

Worries, woes and wonders of ColdFusion 8 PDF handling

Chandan Kumar
Adobe

June 12, 2008

Agenda

- CFPDF: A tag to manipulate existing PDF
 - PDF Operations
 - DDX: What is DDX and how to use it to tightly control PDF assembling
- CFPDFForm : A look into
 - Different types of form
 - Various operations like Population/Extraction/Submission
- CFPRINT: Server side printing and functions to assist
- Q&A

CFPDF

- Provides functionality for manipulating PDF documents
- CFPDFParam is child Tag of CFPDF and its used only for merge
- Functions
 - IsDDX() – validates DDX against schema
 - IsPDFFile() – Validates if supplied file is a valid PDF file
 - IsPDFObject() – validate if supplied CF object is valid PDF Object
 - toBinary() – return byte array from PDFDocWrapper

CFPDF actions

- AddWatermark

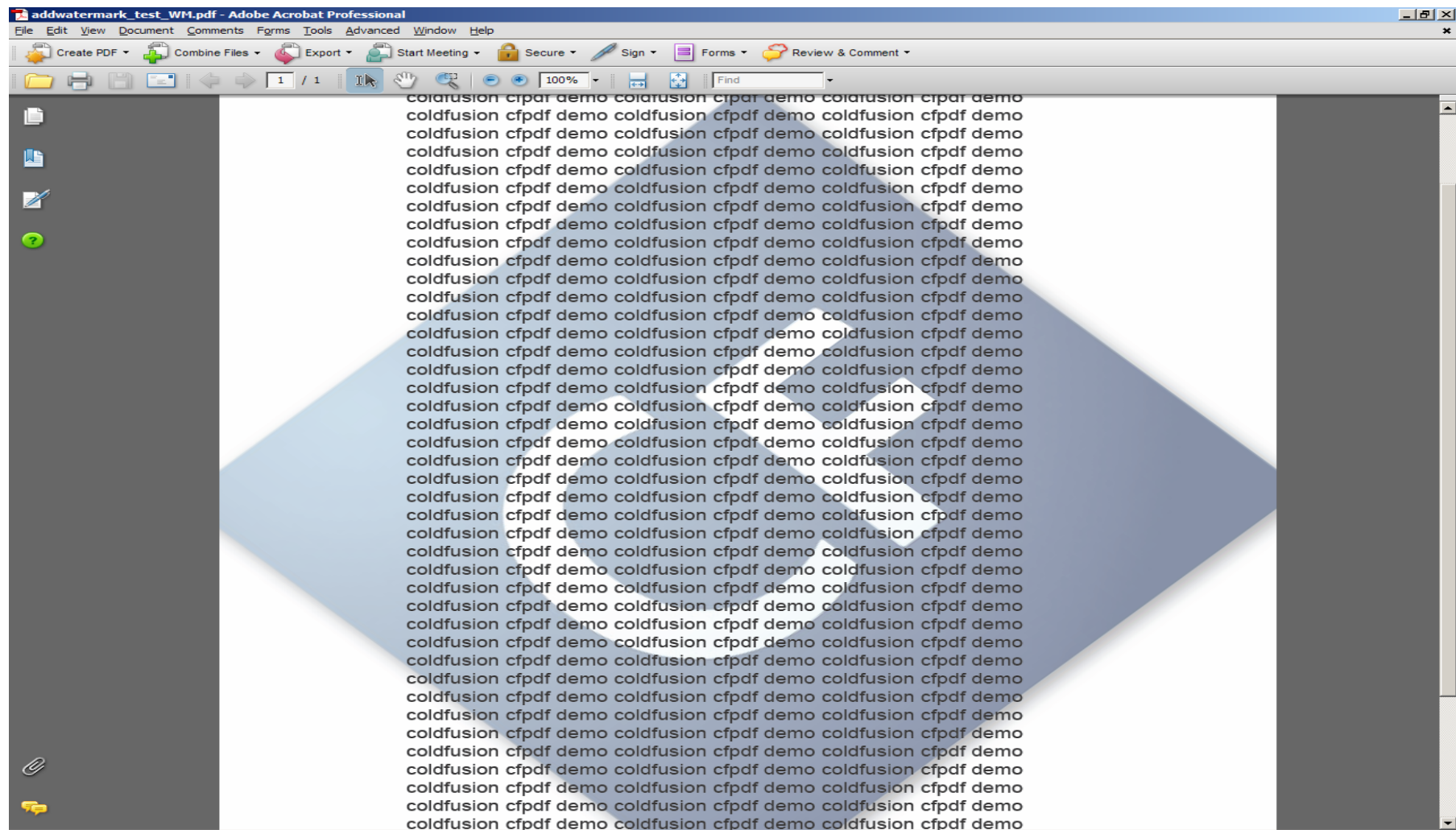
```
<cfpdf action="addwatermark" rotation="45" foreground="no"
  opacity="5" pages="1" showonprint="yes" image="cf8logo.jpg"
  source="addwatermark_test.pdf" destination="addwatermark_test_WM.pdf"
  overwrite="yes">
```

