



# **Fox InCloud**

## **DELIVERING A FILE TO THE USER**

How-To Guide

## DELIVERING A FILE TO THE USER

# TYPICAL SCENARIOS

Delivering a file to the user is a common occurrence in situations like:

- Producing a report in PDF,
- Exporting data in formats such as CSV, XLS, XML, etc.,
- Building a text file you want to supply to the user.

As file saving procedure differs between LAN and Web, implementation needs be slightly different. Fortunately, FoxInCloud leverages these differences to come up with a common, standard and convenient procedure.

In such a situation, typical code could be:

```
LOCAL lcFileSrce, lcFileDest
<some code producing lcFileSrce>
lcFileDest = PutFile('Save File ...', m.lcFileSrce, justExt(m.lcFileSrce))
IF NOT EMPTY(m.lcFileDest)
    COPY FILE (m.lcFileSrce) TO (m.lcFileDest)
ENDIF
DELETE FILE (m.lcFileSrce)
```

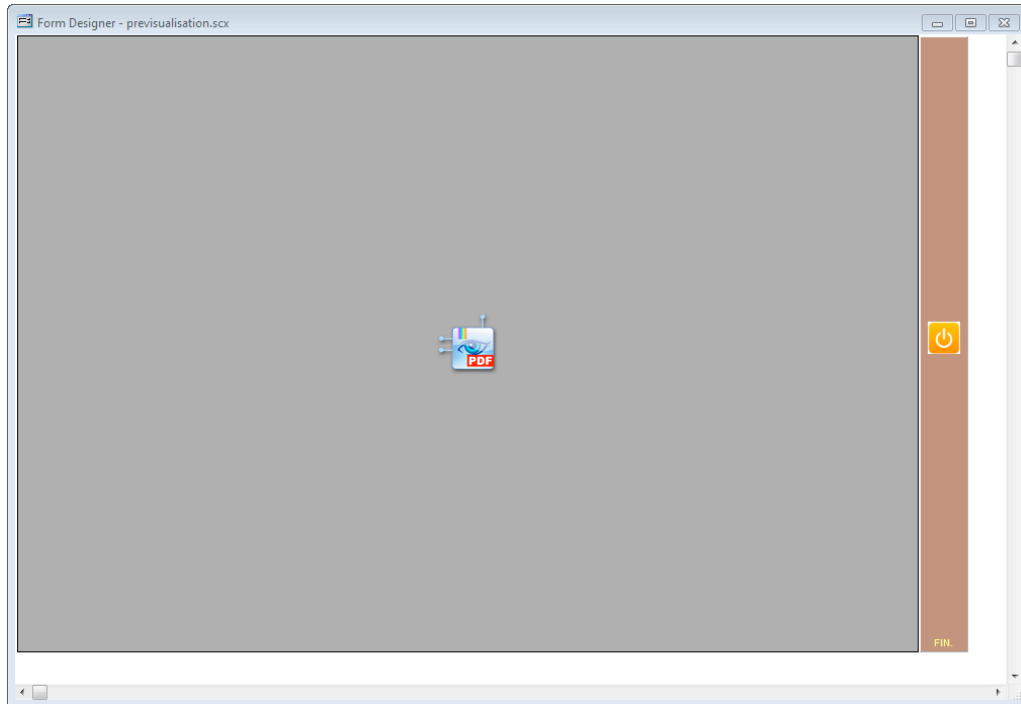
A variation occurs when producing a PDF file that user can view inside a free viewer such as 'Acrobat Reader' or 'PDF xChange Viewer':

```
LOCAL lcFileSrce, loListener AS ReportListener && e.g. XFRX™
loListener.someMethod(m.lcFileSrce) && Instructs listener where to output
REPORT FORM myReport OBJECT loListener
DO FORM myReportViewerForm WITH m.lcFileSrce && Modal
DELETE FILE (m.lcFileSrce)
```

In this exemple, '**myReportViewerForm**' holds a PDF viewer object as illustrated below:

## DELIVERING A FILE TO THE USER

# Typical scenarios ...



**Figure 1 - Sample PDF display form**

This display Form has an **OLEcontrol** hosting an **OLEClass** such as '**PDFXCviewAx.CoPDFXCview.1**'<sup>1</sup>, or '**AcroPDF.PDF.1**'<sup>2</sup>

The **Init()** method of this form is fairly simple:

```
LPARAMETERS p_file
IF PCOUNT()>0 AND FILE(m.p_file)
    THISFORM.OLEpdf.Src = m.p_file
ENDIF
```

<sup>1</sup> PDFxChange Viewer freely available on <http://www.tracker-software.com/>.

<sup>2</sup> Adobe Acrobat Viewer freely available <http://www.adobe.com/>

## DELIVERING A FILE TO THE USER

# ADAPTING FILE SAVE

If your purpose is simply to deliver the file to the user, you just need to replace your File Save code by a single call to `aw.vcx!awFrm:wFileSaveAs()` method.

