

Exporting EndNote Records to MS Access and Importing Endnote Records From Access Using Tab Delimited Files in Both Directions

Robert W. Gear
University of California, San Francisco

Copyright © 2002

The instructions in this document are provided "as is," with no promise or warranty expressed or implied as to their usefulness for any particular purpose. This document and the accompanying source code is copyrighted (2002) and may not be sold or used for any commercial purpose without the permission of the author. However, permission is hereby granted for free distribution for educational purposes.

Contents

Scope.....	2
Compatibility across versions.....	2
Tagged files vs Tab Delimited files.....	2
The carriage return (CR) problem in Tab Delimited files.....	2
Preparing the EndNote library for export.....	3
Creating a CR-free EndNote Library.....	4
Creating a Tab Delimited File in EndNote for Export to Access.....	6
Setting Up Access to accept the imported file.....	10
Importing the Tab Delimited file into Access.....	11
Preparing to Export a Tab Delimited File from Access.....	13
Manual method of exporting a Tab Delimited File from Access.....	13
Exporting a Tab Delimited File using the "ExportToEndnote" query.....	16
Automatic method of exporting a Tab Delimited File from Access.....	18
Troubleshooting the module.....	20
Using basEndnote.....	21
Importing the Tab Delimited File into EndNote.....	22
APPENDIX A: How (and why) to copy record numbers to another field in EndNote....	23
APPENDIX B: basEndNote source code.....	26

Scope

This document covers exporting records from EndNote to a Tab Delimited file that can be imported by Access, importing records into Access from a Tab Delimited file created by EndNote, exporting records from Access to a Tab Delimited file that can be imported by EndNote, and importing records to EndNote from a Tab Delimited file created by Access. The procedures described can be used to export records from Access even if the data did not originate in EndNote.

Compatibility across versions

The examples in this document were taken from EndNote 6 and Microsoft Access XP. Although I am not an expert on all aspects of either EndNote or Access (meaning there may be compatibility issues of which I am not aware), I believe that these directions are valid for Access 97, Access 2000 and Access XP as well as for EndNote versions 3-6.

Different versions of EndNote support different numbers and types of fields. Specifically, EndNote 6 supports all fields of previous versions of EndNote plus two new fields, the Image field and the Caption field. EndNote 4 and EndNote 5 support all fields except these two new fields. EndNote 3 has fewer fields. It is important to know which version of EndNote you are running—and which fields are supported—in order to ensure accurate data alignment with Access. More will be said about this later.

The procedures outlined in these instructions assume that you want to export all EndNote fields to Access (or, from Access to EndNote). I strongly recommend you do this, even if you don't currently use all fields.

Tagged files vs Tab Delimited files

EndNote supports many export styles and import filters. Export styles are mostly used to specify formatting settings for document bibliographies, but they are also used to specify settings for exporting EndNote records from an EndNote library to text files for importation into other software such as a database or other reference manager.

Tagged files are text files that are encoded to allow importing programs such as EndNote to place the correct information into each field and also to indicate the end and/or beginning of a record. EndNote uses a tagging system based on the Refer set, but can also import text files formatted according to other tag sets such as Medline. An example of a Medline tag denoting an author is “AU - Doe J”

Tab Delimited files do not depend on tags. An entire reference is contained on each line of the file and the fields are separated by <tab> characters (ANSII code 09). The end of a record is denoted by the carriage return <CR> character (ANSII codes 13 and 10, which designate carriage return and line feed). Obviously, it is critical that the information be placed in a Tab Delimited file in the order in which the importing program expects it and also that the correct number of <tab> characters are included in each record in order to put the data into the correct fields in the imported file. It is also very critical that all carriage returns (CRs) except those designating the end of a record be removed from the data.

These instructions use both formats, but for different purposes.

The carriage return (CR) problem in Tab Delimited files

CRs are entered into a document (or EndNote record) when the <Enter> key is pressed. In most documents they are not visible; the only evidence of their presence is the start of a new line. Most software (such as EndNote and Word) also starts new lines when there isn't enough space on one line to contain all of the characters, but "wrapping", as this is called, doesn't involve CR characters. In Word, CRs can be made visible as follows: Go to Tools > Options > View. Under "Formatting Marks" make sure the

"Paragraph Marks" box is checked. You will then see the CRs represented as paragraph symbols in your document.

CRs present a problem when exporting references to a Tab Delimited format because when Access encounters a CR, it starts a new record. This is supposed to happen if there is only one CR and it is at the end of the record, but major problems arise if a CR exists somewhere before the end of the record. Any EndNote fields that contain CRs will cause Access to start a new record each time it finds one, and the remaining fields in that record will be placed at the beginning of the next Access record. You get a new Access record for every CR. Obviously, this would result in a completely unusable Access database.

Preparing the EndNote library for export

How to get rid of unwanted CRs. Since EndNote does not automatically strip CRs when it exports to a Tab Delimited file, you have to take steps to get rid of them. CRs are most common in the Author, Editor, Notes and Keywords fields, but they can exist in any field. Although the Author and Editor fields do contain CRs (each author/editor is on a new line), these CRs are automatically stripped by EndNote during export and do not present a problem. The problem arises with CRs in all the other fields.

The easiest method I have found to get rid of unwanted CRs from an EndNote library is to export the library using the "tagged" EndNote Export style, and then re-import the resulting file into a new EndNote library. The new library is a duplicate of the old library with some exceptions:

- All the unwanted CRs have been stripped out (Hurray!).
- All the references are automatically assigned new numbers in the order in which they are added to the new library.
- Any formatting that cannot be preserved in a text file (*e.g.*, superscripting, etc.) will be lost.

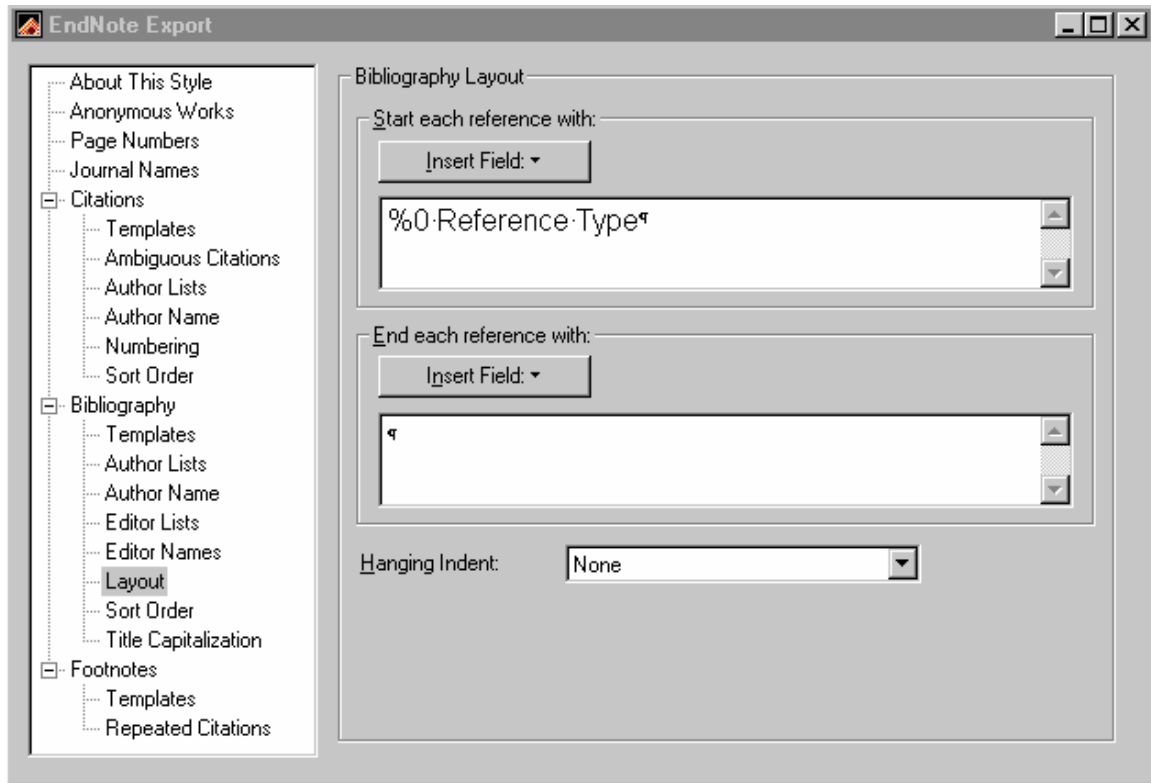
Having new record numbers has consequences. For example, the new library could not be used to format documents where the temporary citations were entered from the old library because the record numbers would no longer match, but presumably this would not be an issue since your purpose is to export the data to Access.

Losing formatting such as superscripting should also not be a problem for the same reason; Access does not support text formatting in its tables so far as I am aware.

