

# Adding Files as Attachments to SAP Interactive Forms in the Java Environment



## Applies to:

SAP NetWeaver 7.0, For more information, visit the [SAP Interactive Forms by Adobe](#).

## Summary

This document demonstrates how to add attachments to your Adobe Form. Very often Adobe Forms are used to provide summary reports that can be downloaded by the user. When these reports have supporting documents linked to them, they can be added to Adobe Form as attachments.

**Author:** Joyer Cardoza

**Company:** L&T Infotech

**Created on:** 26 October 2009

## Author Bio



Joyer Cardoza is a SAP NetWeaver Consultant with L&T Infotech having over 4 years of experience in Java, Enterprise Portal and WebDynpro.

## Table of Contents

Introduction .....	3
Prerequisites .....	3
Business Scenario .....	3
Solution Implementation .....	4
WebDynpro Context .....	4
Sample Code .....	4
Related Content .....	6
Disclaimer and Liability Notice .....	7

## Introduction

This document aims to explain how to add files as attachments in an Adobe Form generated by WebDynpro code.

## Prerequisites

Knowledge Requirement

1. Java
2. WebDynpro Java
3. Adobe Interactive Forms and PDF Document Object

## Business Scenario

User seeks to make an SAP FI Posting using custom developed interactive and intuitive wizard. Due to stringent audit requirement he is required to upload Supporting Documentation in this wizard that supports his request for such a posting. The document can be of any type – an image of email approval from his superior, an excel doc containing transaction details, word doc etc.

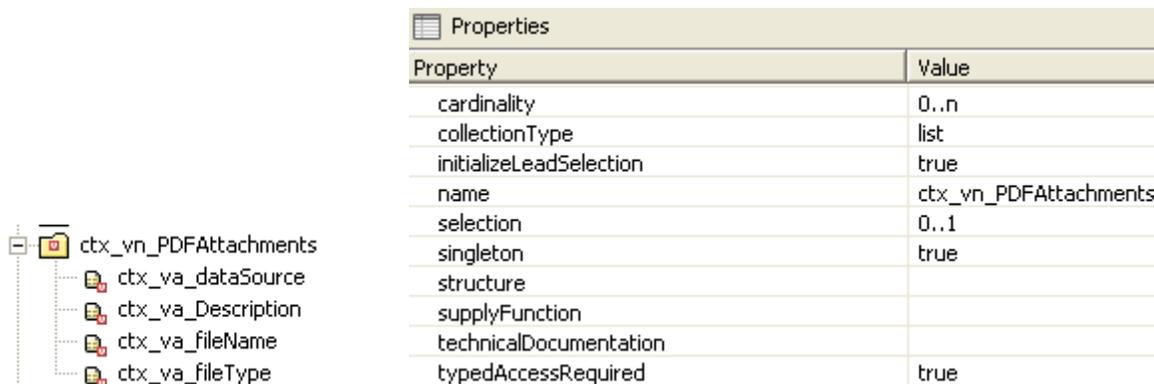
Towards the end of the process, the user wants to maintain a copy of the data he has fed in the wizard along with the supporting documents uploaded by him, for his personal records.

The reviewer uses the data and attachments in the PDF report for his audit.

## Solution Implementation

**Note:** The Adobe Form in this case has been created using an XDP Template and an XML Data Source and is a Non Interactive Form.

### WebDynpro Context



The screenshot shows a context tree on the left with the following structure:

- ctx\_vn\_PDFAttachments
  - ctx\_va\_dataSource
  - ctx\_va\_Description
  - ctx\_va\_fileName
  - ctx\_va\_fileType

To the right is a 'Properties' table:

Property	Value
cardinality	0..n
collectionType	list
initializeLeadSelection	true
name	ctx_vn_PDFAttachments
selection	0..1
singleton	true
structure	
supplyFunction	
technicalDocumentation	
typedAccessRequired	true

Attribute Name	type
ctx_va_dataSource	binary
ctx_va_Description	string
ctx_va_fileName	string
ctx_va_fileType	WDWebResourceType

### Sample Code

```

IWDPDFDocumentCreationContext pdfObject =
WDPDFDocumentFactory.getDocumentHandler().getDocumentCreationContext();

pdfObject.setData(xmlOutputStream); // XML Data Source
pdfObject.setTemplate(byteArrXdpOutputStream); //XDP Template
pdfObject.setInteractive(false);
    
```

