you are using are not used by officeatwork.

---

### Request Syntax

```
<WordVariables>
  <WordVariable Name=" " />
</WordVariables>
```

### Return Syntax

```
<WordVariables>
  <WordVariable Name=" " Value=" " />
</WordVariables>
```

### Parameter

The element «WordVariables» has the following sub-elements:

Name	Description
WordVariable	Element. Each element represents a Word Variable.

The element «WordVariable» has the following attributes:

Name	Description
Name	String. Name of Word Variable.
Value	String. Value of Word Variable

### Request Sample

```
<ParameterDefinitions>
  <WordVariables>
    <WordVariable Name="DMS_Datum" />
    <WordVariable Name="DMS_Status" />
  </WordVariables>
</ParameterDefinitions>
```

### Return Sample

```
<ParameterDefinitions>
  <WordVariables>
    <WordVariable Name="DMS_Datum" Value="05.04.2007" />
    <WordVariable Name="DMS_Status" Value="Draft" />
  </WordVariables>
</ParameterDefinitions>
```

---

## Variables

officeatwork provides numerous variables that can be used within an OOMI-file. The variables can be placed within your OOMI-file and will be replaced with their appropriate values during the output process. The following list explains all the values available through variables.

[[Path]]	provides the current path of the document
[[Filename]]	provides the current filename of the document
[[HostApplicationFullName]]	provides the name of the current office application that triggered the output. For example «Word» or «PowerPoint»
[[Title]]	provides the value of the document Title as defined in the Master Template
[[Subject]]	provides the value of the document Subject as defined in the Master Template
[[Author]]	provides the value of the document Author as defined in the Master Template
[[Manager]]	provides the value of the document Manager as defined in the Master Template
[[Company]]	provides the value of the document Company as defined in the Master Template
[[Category]]	provides the value of the document Category as defined in the Master Template
[[Keywords]]	provides the value of the document Keywords as defined in the Master Template
[[Comments]]	provides the value of the document Comments as defined in the Master Template

[[Hyperlinkbase]] provides the value of the document Hyperlinkbase as defined in the Master Template

With the exception of the [[HostApplicationFullName]] variable, all variables are dependent on the document used in the output process. Many of the variable definitions are defined in the appropriate Master Template.



## CHAPTER 5

# Creating OOMI-Files

---

## Creation

The OOMI-files are located in the OutputMethodInstructions Folder within your Solution folder. The XML-text of the OOMI-file can be created/edited using a simple text editor, such as the application «notepad.exe». Of course, you can also use your favorite XML editor application. The easiest way to create a new OOMI-file is to duplicate an existing file and make your changes within the copied file.



## CHAPTER 6

# Integrating OOMI-Files

---

## Integration into an officeatwork solution

For an OOMI-file to be accessible by an output variant, it needs to be registered in the officeatwork solution. A detailed description on how to integrate OOMI-Files can be found in the «Solution-Manager Manual».

---

## Integration in output variations

For an output variation to utilize an OOMI-file, the OOMI-file needs to be registered in the output variation. It is possible to register multiple OOMI-files in the same output variation. In this case the sequence of the registration is responsible for the order in which officeatwork picks the OOMI-file when executing the output variation. A detailed description on how to register OOMI-Files within an output variation can be found in the «Solution-Manager Manual».



## CHAPTER 7

# Samples

---

## Sample 1: Adobe Acrobat (Save PDF)

The following sample uses the «Adobe PDF» printer driver.

```
<?xml version="1.0" encoding="UTF-8"?>
<OutputMethodInstruction xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Action>
    <Print ResultFilename="[[Path]][[Filename]].pdf" PrinterName="Adobe PDF"/>
  </Action>
</OutputMethodInstruction>
```

---

## Sample 2: Adobe Acrobat (Send PDF)

The following sample uses the «Adobe PDF» printer driver. In this case the path and filename are taken from the active document. The path and filename are additionally written into the registry. This will allow officeatwork to attach the pdf to an email later in the process

```
<?xml version="1.0" encoding="UTF-8"?>
<OutputMethodInstruction xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <BeforeAction RestoreSettingsAfterAction="0">
    <Registry>
      <Keys>
        <Key HKEY="HKEY_CURRENT_USER" Path="Software\Adobe\Acrobat Distiller\PrinterJobControl">
          <Values>
            <Value Name="[[HostApplicationFullName]]" Type="String" Value="[[Path]][[Filename]
].pdf"/>
          </Values>
        </Key>
      </Keys>
    </Registry>
  </BeforeAction>
  <Action>
    <Print ResultFilename="[[Path]][[Filename]].pdf" PrinterName="Adobe PDF"/>
  </Action>
</OutputMethodInstruction>
```