- Merge

Assembles PDF documents or pages from PDF source files into one output file.

```
<cfpdf action="merge" destination="destinationfile" overwrite="true">
  <cfpdfparam source="sourcefile1" pages="3-*">
  <cfpdfparam source="sourcefile2">
  <cfpdfparam source="sourcefile3" pages="2">
</cfpdf>
```

CFPDF addWaterMark



Other CFPDF actions

- **Read**
 - Reads a PDF document into a ColdFusion variable.
- **Protect**
 - Password-protects and encrypts a PDF document.
- **Thumbnail**
 - Generates thumbnail images from specified pages in a PDF document
- **RemoveWatermark**
 - Removes watermarks from specified pages in a PDF document.

Other CFPDF actions

- **deletePages**
 - Deletes one or more pages from a PDF document.
- **getInfo**
 - Extracts information associated with the PDF document, such as the author, title, and creation date.
- **setInfo**
 - Sets the Title, Subject, Author, and Keywords for a PDF document,
- **write**
 - Writes PDF output to a file. Also use to flatten forms created in Acrobat and linearize documents by specifying saveoption

CFPDF Code Snippet

```
<cfpdf action="read" source="test.pdf" name="testVar">
<cfpdf action="merge" source="testVar" name="newvar" pages="1">
<cfpdf action="addwatermark" text="#html#" source="newvar">
<cfpdf action="protect" source="newvar" newUserpassword="user"
permissions="none" newOwnerpassword="owner">
<cfpdf action="write" source="newvar" destination="final.pdf"
overwrite="true">
<cfmail to="#to#" from="#from#" subject="Report" type="html">
  <cfmailparam file="#CFFILE.ServerFile#"remove="true"/>
  <cfmailparam file="final.pdf" content="#newvar#" />
</cfmail>
<cfcontent type="application/pdf" variable="#tobinary(newvar)#">
```

CFPDF DDX processing

Restricted document assembly features using Adobe® LiveCycle™ Assembler DDX
(Refer to DDX Reference Documentation for DDX XML language)

Supported using **processddx** action

What can be done using ddx in CF

- Adding a table of contents
- Add Headers and Footers
- Formatting headers and footers
- Using style profiles
- Grouping PDF documents
- Setting the initial view of a PDF document
- Adding text-string watermarks
- Extracting text from a PDF document

CFPDF DDX processing ...

- **Sample DDX**

```
<DDX xmlns="http://ns.adobe.com/DDX/1.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ns.adobe.com/DDX/1.0/coldfusion_ddx.xsd">
  <PDF result="Out1">
    <TableOfContents maxBookmarkLevel="infinite" bookmarkTitle="Table
      of Contents" includeInTOC="false">
      <TableOfContentsEntryPattern>
        <StyledText font="italic"><p>
          <_BookmarkTitle />
          <leader leader-pattern="dotted" />
          <_BookmarkPageCitation /></p>
        </StyledText>
      </TableOfContentsEntryPattern>
    </TableOfContents>
    <PDF source="Doc1" />
    <PDF source="Doc2" />
  </PDF>
```

CFPDF DDX processingcontinued

```
<PDF result="out2">
  <Footer>
    <Right>
      <StyledText>
        <p>Page <_PageNumber/> of <_LastPageNumber/></p>
      </StyledText>
    </Right>
  </Footer>
  <Watermark rotation="30" opacity="65%">
    <StyledText><p font-size="50pt" font-weight="bold" color="lightgray"
      font="Arial">Draft</p></StyledText>
  </Watermark>
  <PDF source="Doc2" />
  <PDF source="Doc1" />
</PDF>
</DDX>
```

- **Sample Code for DDX processing**

```
<cfset input={Doc1="C:/Source1.pdf", Doc2="C:/Source2.pdf"}>
<cfset output={Out1="C:/Output1.pdf", Out2="C:/Output2.pdf"}>
<cfpdf action="processddx" ddxfile="sample.ddx" inputfiles="#input#"
  outputfiles="#output#" name="statusstruct">
```

Some FAQ

- Some DDX keywords are restricted
 - http://livedocs.adobe.com/coldfusion/8/htmldocs/help.html?content=Tags_p-q_02.html
- Generating TOC fails with “cause given”
 - `getAssemblerFontSet()` : lists all the fonts
- Prefilled Data gets lost while merging two forms
 - Only one form can be merged with other pdfs, try flattening
- CFPDF does not support dynamic XFA form flattening
- Want crisper thumbnails
 - `-Dorg.jpedal. Upscale` (requires more memory)

