

# FDK Changes for FrameMaker 7.1

## Import PDF

The API scriptable property `FS_PDFPageNum` can be used by FDK clients to import a particular page of a PDF document. The following example illustrates the use of this property.

### *A Simple Example*

```
IntT    index;
IntT    pageNum;
F_PropValsTparams;

/* Get the default params list for import */
params = F_ApiGetImportDefaultParams();

/* Get the index of the FS_PDFPageNum property in this list */
index = F_ApiGetPropIndex(&params, FS_PDFPageNum);

/* Specify the page number to be imported. Say you want to import page 3 of
the PDF document*/
pageNum = 3;
params.val[index].propVal.u.ival = pageNum;
/* call F_ApiImport : Please refer to F_ApiImport documentation for details*/
...
F_ApiImport(...);
...
```

## XML enhancement

FrameMaker 7.1 enhanced XML roundtripping by adding conditional text and external cross-references support. The updated struct.lib makes these features available to structured clients. When clients link to the struct.lib supplied with FDK 7.1, these new features become part of the client.

FrameMaker 7.1 added support for opening XML files as XML. As a result .xml files retain their extensions when opened in FrameMaker. These files get saved by default to XML. FrameMaker 7.1 also locks the XML files while opening.

This is for the clients that use `F_ApiSimpleSave` to save xml files as fast. Due to the changes in FrameMaker 7.1, some clients may see different behavior as compared to FrameMaker 7.0.

Here is an example scenario -

An xml client opens an xml file say file1.xml, edits the file and then saves the same file using `F_ApiSimpleSave`.

FrameMaker 7.0, opens file1.xml as file1.fm and hence `F_ApiSimpleSave` saves the file by the name file1.fm.

FrameMaker 7.1, opens file1.xml as file1.xml and locks the file. Now, if `F_ApiSimpleSave` is used to save the file, then you get an error as the file is locked and cannot be saved.

Please note that this will happen only when the file name argument to `F_ApiSimpleSave` is the name of a file that is open. If locking is turned off, then the file will be saved to fast format.

The workaround is to give a name that is different from the name of the file being opened. If the objective is to save to xml, then use `F_ApiSave` instead of `F_ApiSimpleSave`.

The following two samples illustrate how to open and save an XML.

*An example to Open an XML*

```
VoidT openXML ()
{
    F_PropValsT params, *returnParamsp = NULL;
    IntT i;
    /* Get default open properties. Return if it can't be allocated. */
    params = F_ApiGetOpenDefaultParams();
    if (params.len == 0)
    {
        F_Printf(NULL, (StringT) "File not opened. Unable to allocate memory.\n");
        return;
    }

    /* Set properties to open an XML document*/
    /*Specify XML as file type to open*/
    i = F_ApiGetPropIndex(&params, FS_OpenAsType);
    params.val[i].propVal.u.ival = FV_TYPE_XML;

    /*Allow user to select a file to open*/
    i = F_ApiGetPropIndex(&params, FS_ShowBrowser);
    params.val[i].propVal.u.ival = True;

    /* Specify the XML application to be used when opening the document.*/
    i = F_ApiGetPropIndex(&params, FS_StructuredOpenApplication);
    params.val[i].propVal.u.sval = F_StrCopyString((StringT) "XDocBook");

    if (!(F_ApiOpen("", &params, &returnParamsp)))
    {
        F_ApiAlert("Couldn't open file.\n", FF_ALERT_CONTINUE_NOTE);
        /* Deallocate property lists*/
        F_ApiDeallocatePropVals(&params);
        F_ApiDeallocatePropVals(returnParamsp);
        return;
    }
    else
    {
        /*Deallocate property lists*/
        F_ApiDeallocatePropVals(&params);
    }
}
```

```
        F_ApiDeallocatePropVals(returnParamsp);  
        return;  
    }  
}
```

*An example to Save an XML*

```
VoidT saveXML ()
{
    F_PropValsT params, *returnParamsp = NULL;
    IntT i;
    F_ObjHandleT docId = F_ApiGetId(0,0,FP_ActiveDoc);
    /* Get default save properties. Return if it can't be allocated. */
    params = F_ApiGetSaveDefaultParams();
    if (params.len == 0)
    {
        F_Printf(NULL, (StringT)"Couldn't get save parameters!\n");
        return;
    }

    /*Specify XML as file type to save*/
    i = F_ApiGetPropIndex(&params, FS_FileType);
    params.val[i].propVal.u.ival = FV_SaveFmtXml;

    /* Specify the XML application to be used when saving the document.*/
    i = F_ApiGetPropIndex(&params, FS_StructuredSaveApplication);
    params.val[i].propVal.u.sval = F_StrCopyString((StringT) "XDocBook");

    if (!(F_ApiSave(docId, "sample.xml", &params, &returnParamsp)))
    {
        F_ApiAlert("Couldn't save file.\n", FF_ALERT_CONTINUE_NOTE);
        /* Deallocate property lists*/
        F_ApiDeallocatePropVals(&params);
        F_ApiDeallocatePropVals(returnParamsp);
        return;
    }
    else
    {
        /*Deallocate property lists*/
        F_ApiDeallocatePropVals(&params);
        F_ApiDeallocatePropVals(returnParamsp);
        return;
    }
}
```

The first example opens an XML file. In FrameMaker 7.0, the XML file will open as a fm file. In FrameMaker 7.1, the same file will open as an XML. The second example saves the active document as “sample.xml”. In FrameMaker 7.0, the active fm document will be saved to disk as sample.xml keeping the fm document active. In FrameMaker 7.1, the open XML document will be saved to sample.xml making sample.xml active.

## ***FP\_ImportHint Strings***

The following list enumerates the hint strings to be used for the new import filters added in FrameMaker 7.1

- To import a PDF file use the hint "0001FRAMPDF "
- To import a Photoshop file use the hint "0001ADBIPSD "
- To import JPEG2000 files  
For a file with extension
  - JP2 use the hint "0001ADBIJP2 "
  - For a file with extension JPX use the hint "0001ADBIJPX "
  - For a file with extension J2C use the hint "0001ADBIJ2C "
  - For a file with extension J2K use the hint "0001ADBIJ2K "
  - For a file with extension JPC use the hint "0001ADBIJPC "
  - For a file with extension JPF use the hint "0001ADBIJPF "
- To Open a Quark file use the hint "0001ADBIQXD "
- To Open a PageMaker 7.0 document use the hint "0001ADBIEMD "
- To Open a PageMaker 7.0 template use the hint "0001ADBIEMT "
- To Open a PageMaker 6.5 document use the hint "0001ADBIP65 "
- To Open a PageMaker 6.5 template use the hint "0001ADBIT65 "

The filter name specifies a four byte code. If the filter name contains fewer than four alphanumeric characters, the remaining bytes must be filled out with spaces.