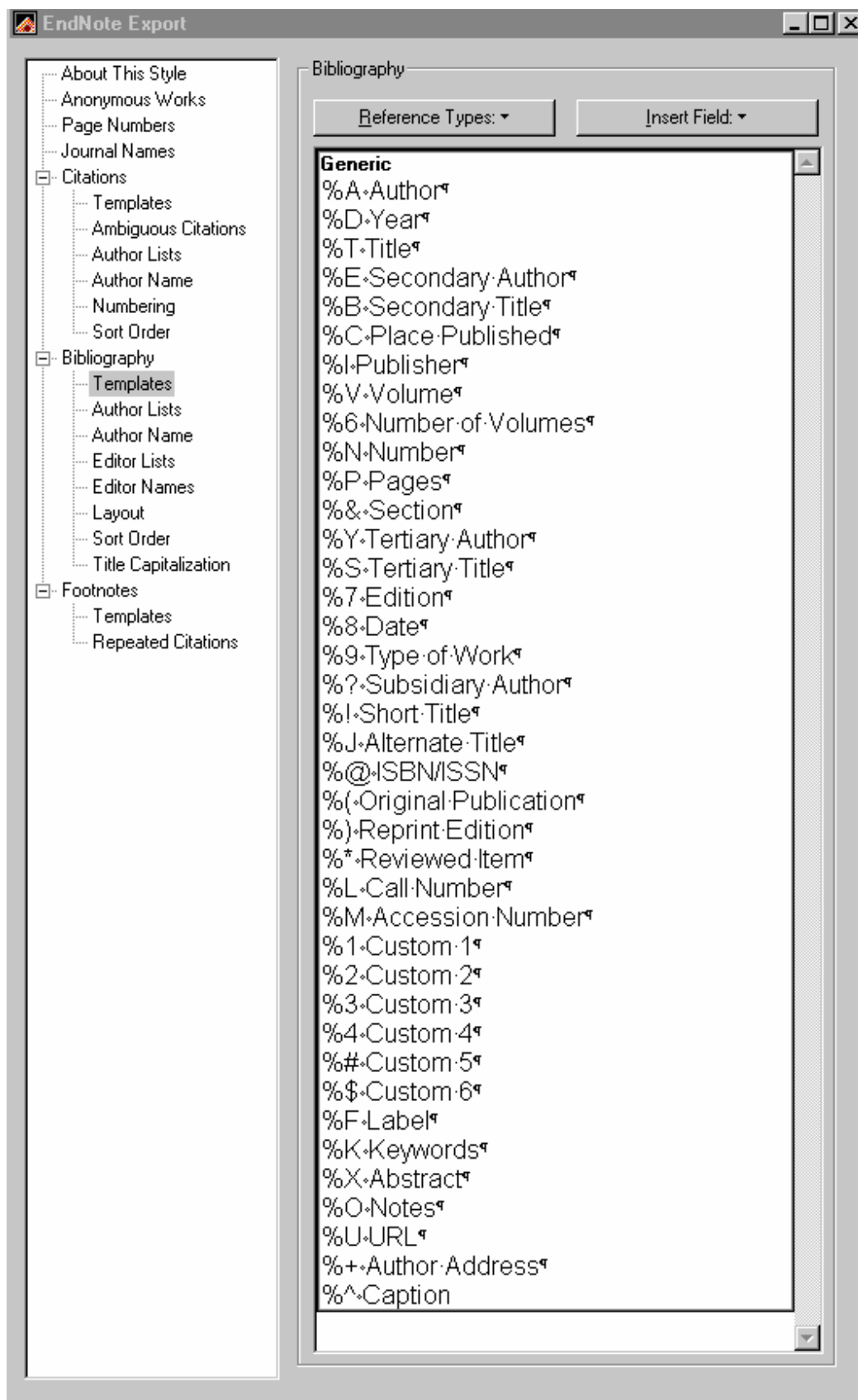
If you want to maintain a record of the record numbers from the old library in Access, see the section, "How (and why) to copy record numbers to another field" at the end of this document. The source code included with this document (Appendix B) can help you reconstruct your original library, including the record numbering, by inserting "dummy" records to fill in the gaps in numbering created by deleting records from your original library. This method is described below.

Creating a CR-free EndNote Library

1. Edit your "EndNote Export.ens" (Edit > Output Styles > Open Style Manager...; then select "EndNote Export" and click the "Edit" button). Make sure that the "Reference Type" field will be placed at the beginning of each record:



2. Next, make sure all the "generic" fields are listed in the Templates dialog box:



Note that this list depends on the version of EndNote you are running.

3. Save any changes to the style, return to your old library, show the records you wish to export in the main library window. If you wish to export a subset of records in your library, select those records and show only the selected references (References > Show Selected References). Export (File > Export) to a text file ("Old Library.txt"); make sure you remember the folder where you saved the file.

4. Create a new library (File > New).

5. Import the tagged text file (File > Import> Import Option = "EndNote Import"). Click the "Choose File" button and navigate to the folder into which you saved the tagged "Old Library.txt" file.

If everything worked properly, you should now have a copy of your EndNote library that is devoid of CRs. You can now export this library to a Tab Delimited File that Access can import.

Creating a Tab Delimited File in EndNote for Export to Access

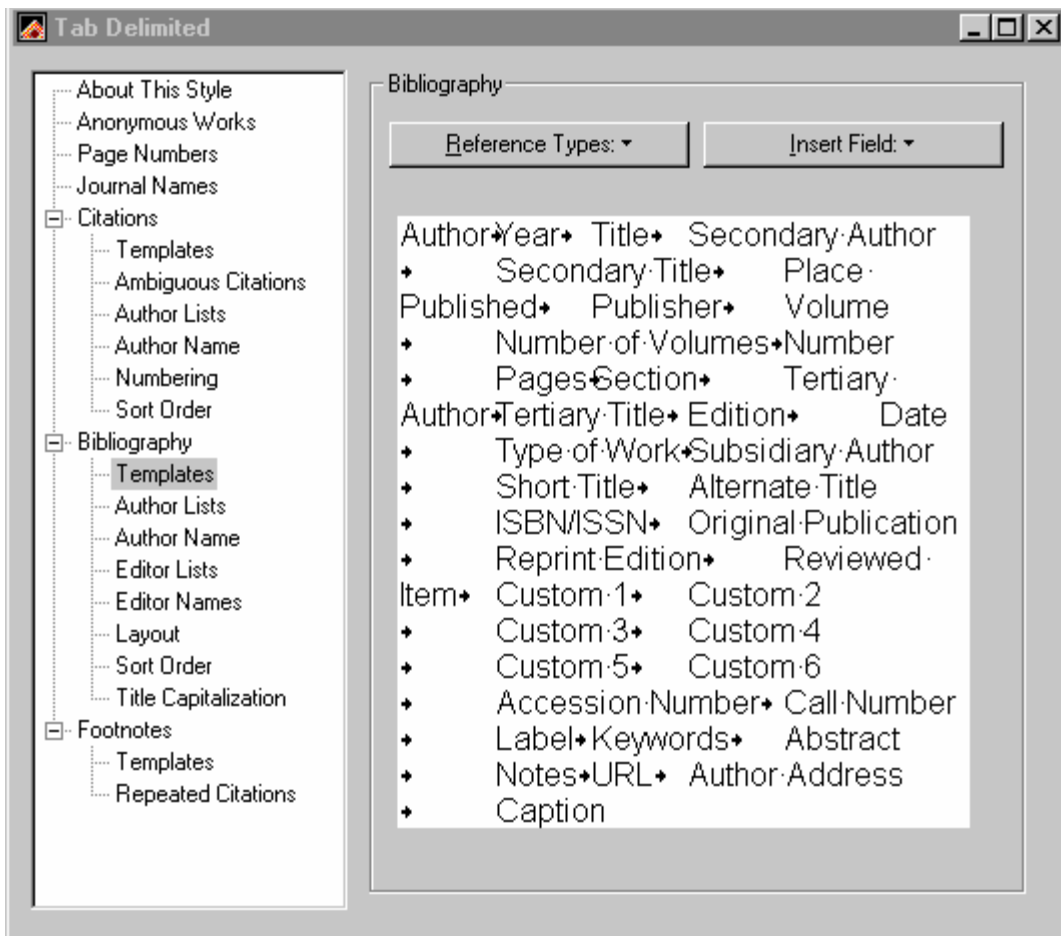
Select "Tab Delimited.ens" as the output style. Edit the style to make sure it matches the following:

Edit the Bibliography Template to include all generic fields except Image, which can't be exported. If you are not running version 6, you will have to leave out the fields that are not supported by your version (see Table __). To insert a <tab> character, click "Insert Field" and then click "Tab".

