HFFlow32

Addendum to halFILE for Windows?
Document Storage & Retrieval

Version 2.1
May 17, 2004

Doc # hfw0022_21

Table of Contents

HFFlow32: Description of Application

HFFlow32.exe, an add-on utility to halFILE (version 2.1 or later), is used to keep the records two databases in sync. When a document is modified in one database, HFFlow32 can be automatically executed to update other documents having the same key. The document that are updated can be in the same database that the modification was made and/or in a second database. HFFlow32 can be set up to automatically execute during either the index and/or search functions of halFILE.

HFFlow32 is typically used as follows:

1. In a workflow scenario, one database might be used to collect the initial information which will start the process of generating a greater number of other documents. An identical second database is set up to collect all of the additional document types and after the initial record is created in the first database all additional maintenance is performed in the second database. Using option 2 of HFFlow32, the record created in the first database (along with any documents that are indexed to that record) could be automatically added to the second database.
2. Using option 1 of HFFlow32, any maintenance performed on the second database could be posted to other like records in the second database and also be passed back to the first database.
3. By using a combination of the two options, all databases can remain synchronized.

Program Features/Limitations

HFFlow32 can be run in two very different ways.

1. Option 1 – update all other records in the same database with all of the values of the first record with the exception of any fields defined in the INI file to skip. The program will then update all records in a second database with the same values, taking into account the same exceptions that are defined in the INI file.
2. Option 2 – add a new record to the second database with all of the values of the record at hand, taking into account any exceptions to skip defined in the INI file. Only add the record to the second database if a record does not exist for the two key fields defined in the INI file. Any documents that are attached to the record in the first database can be optionally copied and linked to the new record in the second database.

HFFlow32 Setup

1. Copy HFflow32.exe into the halFILE program folder.
2. Create HFFlow32.ini in the halFILE program folder and set up the appropriate parameters as described below.
3. For automatic updates during indexing, add the appropriate IndexExecute entries in the halfile.ini located in the c:\windows folder of each workstation that will automatically launch the program HFflow32.exe during the index process.
4. For automatic updates during search/update, add the appropriate SearchExecute entries in the hfglobal.ini located in the halFILE program folder.

HFFLOW32.INI

HFFlow32.ini should be placed in the halFILE program folder (where hfflow32.exe exists). The following describes the parameters in this file (The description of each parameter within the parentheses is not a part of the file).

[XYZHFWDD]	(the database section where XYZ is the application id and DD is the Document Type id for a halFILE database. Each database to be used in a HFFlow32 operation must be configured here.)
OPTION=X	(where X is either a 1 or a 2. Option 1 will update all other records in the same database (other than those fields excluded by the DB1_SKIP1 values explained below) before updating all records in the second database (defined by APP2= and DB2= below). Option 2 will add a new record to the second database (defined by APP2= and DB2=) only if a record does not already exists for the values defined by DB1_KEYFIELD1 and DB1_KEYFIELD2.)
APP1=XYZ	(where XYZ is the application id of the database that will be the source of the data copied.)
DB1=DD	(where DD is the Document Type id for the halFILE database that will be the source of the data copied.)

DB1_FIELD1=	(the name of first user-defined data field for the application id and document type id defined with APP1 and DB1 above. HFFlow32 can handle up to 50 user-defined data fields by defining additional parameters like this: DB1_FIELD2=, DB1_FIELD3=, up to DB1_FIELD50.)
DB1_KEYFIELD1=	(the name of the user-defined field that defines the unique field that all like records will be updated.)
DB1_ME1=	(the name of first user-defined multi-entry data field for the application id and document type id defined with APP1 and DB1 above. HFFlow32 can handle up to 50 user-defined data fields by defining additional parameters like this: DB1_ME2=, DB1_ME3=, up to DB1_ME50.)
DB1_SKIP1=	(the name of the first user-defined data field that one wishes to bypass updating in both the other records within the same table or posting to the other table defined by APP2= and DB2= below. HFFlow32 can handle up to 50 user-defined data fields by defining additional parameters like this: DB1_SKIP2=, DB1_SKIP3=, up to DB1_SKIP50.)
APP2=XYZ	(where XYZ is the application id of the second database that will be updated with the information from the first database defined by APP1= and DB1=)
DB2=DD	(where DD is the Document Type id of the second database that will be updated with the information from the first database defined by APP1= and DB1=.)

Additional parameters when using OPTION=2

IMAGES=X	(where X is either TRUE or FALSE. If set to TRUE then HFFlow32 will copy all documents attached to the document number at hand to the second database defined by APP2= and DB2=.)
DB1_KEYFIELD1=	(the name of the user-defined field that, when used in conjunction with the value defined by the optional parameter DB1_KEYFIELD2, will determine whether the record from the first database will be added to the second database.)
DB1_KEYFIELD2=	(the name of the second user-defined field that, when used in conjunction with the value defined by DB1_KEYFIELD1, will determine whether the record from the first database will be added to the second database. NOTE: This parameter is optional and if not found will then resort back to the value in DB1_KEYFIELD1.)

HALFILE.INI

Halfileini is used for many halFILE parameters. It resides in the c:\windows program folder. The following parameters will configure either the index and/or search functions to automatically execute the HFFlow32 program. (The description of each parameter within the parentheses is not a part of the file)

Please notice the differences in the 'IndexExecute=' and 'SearchExecute=' when running option 1 versus option 2.

[XYZHFWDD]	(the database section where XYZ is the application id and DD is the Document Type id for a halFILE database. Each database to be used in a HFFlow32 operation must be configured here.)
IndexExecute=C:\halfile\hfflow32.exe XYZHFWDD;%DocNum%	(path and file name to execute. Do not use long file names)

NOTES:

The 'IndexExecute=' in this instance would automatically run from within the index function of halFILE. XYZHFWDD would be set up in HFFLOW32.INI to run option 1 (update all like records in the same database and then update like records in a second database).

The next example would automatically run the 'IndexExecute=' during the index function of halFILE and is an example of XYZHFWDD set up in HFFLOW32.INI to run option 2 (add the record from database 1 to database 2 only if a record does not exist for key fields 1 and 2.

IndexExecute=C:\halfile\hfflow32.exe
XYZHFWDD;%DocNum%;%APPL%;%DOCTYPE%;%BASKET%;DestBasketName

The above should be all on one line inside the Halfile.ini file. Do not use long file names when defining the location of the executable hfflow32.exe.

The ‘DestBasketName’ should be replaced with a valid basket name for the database defined inside the HFFLOW32.INI file for APP2= and DB2=. This basket should be unique for each different user who will be using option 2 of HFFlow32.

HFGLOBAL.INI

The ‘SearchExecute=’ in hfglobal.ini operates exactly like the ‘IndexExecute=’. If modifications are made to any data fields during the search function of halFILE, when the changes are saved then the ‘SearchExecute=’ line will be automatically executed by halFILE. Example:

```
[XYZHFWDD]
SearchExecute=C:\halfile\hfflow32.exe XYZHFWDD;%DocNum%
```