## Sample 3: PDF Creator

The following sample uses the «PDF Creator» printer driver. In this case the filename is taken from the active document. The final PDF file is saved to the desktop. Various registry entries are set in the «BeforeAction» section. Those variables will be used by the PDF driver when creating the PDF file. After completion of the «BeforeAction» section, all changes will be reverted to their original state as «RestoreSettingsAfterAction» is set to true.

```
<?xml version="1.0" encoding="UTF-8"?>
<OutputMethodInstruction xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <BeforeAction RestoreSettingsAfterAction="-1">
    <Registry>
      <Keys>
        <Key HKEY="HKEY_CURRENT_USER" Path="Software\PDFCreator\Program" >
          <Values>
            <Value Name="RunProgramAfterSaving" Value="0"></Value>
            <Value Name="RunProgramBeforeSaving" Value="0"></Value>
            <Value Name="UseAutosave" Value="1"></Value>
            <Value Name="UseAutosaveDirectory" Value="1"></Value>
            <Value Name="AutosaveDirectory" Value="%Desktop%"></Value>
            <Value Name="AutosaveFilename" Value="[[Filename]]"></Value>
            <Value Name="AutosaveFormat" Value="0"></Value>
          </Values>
        </Key>
        <Key HKEY="HKEY_CURRENT_USER" Path="Software\PDFCreator\Printing" >
          <Values>
            <Value Name="StandardAuthor" Value="[[Author]]"></Value>
            <Value Name="UseStandardAuthor" Value="1"></Value>
            <Value Name="StandardTitle" Value="[[Title]]"></Value>
            <Value Name="StandardSubject" Value="[[Subject]]"></Value>
            <Value Name="StandardKeywords" Value="[[Keywords]]"></Value>
          </Values>
        </Key>
      </Keys>
    </Registry>
  </BeforeAction>
  <Action>
    <Print ResultFilename="%Desktop%\[[Filename]].pdf" PrinterName="PDFCreator"/>
  </Action>
</OutputMethodInstruction>
```

## Sample 4: BullZip PDF

The following sample uses the «BullZip PDF» printer driver. In this case the filename is taken from the active document. The final PDF file is saved to the desktop. Various registry entries are set in the «BeforeAction» section. Those variables will be used by the PDF driver when creating the PDF file. After completion of the «BeforeAction» section, all changes will be reverted to their original state as «RestoreSettingsAfterAction» is set to true.

```
<?xml version="1.0" encoding="UTF-8"?>
<OutputMethodInstruction xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <BeforeAction RestoreSettingsAfterAction="-1">
    <IniFiles>
      <IniFile Path="%AppData%\Bullzip\PDF Printer" Filename="runonce.ini">
        <Sections>
          <Section Name="PDF Printer">
            <Entries>
              <Entry Name="Output" Value="%Desktop%\[ [Filename] ].pdf"/>
              <Entry Name="ConfirmOverwrite" Value="no"/>
              <Entry Name="ShowSettings" Value="never"/>
              <Entry Name="ShowPDF" Value="yes"/>
              <Entry Name="WatermarkText" Value="[ [Title] ], [ [Author] ]"/>
              <Entry Name="WatermarkSize" Value="5"/>
              <Entry Name="WatermarkRotation" Value="55"/>
            </Entries>
          </Section>
        </Sections>
      </IniFile>
    </IniFiles>
  </BeforeAction>
  <Action>
    <Print ResultFilename="%Desktop%\[[Filename]].pdf" PrinterName="BullZip PDF Printer"/>
  </Action>
</OutputMethodInstruction>
```

