

How to Download Multiple Files in ZIP using WebDynpro Java



Applies to:

SAP NetWeaver 7.0 SP15 and onwards, For more information, visit the [Web Dynpro Java homepage](#).

Summary

The document provides the core logic to download multiple files in ZIP format in WebDynpro Java. It demonstrates how one can use the java zip utilities in WebDynpro application to allow user to save multiple files. As an example PDF files are used, however, any file format (txt, doc, xls etc) can be downloaded.

Author: Abhinav Sharma

Company: L&T Infotech Ltd

Created on: 10 August 2009

Author Bio



Abhinav Sharma is SAP certified NetWeaver Consultant and working in L&T Infotech Ltd, Mumbai.

Table of Contents

Introduction	3
Prerequisites	3
Knowledge Requirement.....	3
System Requirement.....	3
Business Scenario	3
Implementation	3
Overview	3
Flowchart.....	3
Sample Code	4
Related Content.....	6
Disclaimer and Liability Notice.....	Error! Bookmark not defined.

Introduction

This document explains how to handle multiple PDF files in ZIP format and download it on a single click. The document is intended to give an idea of a solution that can be used in WebDynpro applications and provides only the core logic with the sample code.

Prerequisites

Knowledge Requirement

1. Core Java
2. Hands on experience in WebDynpro Java applications
3. Understanding of RFCs and its usage in WebDynpro

System Requirement

1. NWDS 04s SP15

Business Scenario

User searches the PO numbers based on PO status, delivery location, document date and last changed date. Custom RFC is build to get the POs based on user inputs. There is another RFC which converts the PO in PDF format. Internally it calls SAP standard RFC `CONVERT_OTFSPOOLJOB_2_PDF`. The output then comes in RAW format which is equivalent to byte array in Java.

Implementation

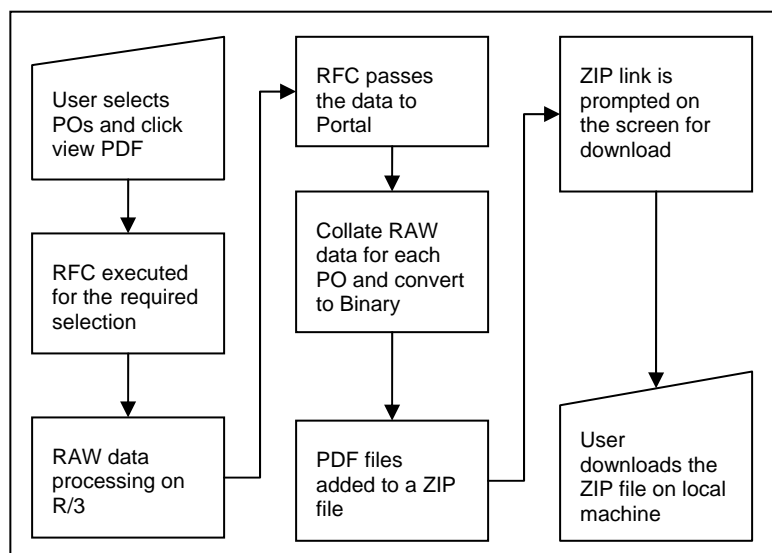
Overview

User is presented with a very simple layout. The result comes in the tabular format with the selection box in each row. User can select as many records as required. Once the selection is made user clicks on the Download button.

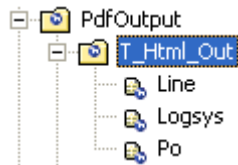
An RFC is executed which gets the PDF data in RAW format. In the core of the RFC is the `CONVERT_OTFSPOOLJOB_2_PDF` is used. Once RFC is executed and output is received, PDF files at runtime. It requires conversion of RAW data into binary format and then to PDF. The PDF files then added in ZIP and a single file is created. `java.util.zip` APIs are used to make ZIP file.

Flowchart

To understand the whole process more refer following diagram



For example let say Z_Po_Adobe_Out is the RFC that returns the output in RAW format. After executing the RFC, the output comes in T_Html_Out. Its cardinality is 0..n and the attribute Line is of type binary



Sample Code

```

IWDMessageManager msg_manager = wdComponentAPI.getMessageManager();
try
{
    //zip all PDFs data into zip file and allows user to save
    String p0No = "";
    String p0NoNext = "";
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    ZipOutputStream zos = new ZipOutputStream(baos);

    int count = wdContext.nodeT_Html_Out().size();
    int destPos = 0;
    byte[] out = new byte[1024 * count];
    byte[] Content = new byte[1024 * count];
    //Read bytes for each P0..
    for (int k = 0; k < count; k++){
        wdContext.nodeT_Html_Out().moveTo(k);

        //Copy to content byte array
        System.arraycopy(wdContext.currentT_Html_OutElement().getLine(), 0,
            Content,
            destPos, wdContext.currentT_Html_OutElement().getLine().length);

        destPos = destPos + wdContext.currentT_Html_OutElement().getLine().length;

        //Get the subsequent P0s
        if(k==0)
        {
            p0No = wdContext.nodeT_Html_Out().getT_Html_OutElementAt(0).getPo();
            p0NoNext =
wdContext.nodeT_Html_Out().getT_Html_OutElementAt(0).getPo();
        }
        else if(k==count-1)
        {
            p0No = wdContext.currentT_Html_OutElement().getPo();
            p0NoNext = p0No;
        }
        else
        {
            p0No = wdContext.currentT_Html_OutElement().getPo();
            p0NoNext =

            wdContext.nodeT_Html_Out().getT_Html_OutElementAt(k+1).getPo();
        }
    }
}

```

```

entry //Compare current and next P0 if P0 is different then add byte array to zip
{
    if(!p0No.equalsIgnoreCase(p0NoNext))
    {
        ZipEntry entry = new ZipEntry(p0No+".pdf");
        entry.setSize(Content.length);
        zos.putNextEntry(entry);
        zos.write(Content);
        zos.closeEntry();
        //bytes initialization
        count = wdContext.nodeT_Html_Out().size();
        destPos = 0;
        Content = new byte[1024 * count];
    }
    //Check for last P0 in the output and then add it in a zip file
    else if(k==count-1)
    {
        ZipEntry entry = new ZipEntry(p0No+".pdf");
        entry.setSize(Content.length);
        zos.putNextEntry(entry);
        zos.write(Content);
        zos.closeEntry();
    }
}
//Close the zipoutputstream
zos.close();

//Get the byte array for P0
out = baos.toByteArray();

//Set the Resource for zip file
IWDResource resource =
    WDRResourceFactory.createResource(
        out,
        viewpdf + ".zip",
        WDWebResourceType.UNKNOWN);

//Get the url form resource created
String url = resource.getUrl(WDFileDownloadBehaviour.AUTO.ordinal());
//Call external window passing the above url
IWDWindow window =
wdComponentAPI.getWindowManager().createNonModalExternalWindow(url,MultiPDF);
window.setWindowSize(0,0);
window.show();

} catch (WDDynamicRFCExecuteException e) {
    msg_manager.reportException(e.getMessage(), false);
}
catch (IOException ioe){
    msg_manager.reportException(ioe.getMessage(), false);
}
}

```

Related Content

To know more about ZIP functionality, refer following links.

- <http://java.sun.com/developer/technicalArticles/Programming/compression/>
- <http://java.sun.com/j2se/1.4.2/docs/api/java/util/zip/package-summary.html>
- http://help.sap.com/saphelp_nw04s/helpdata/EN/c3/76b45d9688e04abe1a1070410ddc1e/content.htm
- For more information, visit the [Web Dynpro Java homepage](#).

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.