

This is a demonstration of a situation where it is desirable to maintain two (or more) separate databases but to have data transfers between them. In the example, invoices entered into an Inventory database will be transferred to an Accounts Payable database for payment.

4D has always made it possible to do this sort of thing. However, the FilePack externals by Bob Puglino allow such operations gracefully than before. Further, these externals allow a higher degree of control over the data integrity than is available with the native 4D commands.

The example...

Demo INVf is a bare bones Inventory database. It allows entry (but not withdrawal!) of items.

Open the database and choose the Receivers file. Receivers are the documents that enter items into inventory. They are also the invoices from the vendor the items are purchased from. You will find some receivers already entered. Note the Exported column. Receivers already been exported to the AP (Accounts Payable) database have a number next to them. There are four records which haven't been exported yet.

You may create some more receivers if you wish.

When you are ready execute the procedure called "Do Export". You are prompted to name the exported data document. The database will search for any un-exported records in the Receivers file and export them.

Now open the Demo AP database.

This is a bare-bones Accounts Payable database.

Click on the Invoices file. You will see several invoices already in place.

To import the data from the inventory file execute the procedure called "DoImport". You will see an open dialog box allowing you to choose any text file. Although you may choose any text file you may only import files "addressed" to that database. More about this in a moment.

Navigate to the data file you created from inventory and open it. The information in this file is read into the AP database and the appropriate records will be created. (You can create checks for these invoices for fun; execute the "Pay Invoices" procedure.)

After the data from inventory is imported the name of the data file will have a "x" placed before it. This indicates to you that this data has been imported.

That's it. This technique is useful in many situations:

- Inventory to accounts payable
- Payroll to accounts payable
- Any module to a General Ledger
- Order entry to inventory
- Accounts receivable to Contacts

(I'm sure there are many others but I focus on accounting type uses.)

How it works...

The key issue for this kind of transfer is to insure data integrity. Three major concerns arise. You want to make sure that data gets to the correct place (you don't want accounts payable to be able to import accounts receivable data). You want to

make sure that data only gets imported once. You want to make sure that even if you have two or three different data sets of the same database (perhaps you maintain payroll for three different businesses) that you don't inadvertently import company A's data into company B's database.

The functions in FilePack 1.1 make this level of data integrity possible.

How to insure data gets to the correct place...

Open the Demo AP*f* database in the User environment. Select the SetUp Info file to view. You should see one record showing the field 'myABRV'. In this case it is "ABCD".

This is the abbreviation for this dataset.

Open the Demo INV*f* database and go to the Design environment.

Note the file named "ExportSetUp". (This information could be added to a constants file, if you use them.)

The required piece of information is the "Target". This is a four character abbreviation that will correspond to an abbreviation identifying the target database. Notice that the target is "ABCD", the abbreviation for the accounts payable dataset.

The other two fields are optional but allow you greater control over data exports. The ExportNumber field is a counter field. Demo INV assigns a unique number every time it creates an export document. This would be important to you if you want to track data export documents very closely.

The third field is a simple Boolean field that you could use as part of the set up information. This would either allow or disallow exports.

The abbreviation is the key. The procedure, ExportRCVRs, changes the creator type of the data document to the abbreviation of the target dataset. Doing this affects nothing significant about the document.

When Demo AP opens the data document the procedure DoImport checks the creator of the document opened. If the creator of the data document doesn't match the abbreviation specified the import is not allowed.

Since the creator attribute is not easily changed by the average user this allows the developer a reliable way of coding exported data. The administrator of the databases will need to be very careful about assigning the abbreviations. As long as the abbreviations aren't duplicated the data will get to the correct place.

FilePack makes it easy to rename documents. Thus, once an import is done Demo AP renames the data file by placing a "x" in front of the document name. You could also elect (or offer the choice) of deleting the document, renaming it or changing the creator attribute again.

Setting Up a Data Document

You may put anything in your data documents you want to. Here are my suggestions for setting the up.

The first line of your data document should be a header that holds all the interesting information about the data to come. This should include the number of records to be imported, the date they were exported and where they should be going. You can include anything else you like, sums of various types, the name of the user doing the transfer, the address of the sending database, whatever.

I like to make the very last line some sort of end of file marker like the string "**/*".

Here are the contents of a demo transfer document from the Demo Inv.

```
3      ABCD      Receivers to A/P  11/17/90
HS    R 0006    11/17/90          1250
KB    R 0007    11/17/90          1000
JB    R 0008    11/17/90          3000
*/*
```

The first line is the header. The first item is the number of records being transferred (3). Next is the address of the receiving db (for further security), a description of what is happening and the data.

The following three lines are the data: vendor code, invoice #, invoice date and amount.

The last line is my end of file marker.

Use It Yourself...

The procedures in these examples are commented and very generic. You may easily adapt them to your applications.

The FilePack externals written by Bob Puglino are included here also. Bob has placed these externals in the public domain and they are yours to use as you wish. Bob's documentation is included.

Let me know what you think of this example. I really would like to hear your comments.

Kirk Brooks
CI\$ 71551,343
(505) 256-7120