---

## Sample 5: PDF Factory Save

The following sample uses the «PDF Factory» printer driver. In this case the path and filename are taken from the active document. The registry entry «ShowDlg» is set in the «BeforeAction» section. Those variables will be used by the PDF driver when creating the PDF file. After completion of the «BeforeAction» section, all changes will be reverted to their original state as «RestoreSettingsAfterAction» is set to true.

```
<?xml version="1.0" encoding="UTF-8"?>
<OutputMethodInstruction xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <BeforeAction RestoreSettingsAfterAction="-1">
    <Registry>
      <Keys>
        <Key HKEY="HKEY_CURRENT_USER" Path="Software\FinePrint
Software\pdfFactory3\FinePrinters\,,Server,pdfFactory">
          <Values>
            <Value Name="ShowDlg" Value="1"></Value>
          </Values>
        </Key>
      </Keys>
    </Registry>
  </BeforeAction>
  <Action>
    <Print ResultFilename="[[Path]][[Filename]].pdf" PrinterName="\\Server\pdfFactory"/>
  </Action>
</OutputMethodInstruction>
```

---

Note: PDF Factory will replace the «,» in the registry entries with «\» before using the entry.

---

---

## Sample 6: PDF Factory Send

The following sample uses the «PDF Factory» printer driver. In this case the path and filename are taken from the active document. This will allow officeatwork to pick up the finished PDF document and attach it to an email. Various registry entries are set in the «BeforeAction» section. Those variables will be used by the PDF driver when creating the PDF file. After completion of the «BeforeAction» section, all changes will be reverted to their original state as «RestoreSettingsAfterAction» is set to true.

```
<?xml version="1.0" encoding="UTF-8"?>
<OutputMethodInstruction xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <BeforeAction RestoreSettingsAfterAction="-1">
    <Registry>
      <Keys>
        <Key HKEY="HKEY_CURRENT_USER" Path="Software\FinePrint
Software\pdfFactory3\FinePrinters\,,Server,pdfFactory">
          <Values>
            <Value Name="ShowDlg" Value="2"></Value>
            <Value Name="PdfAction" Value="0"></Value>
          </Values>
        </Key>
        <Key HKEY="HKEY_CURRENT_USER" Path="Software\FinePrint Software\pdfFactory3">
          <Values>
            <Value Name="OutputFile" Value="[[Path]]\[[Filename]].pdf"></Value>
          </Values>
        </Key>
      </Keys>
    </Registry>
  </BeforeAction>
  <Action>
    <Print ResultFilename="[[Path]]\[[Filename]].pdf" PrinterName="\\Server\pdfFactory"/>
  </Action>
</OutputMethodInstruction>
```

---

**Note:** PDF Factory will replace the «,» in the registry entries with «\» before using the entry.

---

---

## Sample 7: FreePDF XP Save

The following sample uses the «FreePDF XP» printer driver. In this case the filename is taken from the active document. The final PDF file is saved to the desktop. Various registry entries are set in the «BeforeAction» section. Those variables will be used by the PDF driver when creating the PDF file. After completion of the «BeforeAction» section, all changes will be reverted to their original state as «RestoreSettingsAfterAction» is set to true.

```
<?xml version="1.0" encoding="UTF-8"?>
<OutputMethodInstruction xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <BeforeAction RestoreSettingsAfterAction="-1">
    <Registry>
      <Keys>
        <Key HKEY="HKEY_CURRENT_USER" Path="Software\shbox\FreePdfXp" >
          <Values>
            <Value Name="ADefault" Value="desktop"></Value>
            <Value Name="Desktop" Value="%desktop%"></Value>
            <Value Name="OpenPDF" Value="0"></Value>
            <Value Name="psFor" Value="[[Author]]"></Value>
          </Values>
        </Key>
      </Keys>
    </Registry>
  </BeforeAction>
  <Action>
    <Print ResultFilename="%Desktop%\[[Filename]].pdf" PrinterName="FreePDF XP"/>
  </Action>
</OutputMethodInstruction>
```