CFPDFFORM

About PDF Forms

- Two variety of PDF Forms
 - XFA Forms (Lifecycle/XML/Designer Forms)
 - Static XFA Forms
 - Dynamic XFA Forms
 - Acro Forms - Supports two Data Formats
 - FDF (Binary)
 - XFDF(XML based)
- Acro Forms are designed in Adobe Acrobat Professional
- XFA(LiveCycle) Forms are designed in Adobe LiveCycle Designer
- Unlike HTML, A PDF cannot have more than one PDF Form

LiveCycle Forms

- Data Formats

- Data can be filled only in XML format

```
<form1>  
    <subform1>  
        <name>Test Name</name>  
        <empcode>12345</empcode>  
    </subform1>  
</form1>
```

- Dynamic Forms appearance can change to adjust to the supplied data
 - Completely XML based (XDP)
 - Data DOM and Template DOM
 - Submission
 - HTTP Post
 - PDF Document
 - Others
 - Can be created using Adobe LiveCycle Designer

Acroforms

- Data Formats

- Only XFDF format Supported

```
<xfdf xmlns="http://ns.adobe.com/xfdf/" xml:space="preserve">
  <fields>
    <field name="firstname">
      <value>chandan</value>
    </field>
  </fields>
</xfdf>
```

- Appearance is static
- Cannot be used with LiveCycle suite of products
- Submission types
 - HTML
 - PDF Document
 - Others
- Can be created using Acrobat Professional

CFPDF Form

- What <CFPDFForm> can do?
 - Populate a static/dynamic XFA/Acro form
 - Extract values from a filled up static/dynamic XFA/Acro form
 - Can embed XFA/Acro form inside <cfdocument>
 - Can post data in HTTP format or entire PDF can be submitted to server
- What <CFPDFForm> cannot do?
 - It cannot create a PDF Form
 - It does not support FDF format for ACRO Forms

CFPDFForm Populate

- Child Tags
 - CFPDFSubform
 - CFPDFFormParam
- Hierarchy of form elements in designer should match with supplied data
- Does not throw any exception if hierarchy does not match but PDF Form wont show up the filled data
- Data can be supplied using child tags or thru a data xml file/ xml object/ xml string

Static LC Form in Designer

The screenshot displays the Adobe LiveCycle Designer interface for editing an expense claim form. The main design area shows a form titled "Expense Claim Form" with the following sections:

- Expense Claim Form**: Includes a "Submit" button and three input fields for "Employee Name", "Employee Code", and "Location".
- Approver**: Includes two input fields for "Managers Name" and "Managers Email".
- Expense Details**: A table with columns for "Item Type", "Item Detail", "Date", "Amount", and "Currency". The table contains five rows of data with dropdown menus for "Item Type" and "Currency".

The Hierarchy panel on the left lists the form's structure, including "Form1" (the root subform) and "subform1" (a child subform). Annotations with callouts point to specific elements:

- A callout points to "Form1" in the Hierarchy panel, stating: **form 1 is root subform**.
- A callout points to "subform1" in the Hierarchy panel, stating: **Subform1 (child subform of form1)**.
- A callout points to the "name" field in the Hierarchy panel, stating: **"name" form field**.

Item Type	Item Detail	Date	Amount	Currency
Travel				USD
Emp Morale				USD
Training				USD
Books				USD
Software				USD

Example - Populate LiveCycle Form

Population using Tags

```
<cfpdfform source="expenseclaim.pdf" action="populate">
  <cfpdfsubform name="form1">
    <cfpdfsubform name="subform1">
      <cfpdfformparam name="name" value="Test Name">
      <cfpdfformparam name="empcode" value="12345">
      <cfpdfformparam name="location" value="Bangalore">
      <cfpdfformparam name="mgrname" value="Test Mgr">
      <cfpdfformparam name="mgremail" value="mgr@adobe.com">
    </cfpdfsubform>
  </cfpdfsubform>
</cfpdfform>
```

Example – Populate LiveCycle Form XML DataFile

Population using xml data file

```
<cfpdfform source="expense.pdf" action="populate"  
  xmldata="expense_data.xml">`
```

expense_data.xml

```
<form1>  
  <subform1>  
    <name>chandan</name>  
    <empcode>10750</empcode>  
    <location>Bangalore</location>  
    <mgrname>Hemant</mgrname>  
    <mgremail>hkhandel@adobe.com</mgremail>  
  </subform1>  
</form1>
```

Static LC Form in Designer

Adobe LiveCycle Designer - [expenseclaim.pdf]

