

# Adobe® Acrobat® FDF for ActiveX Integration with Lotus® Domino®

This document explains how to generate an Adobe Acrobat Forms Data Format, or FDF, using LotusScript and the FDF Toolkit for ActiveX. In either Adobe PDF or Acrobat, an FDF would be analogous to a Lotus Notes document. It contains field information, but doesn't display the fields. The Adobe PDF or Acrobat document, like a Notes form element, is used to display the field information contained in the document. This is advantageous in that we can separate presentation from content and have smaller files to share data.

The sample application uses the support files from the current Acrobat FDF Toolkit (see the Samples\HTMLtoPDF\SupportFiles folder.) The HTML2PDF.pdf form has typical information that you would normally update -- name, address, contacts and date of change.

## PREREQUISITES

### Server

The server will be running Lotus Domino 5.0.x on a Windows NT 4 or Windows 2000 platform. The Windows platform is required for the ActiveX server component to work.

### Client

To create documents, the client will be required to have a web browser and connectivity to the Lotus Domino server. To read documents, the client will need a web browser and Adobe Acrobat or Adobe Acrobat Reader.

## OVERVIEW OF APPLICATION

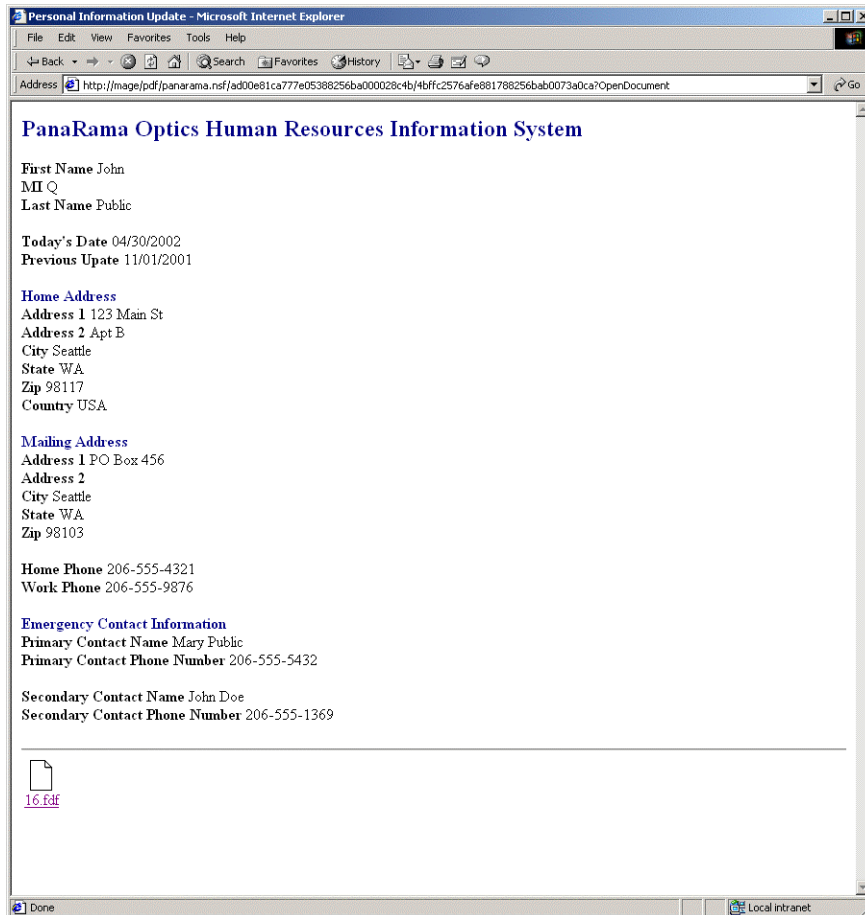
The Lotus Domino application has the same fields that the PDF form contains. We will enter our information from a web browser using the Lotus Domino application residing on the Lotus Domino server. The web form below is in the panarama.nsf Lotus Notes application and is being served from a Lotus Domino server.