---

## Sample 8: FreePDF XP Send

The following sample uses the «FreePDF XP» printer driver. In this case the filename is taken from the active document. Various registry entries are set in the «BeforeAction» section. Those variables will be used by the PDF driver when creating the PDF file. After completion of the «BeforeAction» section, all changes will be reverted to their original state as «RestoreSettingsAfterAction» is set to true.

```
<?xml version="1.0" encoding="UTF-8"?>
<OutputMethodInstruction xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <BeforeAction RestoreSettingsAfterAction="-1">
    <Registry>
      <Keys>
        <Key HKEY="HKEY_CURRENT_USER" Path="Software\shbox\FreePdfXp" >
          <Values>
            <Value Name="Adefault" Value="email"></Value>
            <Value Name="OpenPDF" Value="0"></Value>
            <Value Name="psFor" Value="[[Author]]"></Value>
          </Values>
        </Key>
      </Keys>
    </Registry>
  </BeforeAction>
  <Action>
    <Print ResultFilename="[[Path]]\[Filename].pdf" PrinterName="FreePDF XP"/>
  </Action>
</OutputMethodInstruction>
```

---

**Note:** The «FreePDF XP» driver will not accept a predefined filename. That is why we cannot use officeatwork to generate the email message and attach the pdf document as it does not know the name of the pdf document to attach. Moreover, we use the email function built into the «FreePDF XP» driver to create the email itself. This way the subject and body of the mail message cannot be set by officeatwork. This is, of course, subject to change depending on what the developers of «FreePDF XP» have in mind for future versions of their driver.

---



# Appendix

## LCID's

Language	ID	Language	ID
Afrikaans - South Africa	1078	Chinese - Hong Kong SAR	3076
Albanian - Albania	1052	Chinese - Macao SAR	5124
Amharic - Ethiopia	1118	Croatian	1050
Arabic - Saudi Arabia	1025	Croatian (Bosnia/Herzegovina)	4122
Arabic - Algeria	5121	Czech	1029
Arabic - Bahrain	15361	Danish	1030
Arabic - Egypt	3073	Divehi	1125
Arabic - Iraq	2049	Dutch - Netherlands	1043
Arabic - Jordan	11265	Dutch - Belgium	2067
Arabic - Kuwait	13313	Edo	1126
Arabic - Lebanon	12289	English - United States	1033
Arabic - Libya	4097	English - United Kingdom	2057
Arabic - Morocco	6145	English - Australia	3081
Arabic - Oman	8193	English - Belize	10249
Arabic - Qatar	16385	English - Canada	4105
Arabic - Syria	10241	English - Caribbean	9225
Arabic - Tunisia	7169	English - Hong Kong SAR	15369
Arabic - U.A.E.	14337	English - India	16393
Arabic - Yemen	9217	English - Indonesia	14345
Armenian - Armenia	1067	English - Ireland	6153
Assamese	1101	English - Jamaica	8201
Azeri (Cyrillic)	2092	English - Malaysia	17417
Azeri (Latin)	1068	English - New Zealand	5129
Basque	1069	English - Philippines	13321
Belarusian	1059	English - Singapore	18441
Bengali	1093	English - South Africa	7177
Bengali (Bangladesh)	2117	English - Trinidad	11273
Bosnian (Bosnia/Herzegovina)	5146	English - Zimbabwe	12297
Bulgarian	1026	Estonian	1061
Burmese	1109	Faroese	1080
Catalan	1027	Farsi	1065
Cherokee - United States	1116	Filipino	1124
Chinese - People's Republic of China	2052	Finnish	1035
Chinese - Singapore	4100	French - France	1036
Chinese - Taiwan	1028	French - Belgium	2060

Language	ID	Language	ID
French - Cameroon	11276	Inuktitut	1117
French - Canada	3084	Italian - Italy	1040
French - Democratic Rep. of Congo	9228	Italian - Switzerland	2064
French - Cote d'Ivoire	12300	Japanese	1041
French - Haiti	15372	Kannada	1099
French - Luxembourg	5132	Kanuri - Nigeria	1137
French - Mali	13324	Kashmiri	2144
French - Monaco	6156	Kashmiri (Arabic)	1120
French - Morocco	14348	Kazakh	1087
French - North Africa	58380	Khmer	1107
French - Reunion	8204	Konkani	1111
French - Senegal	10252	Korean	1042
French - Switzerland	4108	Kyrgyz (Cyrillic)	1088
French - West Indies	7180	Lao	1108
Frisian - Netherlands	1122	Latin	1142
Fulfulde - Nigeria	1127	Latvian	1062
FYRO Macedonian	1071	Lithuanian	1063
Gaelic (Ireland)	2108	Malay - Malaysia	1086
Gaelic (Scotland)	1084	Malay - Brunei Darussalam	2110
Galician	1110	Malayalam	1100
Georgian	1079	Maltese	1082
German - Germany	1031	Manipuri	1112
German - Austria	3079	Maori - New Zealand	1153
German - Liechtenstein	5127	Marathi	1102
German - Luxembourg	4103	Mongolian (Cyrillic)	1104
German - Switzerland	2055	Mongolian (Mongolian)	2128
Greek	1032	Nepali	1121
Guarani - Paraguay	1140	Nepali - India	2145
Gujarati	1095	Norwegian (Bokmål)	1044
Hausa - Nigeria	1128	Norwegian (Nynorsk)	2068
Hawaiian - United States	1141	Oriya	1096
Hebrew	1037	Oromo	1138
Hindi	1081	Papiamentu	1145
Hungarian	1038	Pashto	1123
Ibibio - Nigeria	1129	Polish	1045
Icelandic	1039	Portuguese - Brazil	1046
Igbo - Nigeria	1136	Portuguese - Portugal	2070
Indonesian	1057	Punjabi	1094