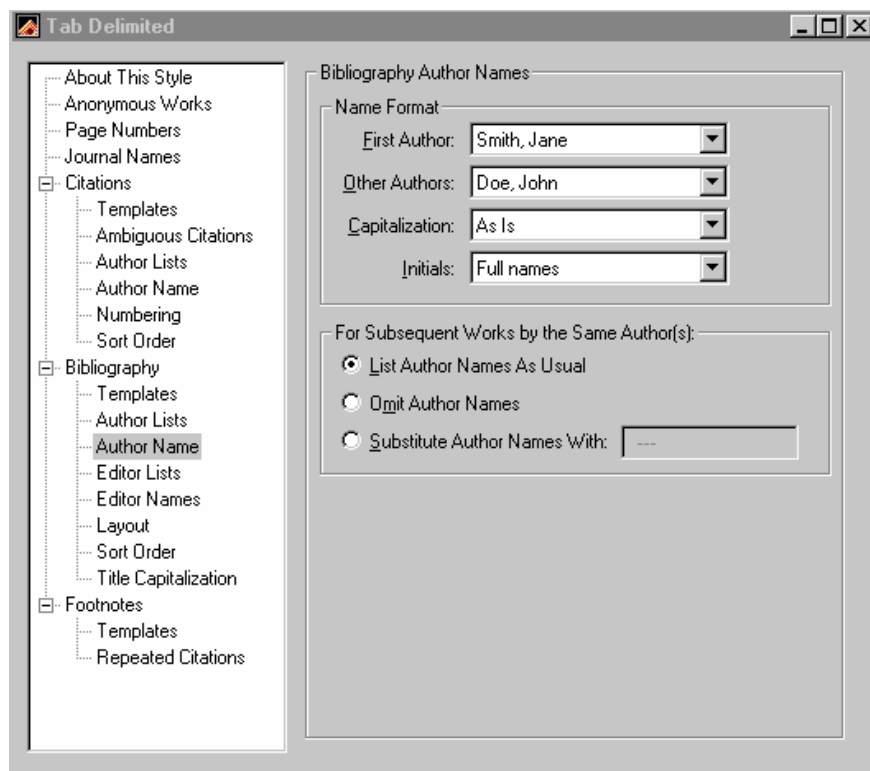
THE ORDER OF THE FIELDS IN THIS LIST IS CRITICAL

Author <tab> Year <tab> Title <tab> Secondary Author <tab> Secondary Title <tab> Place Published <tab> Publisher <tab> Volume <tab> Number of Volumes <tab> Number <tab> Pages <tab> Section <tab> Tertiary Author <tab> Tertiary Title <tab> Edition <tab> Date <tab> Type of Work <tab> Subsidiary Author <tab> Short Title <tab> Alternate Title <tab> ISBN/ISSN <tab> Original Publication <tab> Reprint Edition <tab> Reviewed Item <tab> Custom 1 <tab> Custom 2 <tab> Custom 3 <tab> Custom 4 <tab> Custom 5 <tab> Custom 6 <tab> Accession Number <tab> Call Number <tab> Label <tab> Keywords <tab> Abstract <tab> Notes <tab> URL <tab> Author Address <tab> Caption

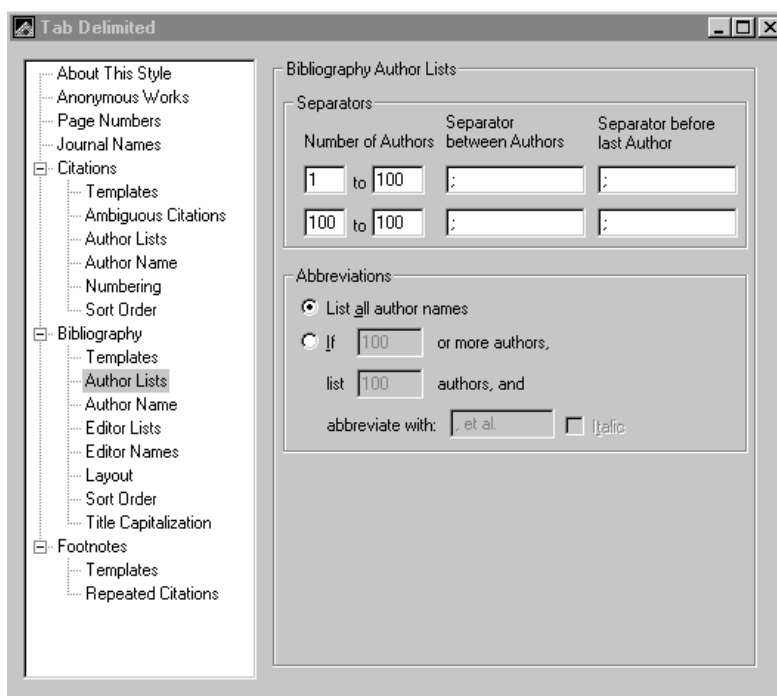
Reminder: include only the fields supported by your version of EndNote.



Next, go to the Author Name dialog and set it up as shown: last name first, comma, first name. Do the same in the Editor Names dialog.

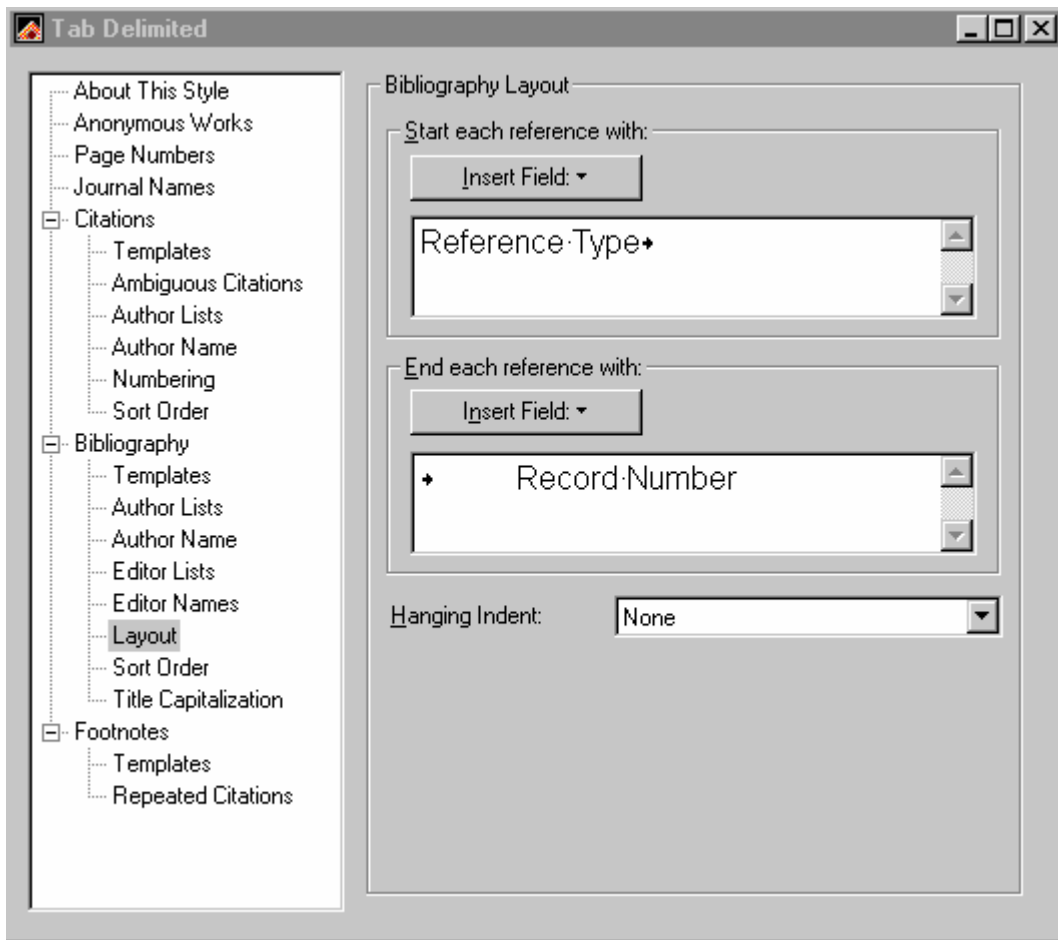


Also, make sure both the Author and Editor Lists are set up as shown:



The point of this is to have the authors and editors exported as follows: “Last, First M.; Last, First M; Last, First M...” This is the way the names will appear in Access and it is also a format that is easily imported back into EndNote through a Tab Delimited file.

In the Layout dialog, make sure the "Start each reference with:" box has "Reference Type <tab>" entered. The <tab> character is represented as a small right-pointing arrow. If you want to include the record number, enter it in the "End each reference with:" box as shown. Note that in this case the <tab> character precedes the field. Exporting the record number is a matter of choice; remember, however, that this is the record number of the references in the New library, not the old one. **Note:** Although it is possible to place the record number in the upper box, DO NOT do this. The order of the fields will be inconsistent and will prevent successful importing and exporting.