The screenshot shows a Microsoft Internet Explorer window titled "Personal Information Update - Microsoft Internet Explorer". The address bar displays the URL: `http://image/pdf/panarama.nsf/7bbe55c9410f9ec988256ba000028dd9?OpenForm`. The main content area displays the "PanaRama Optics Human Resources Information System" form. The form includes the following fields and sections:

- Personal Information:**
  - First Name
  - MI
  - Last Name
  - Today's Date
  - Previous Update
- Home Address:**
  - Address 1
  - Address 2
  - City
  - State
  - Zip
  - Country
- Mailing Address:**
  - Address 1
  - Address 2
  - City
  - State
  - Zip
- Home Phone:**
- Work Phone:**
- Emergency Contact Information:**
  - Primary Contact Name
  - Primary Contact Phone Number
  - Secondary Contact Name
  - Secondary Contact Phone Number
- Submit:**

The status bar at the bottom of the browser window shows "Done" and "Local intranet".

When the browser submits the web form, a WebQuerySave agent will create an FDF and attach it to the Domino document that was just submitted.



#### 4 | Integrating Adobe Acrobat FDF with Lotus Domino

By clicking on the attached FDF file in the Notes document, the user will be able to view the form data using the PDF that is referenced in the /F key.

The screenshot shows a Microsoft Internet Explorer browser window displaying a PDF form. The address bar shows the URL: <http://mage.sea.adobe.com/pdf/panarama.pdf>. The form is titled "PANARAMA Human Resources Information System" and "Personal Information Update". It includes fields for First Name, MI, Last Name, Today's Date, Previous Update, Home Address, Mailing Address, and Emergency Contact Information. The form is displayed in a 109% zoom level.

Personal Information	
First Name	MI
John	Q
Last Name	Public
Today's Date:	April 30, 2002
Previous Update:	November 1, 2001

Home Address	
Address	123 Main St
Apt B	
City	Seattle
State	WA
Zip Code	98117
Country	USA
Home Phone	(206) 555-4321
Work Phone	(206) 555-9876

Mailing Address	
Address	PO Box 456
City	Seattle
State	WA
Zip Code	98103

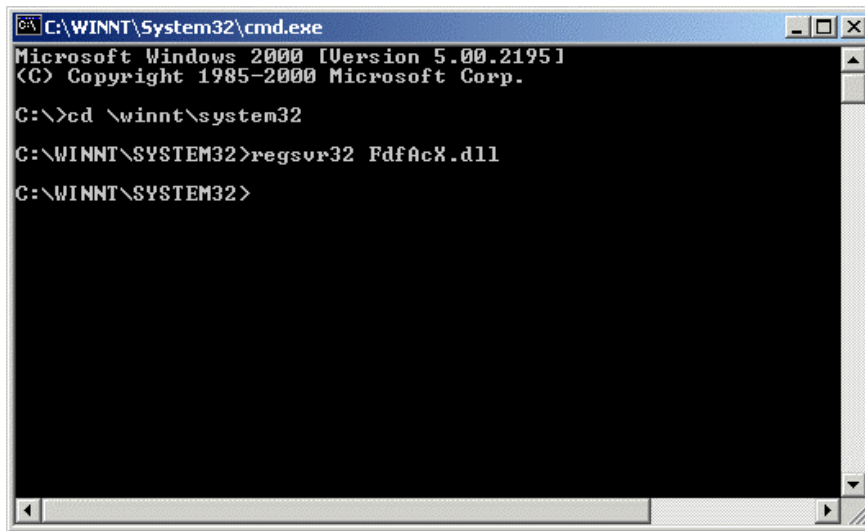
Emergency Contact Information	
Primary Contact Name	Mary Public
Secondary Contact Name	John Doe
Primary Contact Phone Number	(206) 555-5432
Secondary Contact Phone Number	(206) 555-1369

RESET SUBMIT

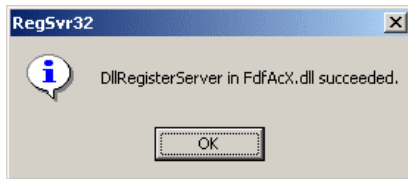
## INSTALLATION

On the server, copy the files FdfAcX.dll and FdfTk.dll to the \winnt\system32 directory. You can find these files in the \FDFToolkit for Windows\Headers and Libraries\WIN\FDFAcX directory where you installed the toolkit.