Language	ID	Language	ID
Punjabi (Pakistan)	2118	Spanish - Peru	10250
Quecha - Bolivia	1131	Spanish - Puerto Rico	20490
Quecha - Ecuador	2155	Spanish - United States	21514
Quecha - Peru	3179	Spanish - Uruguay	14346
Rhaeto-Romanic	1047	Spanish - Venezuela	8202
Romanian	1048	Sutu	1072
Romanian - Moldava	2072	Swahili	1089
Russian	1049	Swedish	1053
Russian - Moldava	2073	Swedish - Finland	2077
Sami (Lappish)	1083	Syriac	1114
Sanskrit	1103	Tajik	1064
Sepedi	1132	Tamazight (Arabic)	414
Serbian (Cyrillic)	3098	Tamazight (Latin)	1119
Serbian (Latin)	2074	Tamil	1097
Sindhi - India	1113	Tatar	1092
Sindhi - Pakistan	2137	Telugu	1098
Sinhalese - Sri Lanka	1115	Thai	1054
Slovak	1051	Tibetan - Bhutan	2129
Slovenian	1060	Tibetan - People's Republic of China	1105
Somali	1143	Tigrigna - Eritrea	2163
Sorbian	1070	Tigrigna - Ethiopia	1139
Spanish - Spain (Modern Sort)	3082	Tsonga	1073
Spanish - Spain (Traditional Sort)	1034	Tswana	1074
Spanish - Argentina	11274	Turkish	1055
Spanish - Bolivia	16394	Turkmen	1090
Spanish - Chile	13322	Uighur - China	1152
Spanish - Colombia	9226	Ukrainian	1058
Spanish - Costa Rica	5130	Urdu	1056
Spanish - Dominican Republic	7178	Urdu - India	2080
Spanish - Ecuador	12298	Uzbek (Cyrillic)	2115
Spanish - El Salvador	17418	Uzbek (Latin)	1091
Spanish - Guatemala	4106	Venda	1075
Spanish - Honduras	18442	Vietnamese	1066
Spanish - Latin America	58378	Welsh	1106
Spanish - Mexico	2058	Xhosa	1076
Spanish - Nicaragua	19466	Yi	1144
Spanish - Panama	6154	Yiddish	1085
Spanish - Paraguay	15370	Yoruba	1130