**Note:** When you have an existing Interactive Form use:  
**IWDPDFDocumentAccessibleContext pdfObject =**  
**WDPDFDocumentFactory.getDocumentHandler().getDocumentAccessibleContext();**  
**pdfObject.setPDF(byte[] pdfStream) or pdfObject.setPDF(OutputStream pdfStream)**

```

//Node Containing files to be added as attachments to the PDF
IPublicPDFComp.ICtx_vn_PDFAttachmentsNode l_attachmentsNode =
wdContext.nodeCtx_vn_PDFAttachments();

IPublicPDFComp.ICtx_vn_PDFAttachmentsElement l_attachmentsElement = null;
    
```

```

IWDPDFDocumentAttachment l_pdfAttachment = null;

for(int i=0; i< l_attachmentsNode.size(); i++){
    l_attachmentsElement =
        l_attachmentsNode.getCtx_vn_PDFAttachmentsElementAt(i);
    l_pdfAttachment = pdfObject.newAttachment();
// Add new attachment, set the binary content, name and description

l_pdfAttachment.setDescription(l_attachmentsElement.getCtx_va_Description());
l_pdfAttachment.setMimeType(l_attachmentsElement.getCtx_va_fileType());
l_pdfAttachment.setName(l_attachmentsElement.getCtx_va_fileName());
l_pdfAttachment.setResource(l_attachmentsElement.getCtx_va_dataSource());
pdfObject.setAttachment(l_pdfAttachment);
}

ByteArrayOutputStream bytArrPDFOutputStream = new ByteArrayOutputStream();

try {

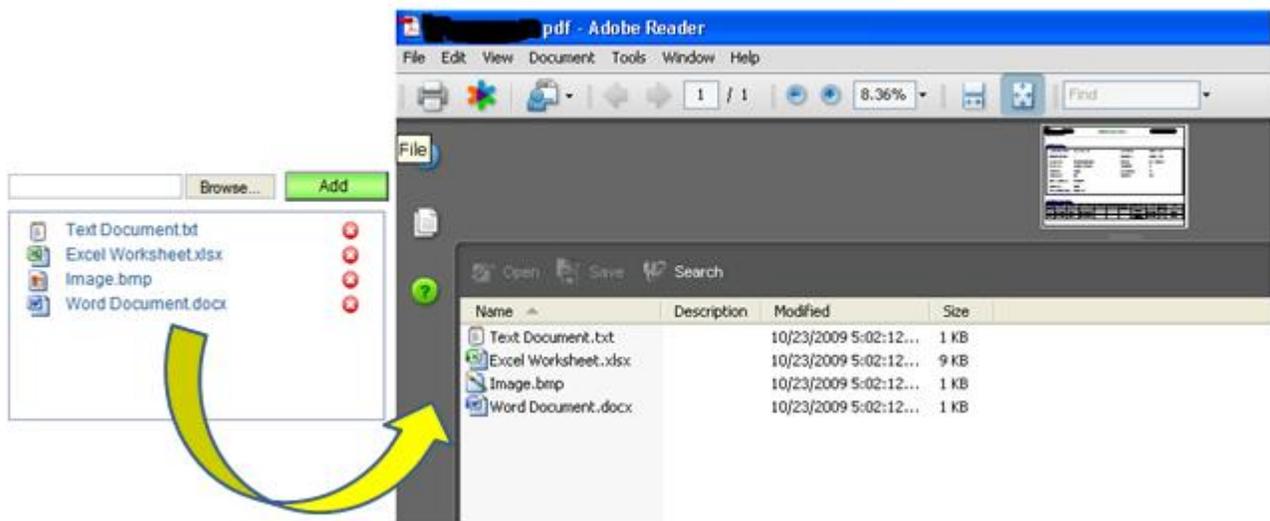
    ByteArrayInputStream inputPDFStream = new
        ByteArrayInputStream(pdfObject.execute().getPDF());

    IOUtil.write(inputPDFStream, bytArrPDFOutputStream);
    inputPDFStream.close();
    bytArrPDFOutputStream.close();

} catch (IOException e) {
    g_messageManager.reportException(e.toString(), true);
} catch (Exception e){
    g_messageManager.reportException(e.toString(), true);
}

// Create an object of IWDResource using the InputStream or byte[]. This can be
// bound with the resource property of the FileDownload
byte[] pdfWithAttachment = bytArrPDFOutputStream.toByteArray();

```



## **Related Content**

To know more, refer following links:

[SAP Interactive Forms by Adobe in the Java Environment](#)

[Javadocs](#)

[Interactive Forms eLearning](#)

For more information, visit the [SAP Interactive Forms by Adobe](#).

## Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.