4. Select the Tab Delimited output style (Edit > Output Styles), then export (File > Export) the file. You should now be able to import this newly created file into either Access or Excel without misaligning the data.

Setting Up Access to accept the imported file.

Although Access can create a table at the time you import your Tab Delimited file, you should create the Access table first. This is because you could loose data if the fields in Access are too small and also because the code in the basEndNote module depends on the existence of the fields in the exact order and named exactly as shown below.

Note on text fields in Access. Access has two field types for text data (as opposed to numeric data). The "Text" field type can be set to a maximum of 255 characters. The "Memo" field type can handle any amount of text. Since many EndNote fields contain many more than 255 characters, the "Memo" field type must be used for those fields. The disadvantage of defining a field as "Memo" instead of "Text" is that it cannot be indexed in Access. On the other hand, you retain all of your data. In my own EndNote library, I have a little more than 2000 records. Although it is not common, I found that there can be more than 255 characters in the following fields: Author, Title, Short Title, Keywords, Abstract, Notes, URL, and Author Address. I don't have Caption data in my library, but I expect some captions could exceed 255 characters as well, hence its designation as a "memo" field. Here is the Access table design window for a table called "References" that works for my data. Note which fields are designated "Memo".

Field Name	Data Type	Description
Reference Type	Text	
Author	Memo	
Year	Text	
Title	Memo	
Secondary Author	Text	
Secondary Title	Text	
Place Published	Text	
Publisher	Text	
Volume	Text	
Number of Volumes	Text	
Number	Text	
Pages	Text	
Section	Text	Omit for EndNote 3
Tertiary Author	Text	
Tertiary Title	Text	
Edition	Text	
Date	Text	
Type of Work	Text	
Subsidiary Author	Text	
Short Title	Memo	Omit for EndNote 3
Alternate Title	Text	
ISBN/ISSN	Text	
Original Publication	Text	Omit for EndNote 3
Reprint Edition	Text	Omit for EndNote 3
Reviewed Item	Text	Omit for EndNote 3
Custom 1	Text	
Custom 2	Text	
Custom 3	Text	
Custom 4	Text	
Custom 5	Text	Omit for EndNote 3
Custom 6	Text	Omit for EndNote 3
Accession Number	Text	
Call Number	Text	
Label	Text	
Keywords	Memo	
Abstract	Memo	
Notes	Memo	
URL	Memo	
Author Address	Memo	Omit for EndNote 3
Caption	Memo	Omit for EndNote versions 3, 4 and 5
Record Number	Number	
ID	AutoNumber	Filled by Access; not imported

Notes:

The field names are exactly the same as in EndNote (except the last one—see below).

The order of the fields is exactly as in the EndNote Tab Delimited Export style.

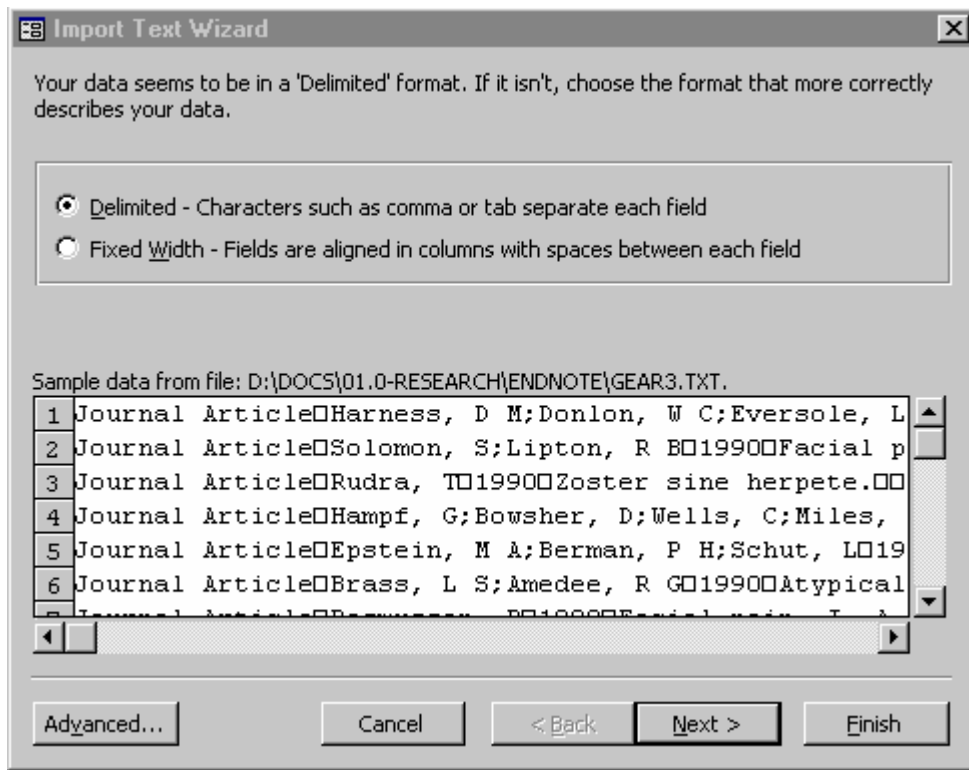
All fields except the last two are text designated as either "Text" or "Memo". This is because EndNote is a text-based program. Numbers are not needed except as text.

The "ID" field at the bottom works similarly to the record number field in EndNote in that it provides an automatically assigned unique number for each Access record. It should be designated as the "primary key" by placing the cursor in the ID filed row and clicking the "key" icon at the top of the window.

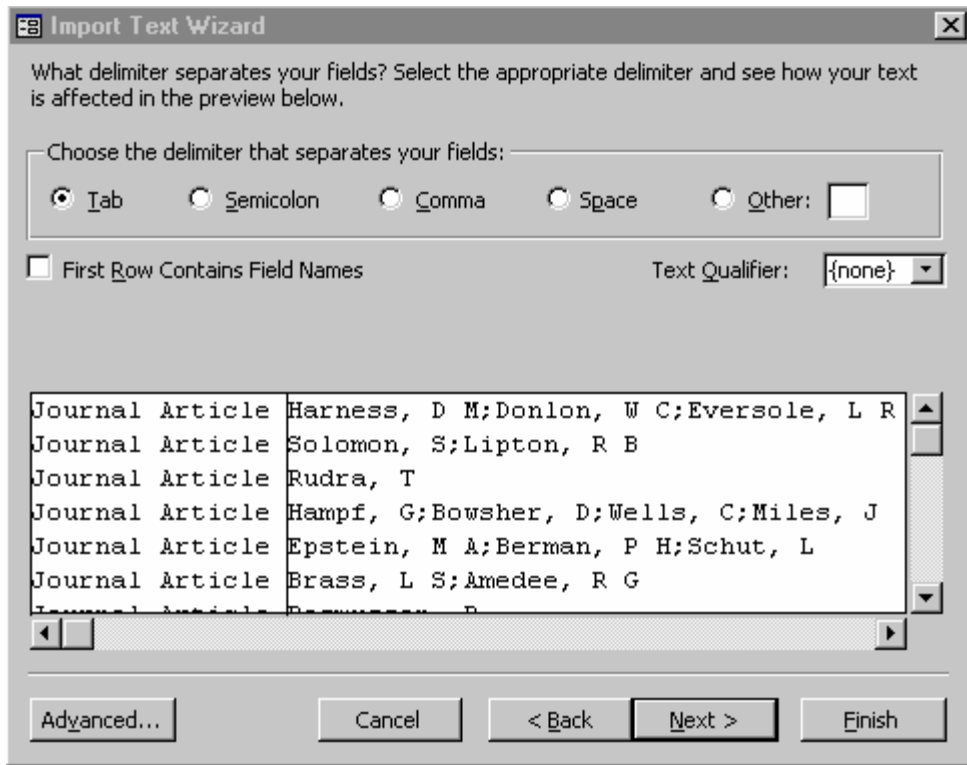
Important: Make sure the "ID" field is the last field. Access won't import the Tab Delimited file correctly otherwise.