File Edit View Insert Table Layout Tools Window Help

71%

Hierarchy PDF Structure Data View

Design View Master Pages Preview PDF

Form1
(Master Pages)
Page1
(untitled Content Area)
subform1
name
StaticText1
Line1
StaticText2
empcode
Line2
StaticText3
location
Line3
StaticText4
date[0]
Rectangle1
itemtype[0]
Line4
Line5
Line6
StaticText5
StaticText6
StaticText7
StaticText8
Line7
StaticText9
mgrname
StaticText10
Line8
mgremail
Line9
StaticText11
Line10
StaticText12
amount[0]
amount[1]
amount[2]

Expense Claim Form

Submit

Employee Name Employee Code Location

Approver

Managers Name Managers Email

Expense Details

Item Type	Item Detail	Date	Amount	Currency
Travel				USD
Emp Morale				USD
Training				USD
Books				USD
Software				USD

2nd amount field can be accessed using index

```
<cfpdfformparam name="amount" value="#val#" index="2">
```

Dynamic LC Form in Designer

The screenshot displays the Adobe LiveCycle Designer interface for an "expenseclaim-dynamic.pdf" document. The main workspace shows a design view of an "Expense Claim Form" with various input fields and buttons. A red circle highlights the "expense" subform in the Hierarchy panel on the left. A red arrow points from this circle to the "expense" subform in the design view, which is also circled in red. A red arrow points from the "expense" subform in the design view to the "Repeat Subform for Each Data Item" checkbox in the Object Properties panel on the right, which is also circled in red. A red arrow points from the "Repeat Subform for Each Data Item" checkbox to the text "Dictates how many times subform should be repeated". Another red arrow points from the "expense" subform in the design view to the text "expense subform, which will be repeated for each record supplied".

Adobe LiveCycle Designer - [expenseclaim-dynamic.pdf]

File Edit View Insert Table Layout Tools Window Help

58%

Hierarchy Structure View

form1

- subform1
- expense
- itemtype
- itemdetail
- amount
- date
- currency

(Master Pages)

(untitled Subform) (page 3)

(Referenced Objects)

Design View Master Pages Preview PDF

Expense Claim Form

Submit as PDF Submit

Employee Name Employee Code Location

Approver

Managers Name Managers Email

Expense Details

Item Type	Item Detail	Date	Amount	Currency
Travel				USD

Object Properties Panel:

Object Layout Border

Subform Pagination Binding

Name: expense 0

Default Binding (Open, Save, Submit): Normal

Import/Export Bindings (Execute):

Repeat Subform for Each Data Item

Min Count: Max:

Initial Count:

Group Trailer:

expense subform, which will be repeated for each record supplied

Dictates how many times subform should be repeated

CFPDFForm - Dynamic Form population

```
<cfpdfsubform name="form1">
  <cfpdfsubform name="subform1">
    <cfloop index=i from=1 to=5>
      <cfpdfsubform name="expense" index="#i#">
        <cfpdfformparam name="itemtype" value="#type[i]#" >
          <cfpdfformparam name="itemdetail" value="#desc[i]#">
            <cfpdfformparam name="date"
              value="#dateformat(now(),"yyyy-mm-dd")#">
              <cfpdfformparam name="amount" value="#1000*i#">
              <cfpdfformparam name="DropDownList1" value="USD">
            </cfpdfsubform>
          </cfloop>
        </cfpdfsubform>
      </cfpdfsubform>
```

Dynamic LC Form – Ouput PDF

Expense Claim Form

sanjeev 10750 Bangalore

Employee Name Employee Code Location

Approver

Hemant hkhandel@adobe.com

Managers Name Managers Email

Expense Details

Item Type	Item Detail	Date	Amount	Currency
Travel	Chicago	Aug 28, 2007	1,000	USD
Training	ColdFusion	Aug 28, 2007	2,000	USD
Emp Morale	Party	Aug 28, 2007	3,000	USD
Software	ColdFusion 8	Aug 28, 2007	4,000	USD
Books	ColdFusion Books	Aug 28, 2007	5,000	USD

LC Form - Types of Submission

The screenshot displays the Adobe LiveCycle Designer interface for editing an "Expense Claim Form". The form layout includes several input fields: "Employee Name", "Employee Code", "Location", "Approver", "Managers Name", "Managers Email", "Expense Details", "Item Type", and "Item Detail". Two submission buttons are present: "Submit as PDF" and "Submit". The "Submit" button is highlighted with a red oval and a red callout line pointing to its properties in the right-hand panel.