Initial code: ...

```
lcFileDest = PutFile('Save File ...', m.lcFileSrce, justExt(m.lcFileSrce))
IF NOT EMPTY(m.lcFileDest)
    COPY FILE (m.lcFileSrce) TO (m.lcFileDest)
ENDIF
DELETE FILE (m.lcFileSrce)
```

... can simply be replaced by:

```
LOCAL lcResult
IF NOT thisForm.wFileSaveAs(
    m.lcFileSrce; && The file your program just produced
    , @m.lcResult; && By reference (@) to get result back
)
    thisForm.wMessageBox(m.lcResult) && or anything else you see fit
ENDIF
```

`.wFileSaveAs()` does all the job for you:

- **In LAN mode**, prompts user for location where file should be saved, and if `<lcFileSrce>` is a temporary file (produced less than 2 seconds ago), deletes `<lcFileSrce>`
- **In Web mode**, copies the file to your web site's temp Folder, sends back a 'Save File' dialog to the user, deletes source file if temporary as mentioned earlier, deletes previous expired files from web site's temp Folder.

If any error occurs during this process, `.wFileSaveAs()` returns `.F.` and `lcResult` holds the reason for this error.

## DELIVERING A FILE TO THE USER

# ADAPTING A PDF DISPLAY FORM

In this second scenario, your application displays the PDF produced in a form such as the one described earlier.

If this form was processed by FoxInCloud Adaptation Assistant, the `OLEcontrol` hosting the PDF viewer `OLEClass` was sub-classed into the corresponding `aw.vcx` base class: `awOLEPDFxc`, `awOLEPDFacro` or the like.

Due to the non-VFP behavior of OLE classes, the `aw.vcx` PDF viewer base classes implement file set up into the `'Tag'` property instead of the `'Src'` property.

Consequently, form's `.Init()` code shown above:

```
LPARAMETERS p_file
IF PCOUNT()>0 AND FILE(m.p_file)
    THISFORM.OLEpdf.Src = m.p_file
ENDIF
```

... Should be adapted as follows:

```
LPARAMETERS p_file
IF PCOUNT()>0 AND FILE(m.p_file)
    THISFORM.OLEpdf.Tag = m.p_file
ENDIF
```

A `.Tag_assign()` method takes care of all necessary chores, including **DELETING** source PDF file if temporary, based on the same criterion as above explained.

The code producing the PDF file before form call needs no adaptation.

On next page is complete sample code producing an output file and displaying if it's a PDF, else prompting user for save.



# SAMPLE CODE

```

* Set up Report Listener
loSession=EVALUATE([xfrx("XFRX#LISTENER")]) && XFRX™ is used here
IF 0 = ICase(
  Inlist(cMode,"DOC","ODT","ODS","XML","TXT","HTML","CPDF","RTF","XFF"),;
  m.loSession.SetParams(m.cFileXfrx,,.T.,wlWeb(),,m.cMode),;
  m.cMode="PDF",;
  m.loSession.SetParams(m.cFileXfrx,,.T.,wlWeb(),,"PDF",,,;
  ,NOT EMPTY(m.cNomPDFMerge)),;
-1)

IF m.cMode="PDF" && regular PDF
  m.loSession.setPermissions(.T., .F., .F., .F.)
ENDIF
IF m.cMode="CPDF" && protected PDF
  loSession.setPasswords("myAccount","myPassword")
  loSession.setPermissions(.T., .F., .F., .F.)
ENDIF

loSession.Successor = NEWOBJECT("myReportListener","myClasses.vcx")

* Produce Label or Report
IF m.cType='LABEL'
  LABEL FORM (m.cTmpReport) OBJECT m.loSession
ELSE
  REPORT FORM (m.cTmpReport) OBJECT m.loSession
ENDIF
loSession.finalize

IF m.cMode="PREVIEW" AND Inlist(m.cMode, "PDF", "CPDF")
  * Display PDF in a preview form
  =THISFORM.wForm(
    "preview.scx" && preview form
    , wlLan(); && modal in LAN mode, modeless in WEB mode
    , m.cFileXfrx; && PDF just produced
  )
ELSE
  * Prompt user to save file
  =THISFORM.wFileSaveAs(m.cFileXfrx)
ENDIF

DELETE FILE (m.cFileXfrx)
&& safe in Web mode as FoxInCloud copies file to site's Temp directory.
ENDIF

```

**Sample code producing a PDF report, and displaying it in a preview form.**



# **Fox InCloud**

[www.foxincloud.com](http://www.foxincloud.com) | [contact@foxincloud.com](mailto:contact@foxincloud.com)

---

Abaque | SARL au capital de 41 200 € | RCS Paris B 379 373 566 | 66, rue Michel Ange  
75016 Paris +33 (0)9 53 41 90 90 | [www.zenbuyer.com](http://www.zenbuyer.com) | [info@zenbuyer.com](mailto:info@zenbuyer.com)