Importing the Tab Delimited file into Access

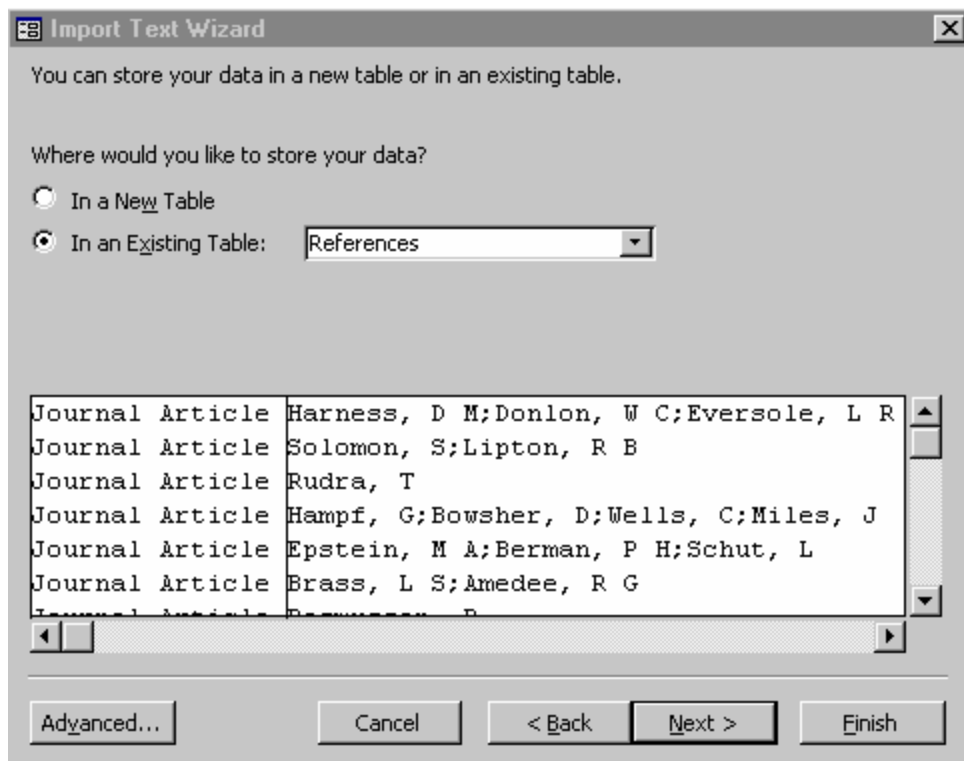
1. Open the Import Data wizard (File > Get External Data > Import. Navigate to your Tab Delimited text file, click "Import"). Here is what you will see:



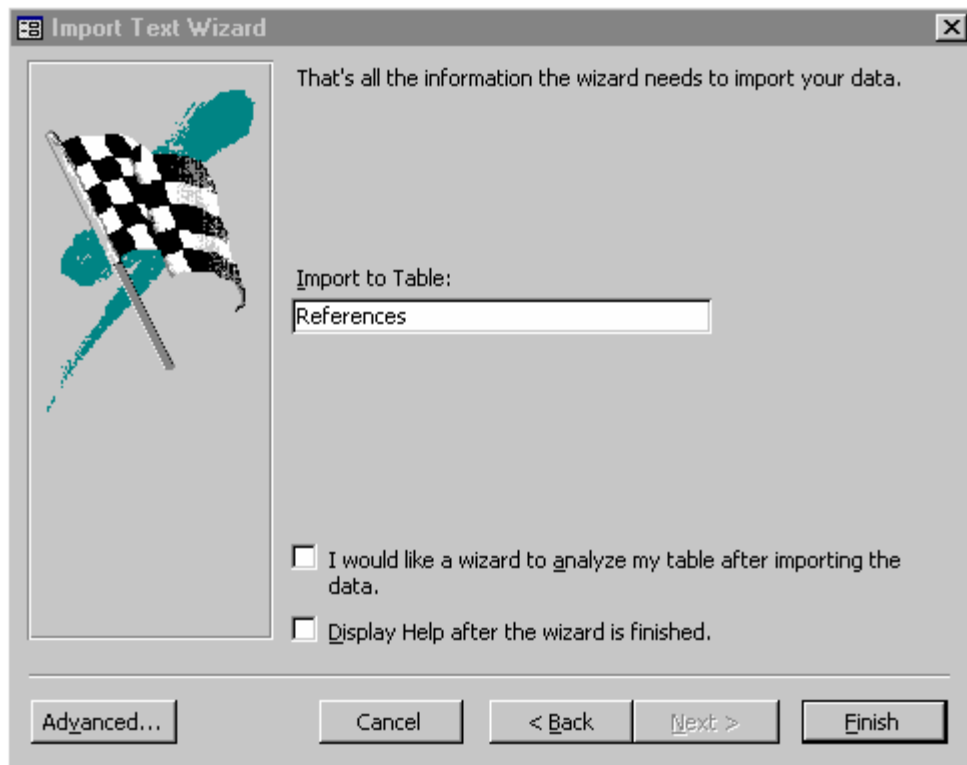
2. Click the "Next >" button. You then get a box that allows you select the type of delimiter and provide some other information that Access needs.



Make sure the "Tab" option is selected. The "Text Qualifier:" should say "{none}". The "First Row Contains Field Names" should be left unchecked. Click "Next>".



Select "In an Existing Table:" and then select the table just set up above, in this case named "References". Click "Next>".



Click "Finish".

If everything goes as it should, you will get a notice that the data was imported successfully. If Access has trouble, it will either import nothing or it will automatically create a table that lists the errors it encountered during the import. Following the steps outlined above, however, resulted in no errors in my file of more than 2000 records.

Preparing to Export a Tab Delimited File from Access

EndNote can import a Tab Delimited File. The description of how to do this in EndNote 6 is found on page 155 of the manual. There are two ways to set up the file. I will describe the more general way that allows you to tell EndNote what the Reference Type is for each reference. Here are the rules:

The first line of the Tab Delimited file should consist of the following line:

*Generic<CR>

where <CR> is the carriage return character.

The second line should consist of all the EndNote generic fields in the *exact* order and spelling as in the EndNote Tab Delimited Export style as well as in the Access table. Do NOT, however, include the record number field because EndNote will automatically provide new Record Numbers as it imports your references into the library. Also, of course, don't include the Access ID field or any other fields you may have created for the table.

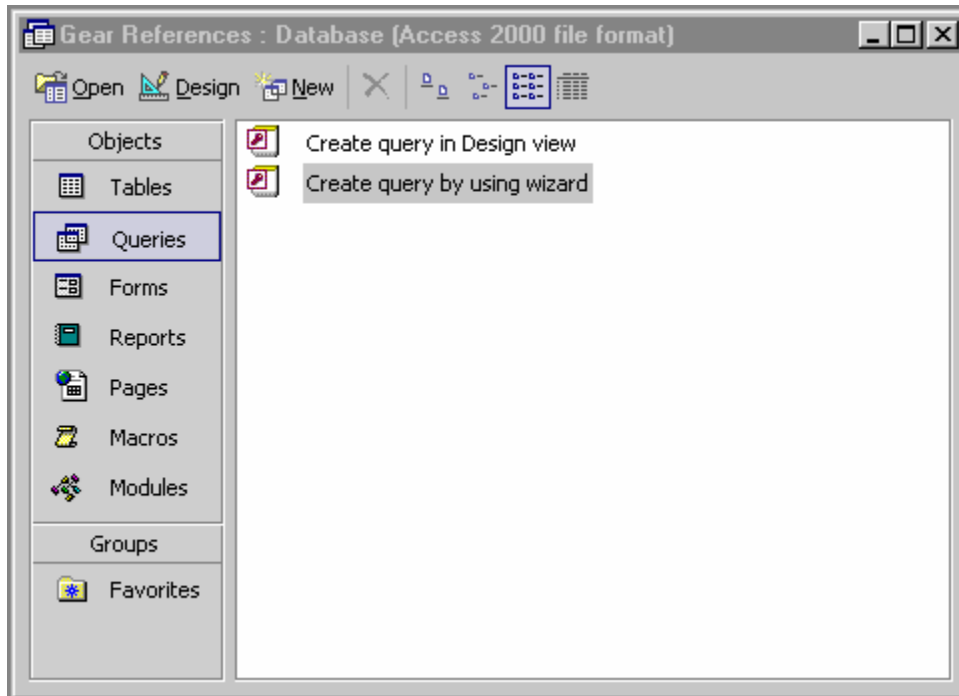
The rest of the lines consist of the data for each field separated by <tab> characters between the fields and CRs at the end of each record.