The right-hand panel shows the properties for the selected "Submit" button:

- Object:** Submit
- Field:** Submit
- Submit to URL:** pdfreceiver.cfm
- Submit As:** PDF
- Sign Submission:**
- Include:** XML Data Package (XDP), PDF, XML Data (XML), URL-encoded Data (HTTP Post), Document Signatures, Template
- Other:** [Empty field]
- Data Encoding:** UTF-16

The bottom panel shows the properties for the selected "Submit as PDF" button:

- Object:** Submit
- Field:** Submit
- Type:** Button
- Caption:** Submit as PDF
- Appearance:** Raised Border
- Highlighting:** Push
- Rollover Caption:** [Empty field]
- Down Caption:** [Empty field]
- URL:** processForm.cfm
- Sign Submission:**
- Control Type:** Regular, Execute, Submit (selected)

CFPDFForm Submission/Extract

- Form Submission

- HTTP Post

- ProcessForm.cfm code

- ```
<cfdump var="#FORM.form1#">
```

- Entire PDF Document Submission

- PdfReceiver.cfm code

- ```
<cfpdfform source="#PDF.content#" action="read" result="struct"/>
```

- It can extract the form data as

- struct

- ```
<cfpdfform source="expenseclaim-filled.pdf" action="read" result="struct"/>
```

- xml data string

- ```
<cfpdfform source="expenseclaim-filled.pdf" action="read" xmldata="XMLString"/>
```