From a command prompt, change to the \winnt\system32 directory and run `regsvr32 FdfAcX.dll`.



You will see this when you have successfully registered the DLL.



## DECIDING WHERE TO STORE THE PDF

Your users will need access to the PDF that will be used to display the FDF. The FDF will use the /F key to look for the PDF form. The returned FDF data must include an /F key giving the full URL of the targeted PDF file. Acrobat automatically loads the PDF file upon opening the FDF file. You set the /F key with the FDFSetFile method. There are three options for storing the /F key.

### Option One

Store the PDF in a directory share and use the UNC as a reference. For example, we store the panarama.pdf document on the HDC01 server in the PDF share folder. We would use "\\HDC01\public\PDF\panarama.pdf" as the /F key.

- The UNC might not work for all client platforms.
- + Access to the PDF document would be controlled by access to the shared folder.
- + Access to the PDF document could be slightly faster than storing it in a Domino application.

### Option Two

Store it as an attachment in a Page element in your Lotus Domino application. Create a Page element named PDF and attach the panarama.pdf document to it. If your server is named "DOMWEB1" and your application is named "hrchange.nsf", then you could use "http://domweb1/hrchange.nsf/PDF/\$file/panarama.pdf" as your /F key.

- + Allows you to replicate changes to the Page element.
- + Gives you security over who can use the application and have access to the PDF form.

### Option Three

Store it the /HTML directory on the Domino server. If your Lotus Domino server is named DOMWEB1 and stores its data files in "d:\lotus\domino\data\", then you can put the file in d:\lotus\domino\data\domino\html\pdf and refer to it as "http://domweb1/pdf/panarama.pdf."

- + Access to the PDF document could be slightly faster than storing it in a Domino application.