Manual method of exporting a Tab Delimited File from Access

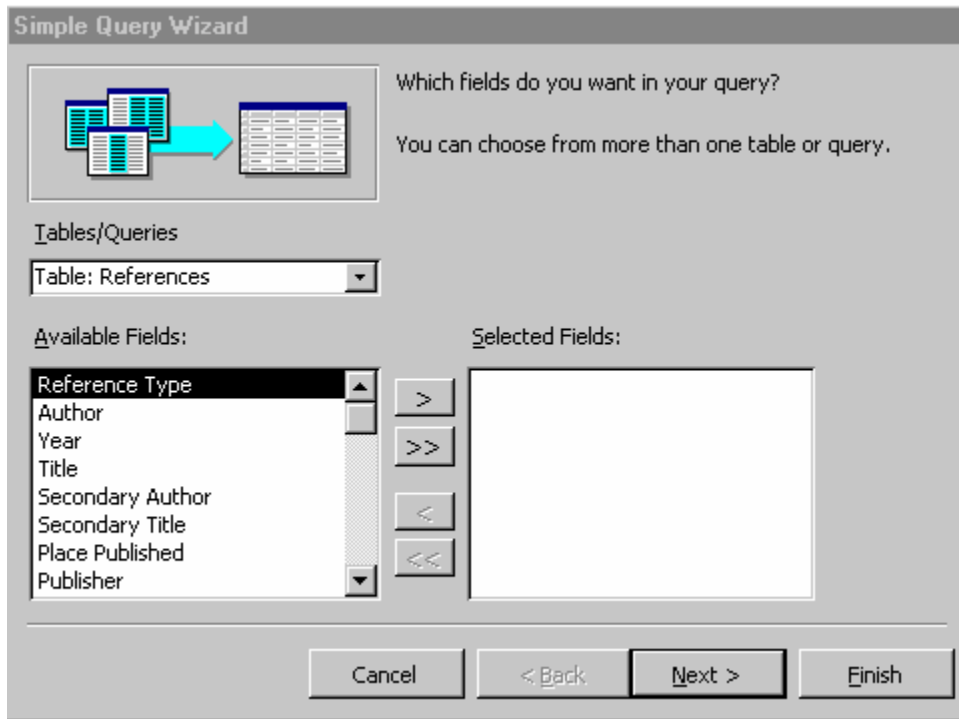
Note: You should be aware that the "CR Problem" is just as important when importing a Tab Delimited File into EndNote as into Access. Any CRs included in fields prior to the end of the record will result in EndNote placing the remaining fields in a new record, causing major data misalignment. Therefore, if CRs exist in any field in any record to be exported by Access to EndNote, they must be removed prior to exportation.

Although you can tell Access to export data directly from your References table, you cannot choose which fields to export (or not to export) by this method. You will get all of the fields in the table. For the reasons given above, this won't work for EndNote. So, you must create a query to do it. It is easier than it sounds.

Select the "Queries" button in the Access database window:



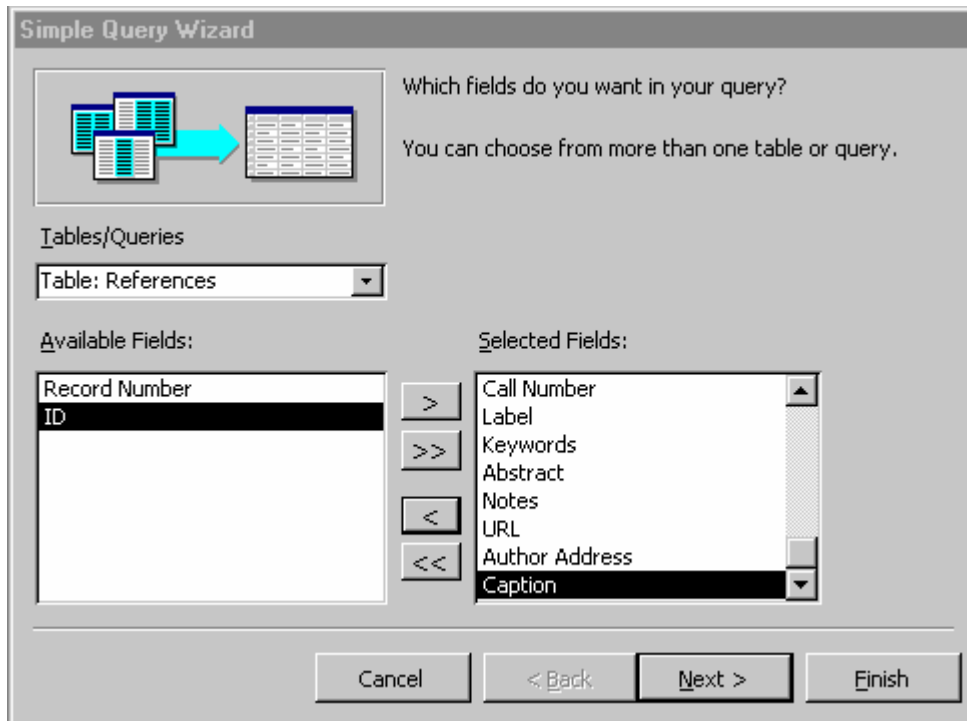
Select the "Create query by using wizard" option:



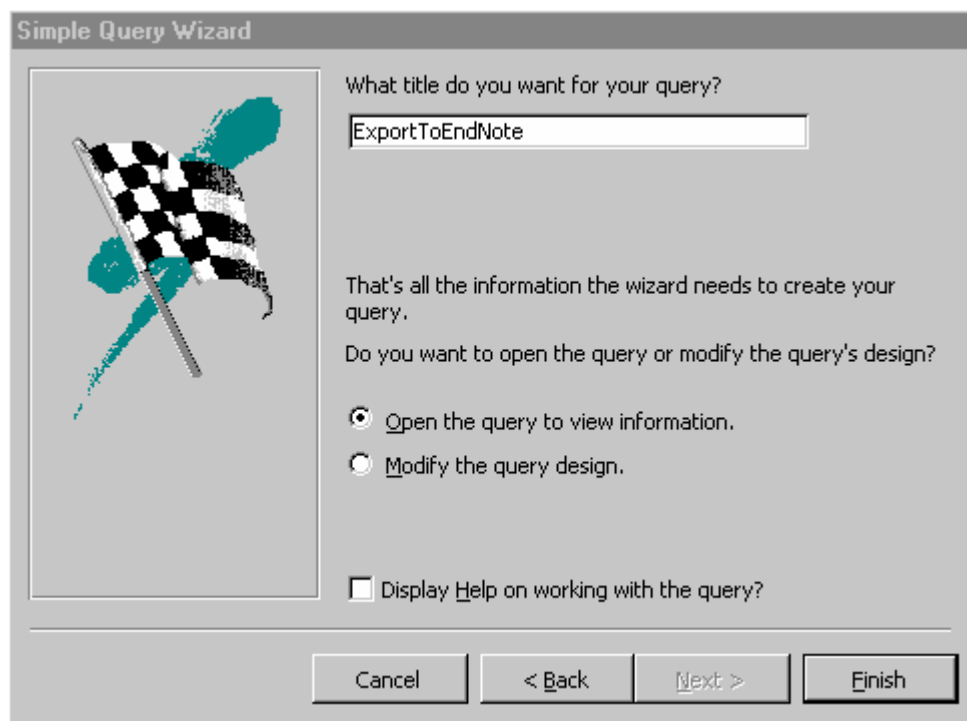
If it isn't already showing in the "Table/Queries" box, select the references table, in this case called "References". Then click on the ">>" button, which will transfer all the "Available Fields" over to the "Selected Fields" box. The fields will be in the same order as they are in the table itself. You must then scroll to the bottom of the list on the right

and find the two unwanted fields, Record Number and ID, select them and click them back over to the "Available Fields" box with the "<" button.

Here is how it should look when you are finished:



Click the "Next >" button.



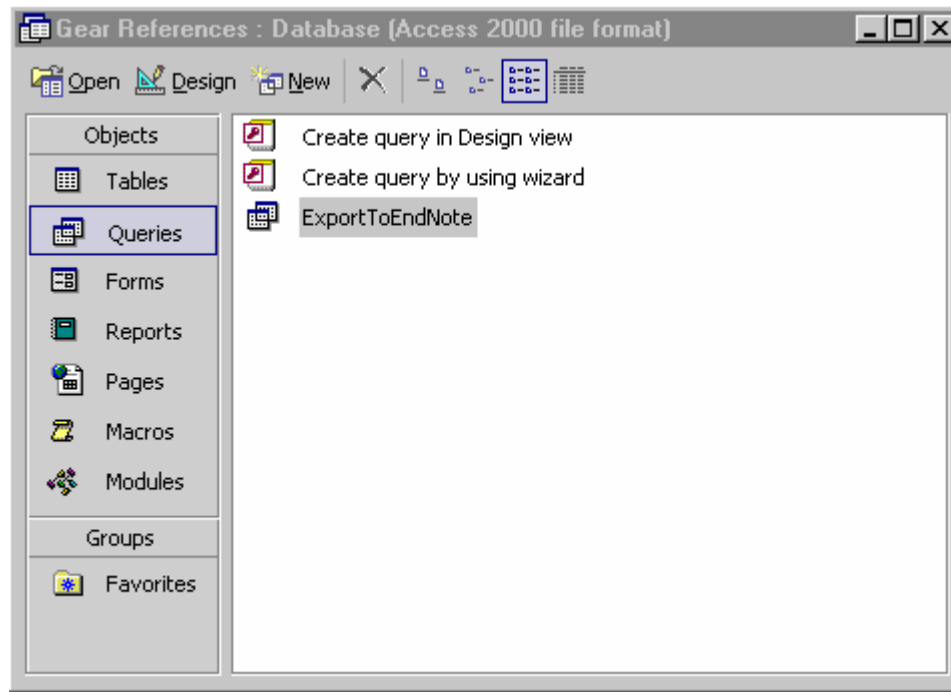
Give the new Query a name such as "ExportToEndNote". Click "Finish". Access will then run your query and show you the result:

Reference Type	Author	Year	Title	Second
Journal Article	Harness, D M;Donlon, W C;E	1990	Comparison of clinical	
Journal Article	Solomon, S;Lipton, R B	1990	Facial pain.	
Journal Article	Rudra, T	1990	Zoster sine herpete.	
Journal Article	Hampf, G;Bowsher, D;Wells,	1990	Sensory and autonom	
Journal Article	Epstein, M A;Berman, P H;S	1990	Cavernous angioma pi	
Journal Article	Brass, L S;Amedee, R G	1990	Atypical facial pain.	
Journal Article	Rasmussen, P	1990	Facial pain. I. A prosp	
Journal Article	Rasmussen, P	1990	Facial pain. II. A prosp	

You can scroll either across to see the rest of the fields or down to see the records. If you scroll across, you will see all the fields except the two not included in the query.