CFPDFForm result struct after Read action

struct																																																	
form1	<table border="1"><thead><tr><th colspan="2">struct</th></tr></thead><tbody><tr><td>expense</td><td><table border="1"><thead><tr><th colspan="2">array</th></tr></thead><tbody><tr><td>1</td><td><table border="1"><thead><tr><th colspan="2">struct</th></tr></thead><tbody><tr><td>DropDownList1</td><td>USD</td></tr><tr><td>amount</td><td>1000</td></tr><tr><td>date</td><td>2007-09-29</td></tr><tr><td>itemdetail</td><td>Chicago</td></tr><tr><td>itemtype</td><td>Travel</td></tr></tbody></table></td></tr><tr><td>2</td><td><table border="1"><thead><tr><th colspan="2">struct</th></tr></thead><tbody><tr><td>DropDownList1</td><td>USD</td></tr><tr><td>amount</td><td>2000</td></tr><tr><td>date</td><td>2007-09-29</td></tr><tr><td>itemdetail</td><td>ColdFusion</td></tr><tr><td>itemtype</td><td>Training</td></tr></tbody></table></td></tr></tbody></table></td></tr><tr><td>subform1</td><td><table border="1"><thead><tr><th colspan="2">struct</th></tr></thead><tbody><tr><td>empcode</td><td>10750</td></tr><tr><td>location</td><td>Bangalore</td></tr><tr><td>mgremail</td><td>hkhandel@adobe.com</td></tr><tr><td>mgrname</td><td>Hemant</td></tr><tr><td>name</td><td>sanjeev</td></tr></tbody></table></td></tr></tbody></table>	struct		expense	<table border="1"><thead><tr><th colspan="2">array</th></tr></thead><tbody><tr><td>1</td><td><table border="1"><thead><tr><th colspan="2">struct</th></tr></thead><tbody><tr><td>DropDownList1</td><td>USD</td></tr><tr><td>amount</td><td>1000</td></tr><tr><td>date</td><td>2007-09-29</td></tr><tr><td>itemdetail</td><td>Chicago</td></tr><tr><td>itemtype</td><td>Travel</td></tr></tbody></table></td></tr><tr><td>2</td><td><table border="1"><thead><tr><th colspan="2">struct</th></tr></thead><tbody><tr><td>DropDownList1</td><td>USD</td></tr><tr><td>amount</td><td>2000</td></tr><tr><td>date</td><td>2007-09-29</td></tr><tr><td>itemdetail</td><td>ColdFusion</td></tr><tr><td>itemtype</td><td>Training</td></tr></tbody></table></td></tr></tbody></table>	array		1	<table border="1"><thead><tr><th colspan="2">struct</th></tr></thead><tbody><tr><td>DropDownList1</td><td>USD</td></tr><tr><td>amount</td><td>1000</td></tr><tr><td>date</td><td>2007-09-29</td></tr><tr><td>itemdetail</td><td>Chicago</td></tr><tr><td>itemtype</td><td>Travel</td></tr></tbody></table>	struct		DropDownList1	USD	amount	1000	date	2007-09-29	itemdetail	Chicago	itemtype	Travel	2	<table border="1"><thead><tr><th colspan="2">struct</th></tr></thead><tbody><tr><td>DropDownList1</td><td>USD</td></tr><tr><td>amount</td><td>2000</td></tr><tr><td>date</td><td>2007-09-29</td></tr><tr><td>itemdetail</td><td>ColdFusion</td></tr><tr><td>itemtype</td><td>Training</td></tr></tbody></table>	struct		DropDownList1	USD	amount	2000	date	2007-09-29	itemdetail	ColdFusion	itemtype	Training	subform1	<table border="1"><thead><tr><th colspan="2">struct</th></tr></thead><tbody><tr><td>empcode</td><td>10750</td></tr><tr><td>location</td><td>Bangalore</td></tr><tr><td>mgremail</td><td>hkhandel@adobe.com</td></tr><tr><td>mgrname</td><td>Hemant</td></tr><tr><td>name</td><td>sanjeev</td></tr></tbody></table>	struct		empcode	10750	location	Bangalore	mgremail	hkhandel@adobe.com	mgrname	Hemant	name	sanjeev
struct																																																	
expense	<table border="1"><thead><tr><th colspan="2">array</th></tr></thead><tbody><tr><td>1</td><td><table border="1"><thead><tr><th colspan="2">struct</th></tr></thead><tbody><tr><td>DropDownList1</td><td>USD</td></tr><tr><td>amount</td><td>1000</td></tr><tr><td>date</td><td>2007-09-29</td></tr><tr><td>itemdetail</td><td>Chicago</td></tr><tr><td>itemtype</td><td>Travel</td></tr></tbody></table></td></tr><tr><td>2</td><td><table border="1"><thead><tr><th colspan="2">struct</th></tr></thead><tbody><tr><td>DropDownList1</td><td>USD</td></tr><tr><td>amount</td><td>2000</td></tr><tr><td>date</td><td>2007-09-29</td></tr><tr><td>itemdetail</td><td>ColdFusion</td></tr><tr><td>itemtype</td><td>Training</td></tr></tbody></table></td></tr></tbody></table>	array		1	<table border="1"><thead><tr><th colspan="2">struct</th></tr></thead><tbody><tr><td>DropDownList1</td><td>USD</td></tr><tr><td>amount</td><td>1000</td></tr><tr><td>date</td><td>2007-09-29</td></tr><tr><td>itemdetail</td><td>Chicago</td></tr><tr><td>itemtype</td><td>Travel</td></tr></tbody></table>	struct		DropDownList1	USD	amount	1000	date	2007-09-29	itemdetail	Chicago	itemtype	Travel	2	<table border="1"><thead><tr><th colspan="2">struct</th></tr></thead><tbody><tr><td>DropDownList1</td><td>USD</td></tr><tr><td>amount</td><td>2000</td></tr><tr><td>date</td><td>2007-09-29</td></tr><tr><td>itemdetail</td><td>ColdFusion</td></tr><tr><td>itemtype</td><td>Training</td></tr></tbody></table>	struct		DropDownList1	USD	amount	2000	date	2007-09-29	itemdetail	ColdFusion	itemtype	Training																		
array																																																	
1	<table border="1"><thead><tr><th colspan="2">struct</th></tr></thead><tbody><tr><td>DropDownList1</td><td>USD</td></tr><tr><td>amount</td><td>1000</td></tr><tr><td>date</td><td>2007-09-29</td></tr><tr><td>itemdetail</td><td>Chicago</td></tr><tr><td>itemtype</td><td>Travel</td></tr></tbody></table>	struct		DropDownList1	USD	amount	1000	date	2007-09-29	itemdetail	Chicago	itemtype	Travel																																				
struct																																																	
DropDownList1	USD																																																
amount	1000																																																
date	2007-09-29																																																
itemdetail	Chicago																																																
itemtype	Travel																																																
2	<table border="1"><thead><tr><th colspan="2">struct</th></tr></thead><tbody><tr><td>DropDownList1</td><td>USD</td></tr><tr><td>amount</td><td>2000</td></tr><tr><td>date</td><td>2007-09-29</td></tr><tr><td>itemdetail</td><td>ColdFusion</td></tr><tr><td>itemtype</td><td>Training</td></tr></tbody></table>	struct		DropDownList1	USD	amount	2000	date	2007-09-29	itemdetail	ColdFusion	itemtype	Training																																				
struct																																																	
DropDownList1	USD																																																
amount	2000																																																
date	2007-09-29																																																
itemdetail	ColdFusion																																																
itemtype	Training																																																
subform1	<table border="1"><thead><tr><th colspan="2">struct</th></tr></thead><tbody><tr><td>empcode</td><td>10750</td></tr><tr><td>location</td><td>Bangalore</td></tr><tr><td>mgremail</td><td>hkhandel@adobe.com</td></tr><tr><td>mgrname</td><td>Hemant</td></tr><tr><td>name</td><td>sanjeev</td></tr></tbody></table>	struct		empcode	10750	location	Bangalore	mgremail	hkhandel@adobe.com	mgrname	Hemant	name	sanjeev																																				
struct																																																	
empcode	10750																																																
location	Bangalore																																																
mgremail	hkhandel@adobe.com																																																
mgrname	Hemant																																																
name	sanjeev																																																

```
<cfpdfform
  source="expenseclaim-filled.pdf"
  action="read" result="struct"/>
```