## USING THE FDF TOOLKIT IN LOTUSSCRIPT

Below is the code for the agent that generates the FDF from the Notes document.

```
'webUsingActiveX agent:

Option Public

Sub Initialize
'
' Set up for using the FDF Toolkit for ActiveX
'
Dim FdfAcX As Variant
Dim outputFDF As Variant
Dim sPdfFileName As String, sFdfFileName As String, sFilePath As String
Set FdfAcX = CreateObject("FdfApp.FdfApp")
Set outputFDF = FdfAcX.FDFCreate()
'
' % REM
'
' Change these paths to reflect your server environment
' Ideally, you could use a Lotus Notes profile document to store this
' information.
'
' Below are the three options for storing the PDF that was discussed in
' the documentation for this application.
'
' Option 1 : UNC Directory Share
' This option oddly asks if you would like to save on close.
'
' sPdfFileName = "\\Mage\domdata\domino\html\pdf\panarama.pdf"
'
' Option 2 : Store in a Lotus Notes Page element
'
sPdfFileName = "http://mage/pdf/panarama.nsf/PDFs/$file/panarama.pdf"
'
' Option 3 : Store in the Domino HTML directory
'
' REM %
'
' sPdfFileName = "http://mage/pdf/panarama.pdf"

sFilePath = "d:\domdata\domino\html\pdf\tmp\"
'
' Set up for getting Lotus Notes information
'
Dim session As New NotesSession
Dim contextDoc As NotesDocument
Dim dateTime As NotesDateTime
Dim tempNumber As Single
Dim rounded As Integer
Dim fileName As String
Set contextDoc = session.documentContext
'
```

```
' This will create a random number between 0 and 100.
' The number will give
' us a good chance of not temporarily having the same file name in the tmp
' directory at the same time. The file will briefly be in the tmp
' directory,
' attached to the Notes document and then deleted.
,
Randomize
temp = Rnd
rounded = Round((temp * 100), 0)
fileName = Cstr(rounded)
sFdfFileName = sFilePath & fileName & ".fdf"
,
' FDFSetValue will set the value you pass it. See page 139 of the
' FDF Toolkit Overview and Reference.
,
outputFDF.FDFSetValue "FirstName", contextDoc.FirstName(0), False
outputFDF.FDFSetValue "MidInit", contextDoc.MidInit(0), False
outputFDF.FDFSetValue "LastName", contextDoc.LastName(0), False
,
' Get the value of the date fields. The DateOnly property
' returns a string.
,
Set dateTime = New NotesDateTime( contextDoc.HireDate(0) )
outputFDF.FDFSetValue "HireDate", dateTime.DateOnly, False
Set dateTime = New NotesDateTime( contextDoc.ModDate(0) )
outputFDF.FDFSetValue "ModDate", dateTime.DateOnly, False
,
outputFDF.FDFSetValue "Address1", contextDoc.Address1(0), False
outputFDF.FDFSetValue "Address2", contextDoc.Address2(0), False
outputFDF.FDFSetValue "City", contextDoc.City(0), False
outputFDF.FDFSetValue "State", contextDoc.State(0), False
outputFDF.FDFSetValue "ZipCode", contextDoc.ZipCode(0), False
outputFDF.FDFSetValue "Country", contextDoc.Country(0), False
,
outputFDF.FDFSetValue "MAddress1", contextDoc.MAddress1(0), False
outputFDF.FDFSetValue "MAddress2", contextDoc.MAddress2(0), False
outputFDF.FDFSetValue "MCity", contextDoc.MCity(0), False
outputFDF.FDFSetValue "MState", contextDoc.MState(0), False
outputFDF.FDFSetValue "MZipCode", contextDoc.MZipCode(0), False
,
outputFDF.FDFSetValue "HomePhone", contextDoc.HomePhone(0), False
outputFDF.FDFSetValue "WorkPhone", contextDoc.WorkPhone(0), False
,
outputFDF.FDFSetValue "Emer1Contact", contextDoc.Emer1Contact(0), False
outputFDF.FDFSetValue "Emer1Phone", contextDoc.Emer1Phone(0), False
outputFDF.FDFSetValue "Emer2Contact", contextDoc.Emer2Contact(0), False
outputFDF.FDFSetValue "Emer2Phone", contextDoc.Emer2Phone(0), False
,
' Tell it which file we're writing to and do a save.
,
outputFDF.FDFSetFile sPdfFileName
outputFDF.FDFSaveToFile sFdfFileName
,
' Always close your open FDF files.
,
outputFDF.FDFClose
```



```
'  
' This will attach the FDF file we just closed to our Notes document  
' and then delete it from the tmp directory.  
'  
    Call MoveFdfToAttachment(contextDoc, sFdfFileName)  
End Sub  
  
Public Function MoveFdfToAttachment(doc As NotesDocument, fileName As  
String) As Integer  
  
    '// move each attachment to a rich text field.  
  
    Dim attachItem As New NotesRichTextItem(doc, "fdfAttachment")  
    Dim attachObj As notesEmbeddedObject, newObjName As String  
    Call attachItem.EmbedObject(EMBED_ATTACHMENT, "", fileName, "")  
    Kill fileName  
    MoveFdfToAttachment = True  
End Function
```

## REFERENCE MATERIALS

### FDF Toolkit

<http://partners.adobe.com/asn/developer/acrosdk/forms.html>

### PDF Reference

<http://partners.adobe.com/asn/developer/acrosdk/docs.html#filefmtspecs>

### LotusScript for Visual Basic Programmers IBM Redbook

<http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg244856.html>