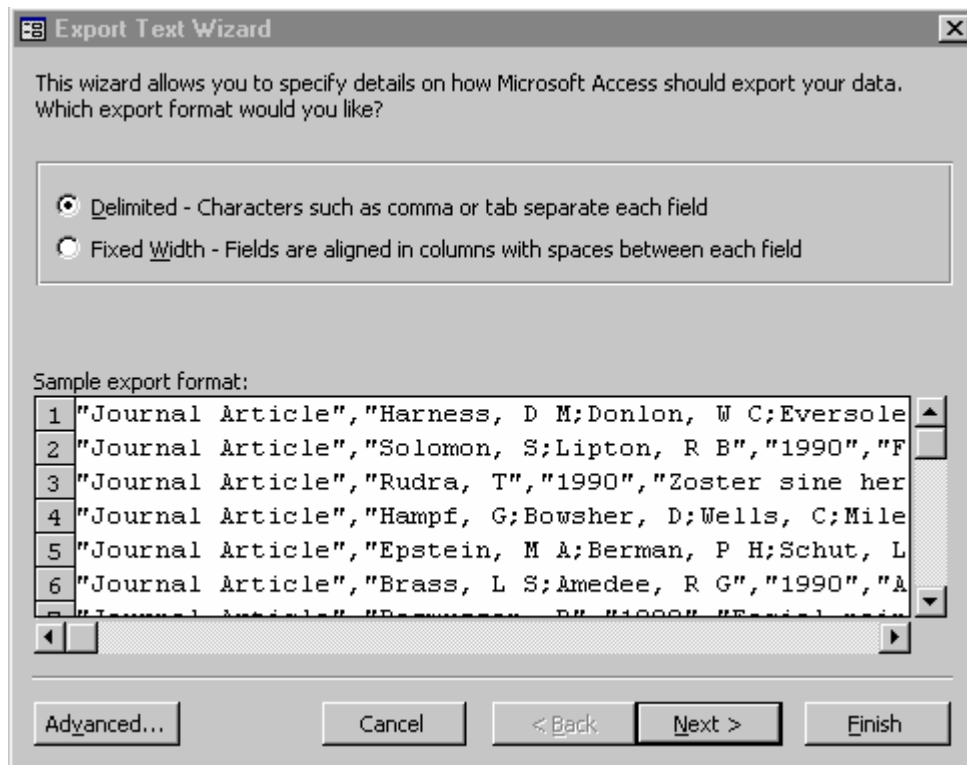
Notice at the bottom of the window, Access lists 2061 records, which is the number in my library. If you want to export a subset of your records, you will have to use the "Where" clause in your query, but this is beyond the scope of the present discussion.

Close the query window and you will see the new query listed in the database window:

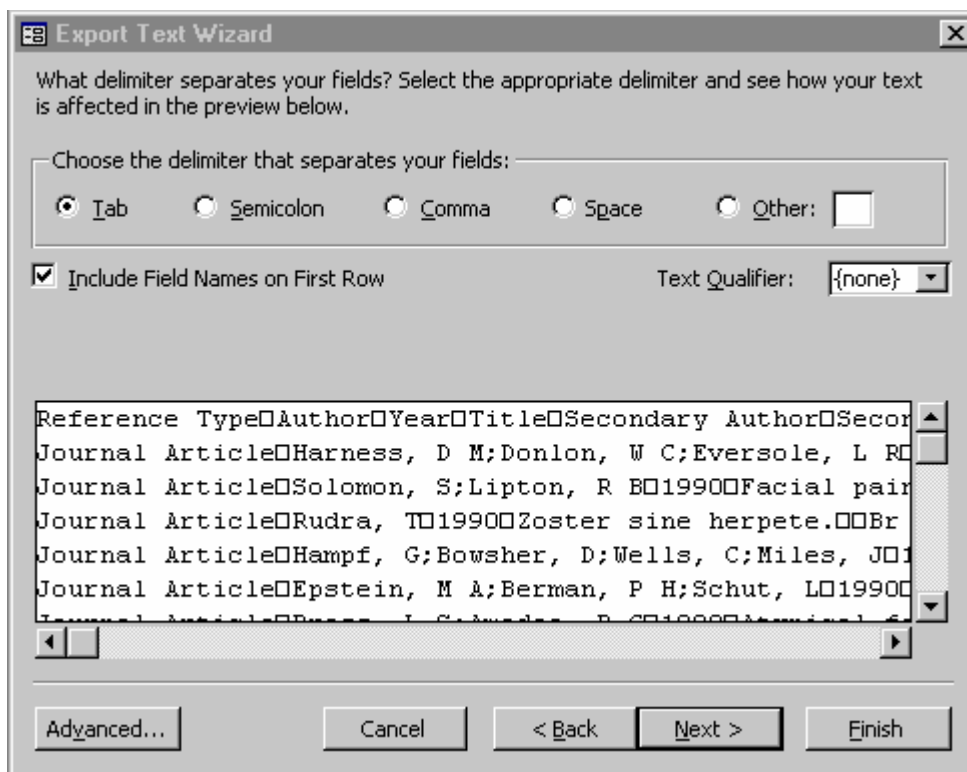


Exporting a Tab Delimited File using the "ExportToEndnote" query.

In the Queries section of the database window (above) select the "ExportToEndnote" query and start the export process (File > Export). Navigate your way to the EndNote folder. In the "Save as type:" box, select "Text Files (*.txt; *.csv; *.tab; *.asc)". Change the name of the default file name if you wish, then click the "Export" button to start the Export Text Wizard:

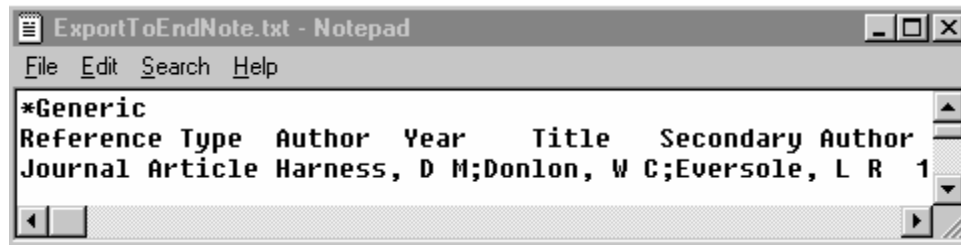


Make sure the "Delimited" option is selected, then click "Next >".



In this dialog, three things are important. Make sure "Tab" is the selected delimiter (top row), change the "Text Qualifier:" option to "{none}", and make sure the box for "Include Field Names on the First Row" is checked. This is important because EndNote expects these field names in its import file. You can then click the "Finish" button, and the file will be exported the designated folder.

There is still one thing left to do to prepare the file for importing into EndNote. Open the newly exported file in a text word processor such as Notepad. Place the cursor in front of the very first character in the file and type "*Generic" and then press the <Enter> key once:



Note that the <tab> characters are not visible in Notepad, but they are there just the same. Save the file. It is now ready to import into EndNote.

Automatic method of exporting a Tab Delimited File from Access

The code for automating this process (Appendix B) was written with two purposes in mind. First, to remove all CRs from the Access table created above. Second, to automate the export process. I deliberately wrote the code as an Access module, rather than as a “class module” that is associated with a particular Access form. It will function without a form; but to start it, you will need to create a one line macro using the “RunCode” action. You could also run it by creating a command button in a form. The name of the function to run is called “ExportReferencesToEndNote()”.