Structure will be complex object containing nested CF Array and CF Struct

Repeating subforms and repeating form fields appear as array elements

Acro Form in Acrobat Professional

The screenshot displays the Adobe Acrobat Professional interface. The main window shows a document titled "sample-acro.pdf" with a form titled "Sample Acro Form". The form contains two text fields: "Name" with the value "employee.name" and "Emp Code" with the value "employee.empcode". A "Submit" button is located below the fields. A "Text Field Properties" dialog box is open, showing the "General" tab. The "Name" field in the dialog is set to "employee.name". The "Form Field" is set to "Visible", and the "Orientation" is set to "0 degrees". The "Read Only" and "Required" checkboxes are unchecked. The "Locked" checkbox is also unchecked. A callout box points to the "Name" field in the form, containing the following code:

```
<cfpdfsubform name="employee">  
  <cfpdfformparam name="name" value="chandan">  
</cfpdfsubform>
```

AcroForm - Populate Example

Population using Tags

```
<cfpdfform source="sample-acro.pdf" action="populate" >
  <cfpdfsubform name="employee">
    <cfpdfformparam name="name" value="chandan">
    <cfpdfformparam name="empcode" value="99999">
  </cfpdfsubform>
</cfpdfform>
```

Population using xml data file

```
<cfpdfform source="expenseclaim.pdf" action="populate" xmldata="sample-acro.xfdf"/>
sample-acro.xfdf contains
<xfdf xmlns="http://ns.adobe.com/xfdf/" xml:space="preserve">
  <fields>
    <field name="employee">
      <field name="empcode">
        <value>99999</value>
      </field>
      <field name="name">
        <value>chandan</value>
      </field>
    </field>
  </fields>
</xfdf>
```

Embed CFPDFForm inside CFDocument

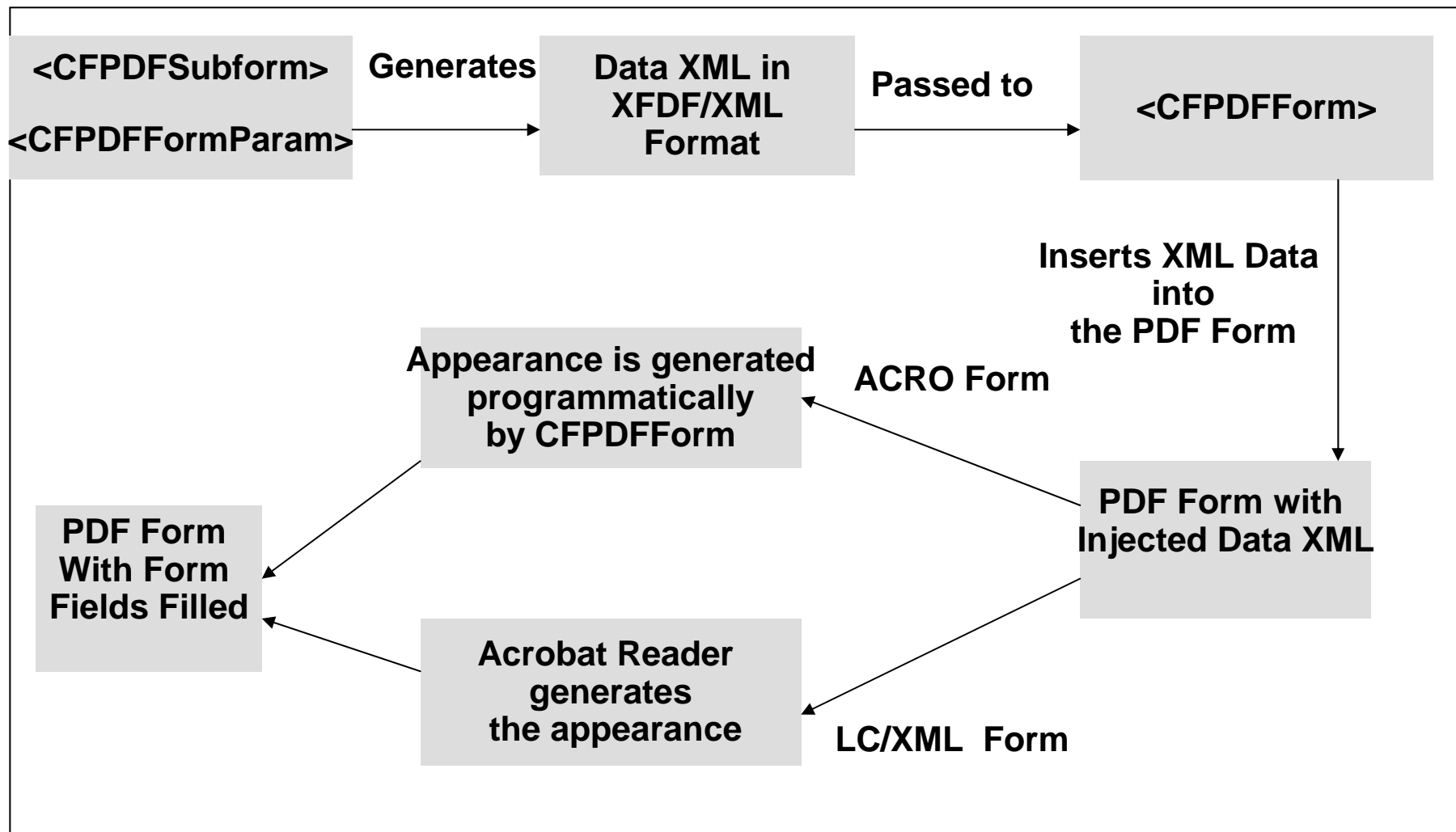
- Only one `<CFPDFForm>` tag can be used in `<CFDocument>`

```
<cfdocument format="pdf" overwrite="true">
  <cfdocumentsection>
    This is the documentation section.
  </cfdocumentsection>
  <cfpdfform source= "expenseclaim.pdf" action="populate">
    <cfpdfsubform name="form1">
      <cfpdfsubform name="subform1">
        <cfpdfformparam name="name" value="Test Name">
        <cfpdfformparam name="empcode" value="12345">
      </cfpdfsubform>
    </cfpdfsubform>
  </cfpdfform>
</cfdocument>
```