# OutputMethodInstruction schema

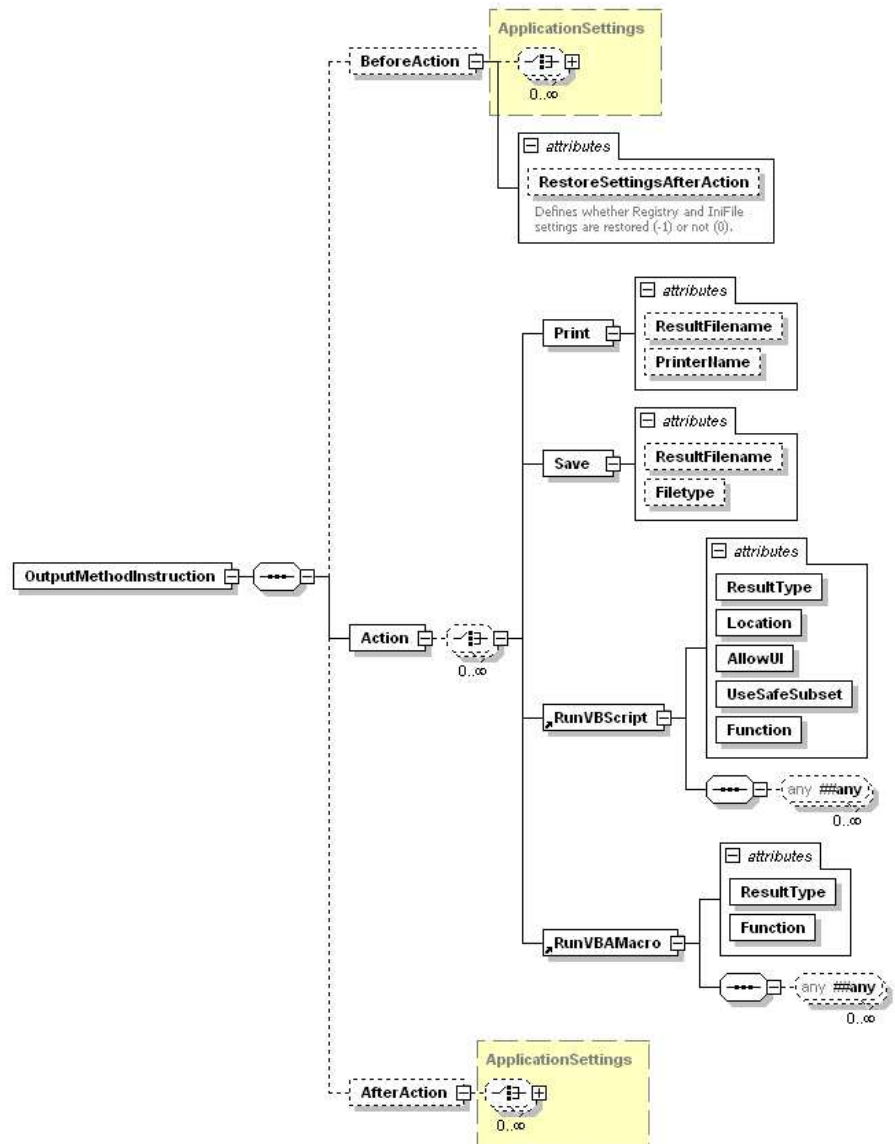


Figure 2: officatwork OOMI-XML Structure

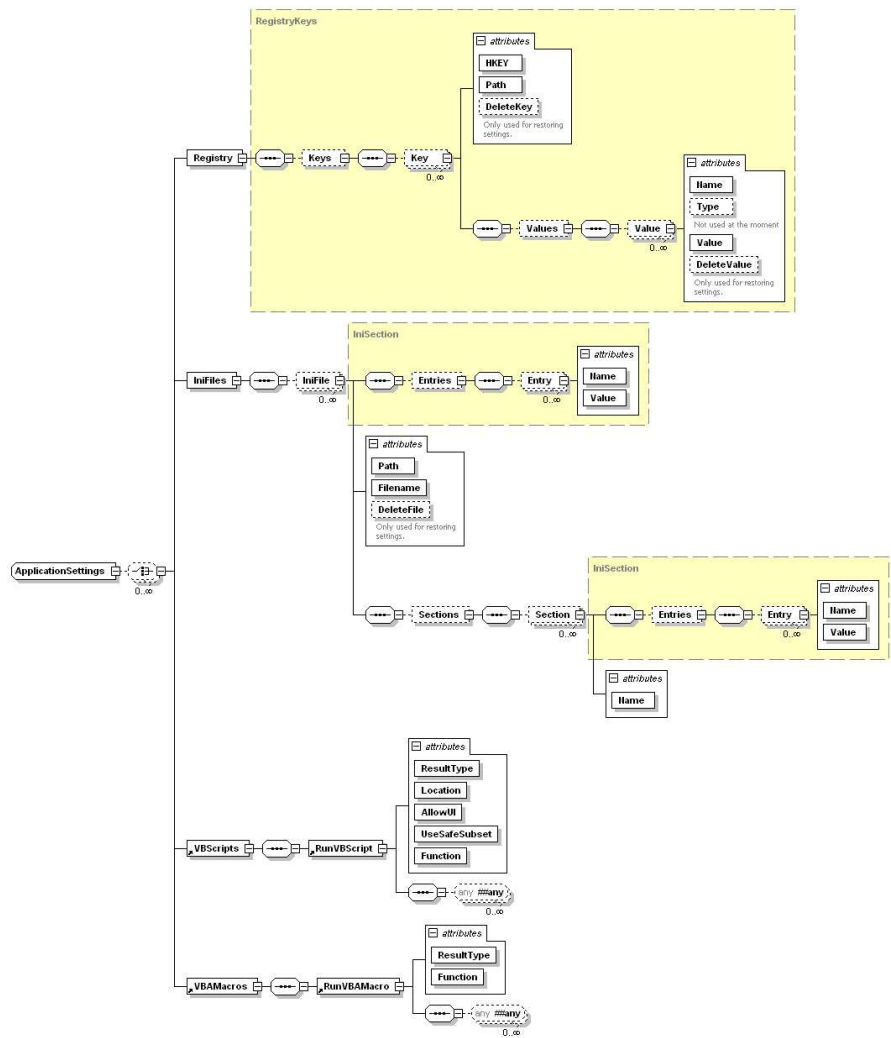


Figure 3: officeatwork OOMI-XML structure, ApplicationSettings

# Index

## —A—

ActiveFileName(), 19  
 ActivePath(), 20  
 Administrators, 5, 7  
 Adobe Acrobat, 39  
 API, 7

## —B—

BullZip PDF, 41

## —C—

Cancel(), 20  
 Changed(), 21  
 Concept, 11  
   Introduction, 11  
 Create  
   OOMI-files, 35

## —D—

Definitions, 15, 16  
 Developers, 7  
   API, 7  
   officeatwork API, 7  
   officeatwork Developer Manual, 7  
 DocumentLanguage(), 22  
 Download  
   manuals, 8

## —E—

Erstellung von OOMI-Dateien, 35

## —F—

FreePDF XP, 44, 45

## —H—

Help, 7  
 Help desk, 7  
 HostApplication(), 23

## —I—

Integrate  
   officeatwork solution, 37  
   OOMI-files, 37  
   output variations, 37

## —L—

LCID, 48

## —M—

Manuals, 7  
   officeatwork 20-Minute Guide, 7  
   officeatwork API, 7  
   officeatwork Installation Guide, 7  
   officeatwork Lösung, 7  
   officeatwork Master-Template Manager, 7  
   officeatwork Output Method Instructions  
     (OOMI) Guide, 7  
   officeatwork Project Management, 7  
   officeatwork Smart-Content Manager, 7  
   officeatwork Smart-Template Manager, 7  
   officeatwork Solution Manager, 7  
   officeatwork User Manual, 7  
 MasterProperties(), 24

## —O—

officeatwork  
   Parameter, 19  
   Variables, 32  
 OOMI-files, 35, 37  
 OutputFileName(), 26  
 OutputMethodInstruction, 51  
 OutputPath(), 27

## —P—

Parameter definition  
   ActiveFileName, 19  
 Parameter definitons, 19  
 PDF Creator, 40  
 PDF Factory, 42, 43  
 Project Managers, 7

## —Q—

Questions  
   users, 7

## —S—

Sample  
   BullZip PDF, 41  
   PDF Creator, 40  
 Samples, 39  
   Adobe Acrobat, 39  
   FreePDF XP, 44, 45  
   PDF Factory, 42, 43  
 Schema OutputMethodInstruction, 51  
 Service desk, 7  
 Steps, 12

Structure, 15  
Strucutre, 12  
Support, 7, 8

**—T—**

Typographic codes & conventions, 5

**—U—**

Users, 7

**—W—**

WordBookmarks(), 27  
WordBuiltInDocumentProperties(), 29  
WordCustomDocumentProperties(), 30  
WordVariables(), 31

**—X—**

XML, 5, 12, 15, 16



LIB Consultants AG  
Bundesplatz 12  
6300 Zug, Switzerland

T +41 (0)41 763 16 70  
F +41 (0)41 763 16 75

Hotline (Switzerland only)  
0900 566 088  
(CHF 3.50 per minute)

[www.officeatwork.com](http://www.officeatwork.com)  
[mail@officeatwork.com](mailto:mail@officeatwork.com)