TroubleShooting CFPDFForm

- In Acrobat go to - Manage Forms Data> Export Data
 - This gives you xml data for the all fields
- Acrobat > Manage Forms Data> Import Data
 - Importing data XML file to see if it populates the fields as desired
- Field Names are case sensitive
- Check binding in LC forms
 - Normal
 - Global
- If you want Users having Reader to fill up and save the form locally, Enable Usage Rights
- OverWriteData attribute dictates whether old data in the form should be overwritten. If you want to fill form from scratch , it's good to specify OverWriteData ="true"
- Flattening of XFA Forms is not supported completely
 - Workaround - open the filled out form in Acrobat/Reader and save it to disk and run cfpdf flatten
 - Use LC Output WebService for flattening

PDF Form Architecture



CFPRINT

About CFPrint

- <cfprint> tag for Server side printing
- GetPrinterInfo(printerName) - set of attributes that the named printer supports
- Only supported type is PDF
- CF Administrator lists all printers available and default printer in System Information section

<cfprint

```
    type="pdf"
    source="test1.pdf"
    password="password1"
    printerName="path to printer"
    autoRotateAndCenter="yes | no"
    pages="3-6, 11-19"
    copies="number of copies"
    orientation="portrait | landscape | reverse_landscape | reverse_portrait"
    color="yes | no "
    sides="duplex | one_sided | tumble | two_sided_long_edge |
two_sided_short_edge"
    paperSize ="letter | legal | A4 | A5 | B4 | B5 | B4-JIS | B5-JIS"
    jobName="job description"
    coverPage="yes | no"
    quality="draft | high | normal"
    attributeStruct="ColdFusion structure containing additional standard print
request attributes "
    printRequestAttributeSet="ColdFusion
javax.print.attribute.HashPrintRequestAttributeSet object           containing any
javax.print.attribute "
>
```

Some FAQ

- If the PDF file is encrypted, the permissions for the file must be set to “AllowPrinting “
- If a Security Manager is installed, the following permission is required in the coldfusion.policy file to initiate a print job request:
 - `grant { permission java.lang.RuntimePermission "queuePrintJob"; };`
- In Windows systems, ColdFusion server must have PRINTER_ACCESS_USE access rights
- To add fonts for print
 - `-Dorg.jpedal.fontdirs=dirList` (where dirList is a comma-separate list of possible directories)
 - TrueType fonts (.ttf), OpenType fonts (.otf), TrueType font collections (.ttc)
- Does not support JPEG2000
- Forms prefilled with non English text is not supported
- Cannot batch Print jobs

Reference

- CF8 LiveDocs

<http://livedocs.adobe.com/coldfusion/8/htmldocs/help.html>

- DDX Reference

<http://www.adobe.com/support/documentation/en/lifecycle/>

- DDX in ColdFusion - Jeff Chastain

http://www.adobe.com/devnet/coldfusion/articles/ddx_pdf_print.html

Questions ?

[chkumar@adobe.com]

Better by Adobe.™