As written, the procedure finds and removes all occurrences of the two ANSI codes (13 and 10) that together make up a CR and puts a space character in the same places. The procedure that does this is called “ReplaceStr”. This is actually a general purpose function that could replace any string of characters with any other string of characters in a given string of text that you want to change. These changes are made to the References Table itself, not only to the exported file.

Once the Table is free of CRs, the procedure goes on to create a Tab Delimited text file that is formatted for importation into EndNote with no further editing (as is required for the “Manual Method” described above).

This procedure will work only with the Table described above (containing only the fields supported by your version of EndNote). Therefore, in order to use it as is, you will need to create the “References” table as shown above and put your bibliographic records into it. If you already have an Access database that either did or didn’t come from EndNote, you should still create the table. Then use Access queries to fill the References table in the correct format. If the correct data is in the References table, it will be imported into the correct fields in EndNote after being exported from Access.

There are a number of steps required to get this procedure up and running:

Create a table called “References” as described above.

Import your references into the table in the correct format. This is not necessarily trivial. Be aware of the following:

Authors and Editors should be formatted as “Lastname1, F M; Lastname2, A B”...etc. That is Last name first followed by a comma followed by initials or first name(s). Note that individuals are separated by semicolons.

Be sure to put the data into the correct fields. You should be familiar with the section of the EndNote manual that describes which information goes into which fields. Along these lines, it should be noted that the “Generic” reference type in EndNote contains all fields and in actuality defines the field names for all reference types. All other reference types are based on the “Generic” reference type. So, for import and export purposes, the “Generic” reference type is all you need to know about. To import

references into the correct reference types in EndNote, the first field in each record in the Tab Delimited file contains the name of the reference type destination. In the Access table, "References", this field is called "Reference Type".

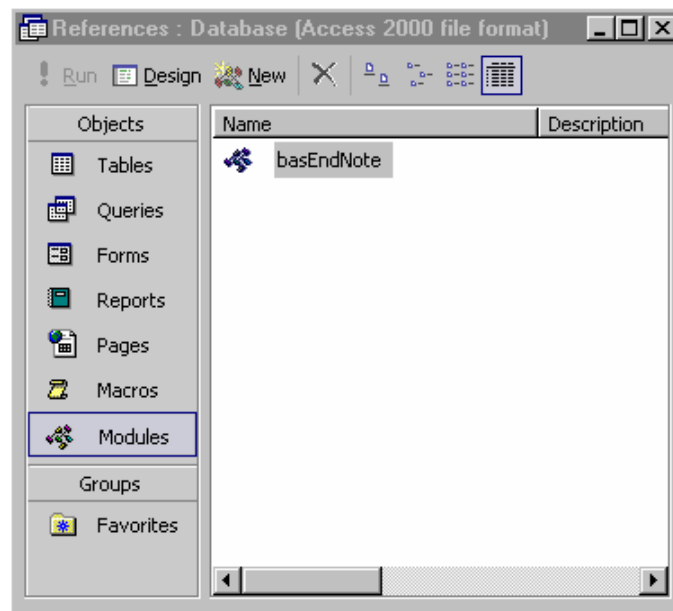
Create a new module in the Modules section of your Access data base. Click Modules > New. Copy and paste the contents of "basEndNote.txt", the text file accompanying this pdf file, into the module window that appears. The source code can also be found in Appendix B. I suggest you name your new module "basEndNote". When pasting the code into the new module window, make sure the top line "Option Compare Database" appears only once; an error will occur otherwise.

You will need to enter the correct path and the correct file name for the Tab Delimited file you intend to export. This information must be typed into the code itself. Near the top of the source code, locate these lines:

```
strPath = "C:\Program Files\Endnote\  
strFileName = "ImportToEndNote.txt"
```

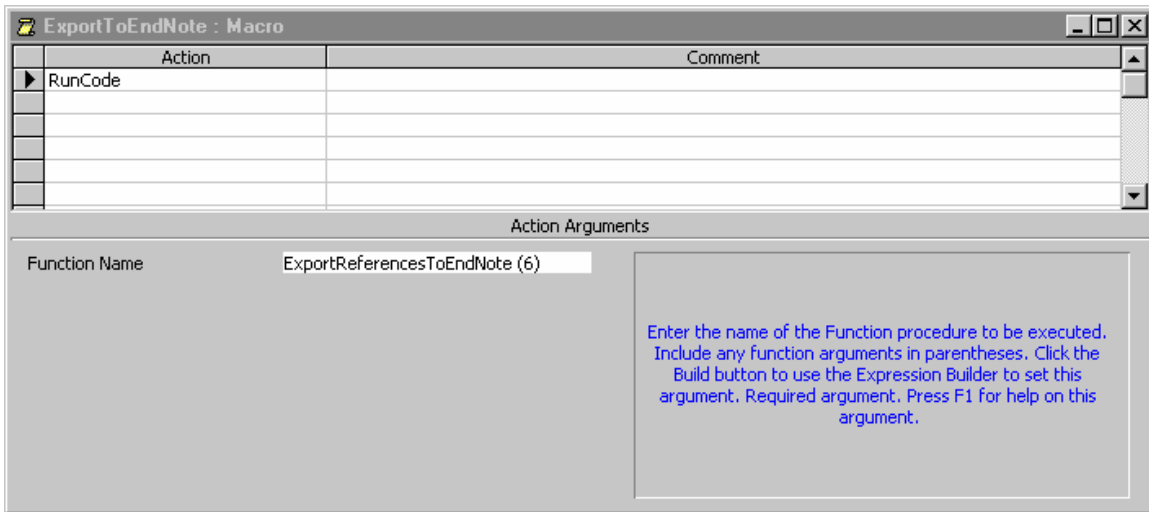
In the "strPath" line, change the text inside the quotes to the correct path for your EndNote directory. The "strFileName" line contains the name of the exported Tab Delimited file that you will import into EndNote. You can keep the name as it is or change it. Note that each time you create a new export file, you will overwrite the old file without warning. If you want to save an old exported file, either change the name in the code or change the old file name.

Save the module and name it "basEndNote".



If you have structured the References table as shown above, such that it contains the fields supported by your version of Endnote, the code in basEndNote should be able to automatically export a Tab Delimited file that EndNote can import.

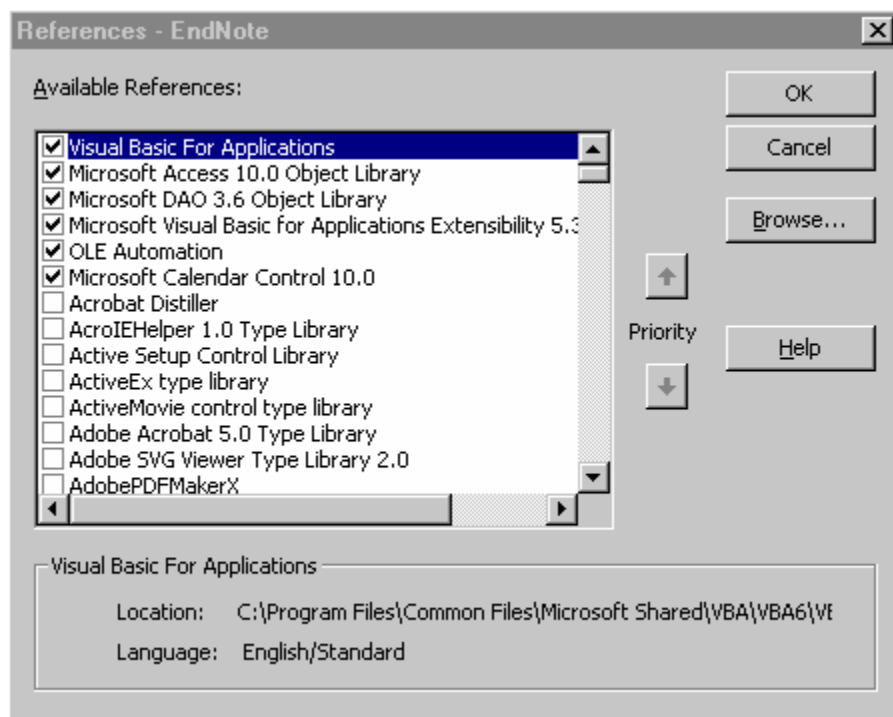
Create a macro to run the code. In the database window (above) click on "Macros", then click "New". Type "RunCode" on the first line in the upper box. In the Function Name box below type "ExportReferencesToEndNote(#)", where # stands for your EndNote version number (version number are limited to 3,4,5, or 6).



Run this macro when you want to export your Access records to Endnote using a Tab Delimited file.

Troubleshooting the module.

Although there are many possible sources of problems, most of which can be solved by anyone a little familiar with Access programming, it is possible that a more subtle problem involving "references" can arise (these "references" are intrinsic to Access and are completely different from the References table—the common name is completely coincidental). If, when you attempt to run the code, Access acts as if it never heard of some of the commands in the code, the problem could be in "references." These references have to do with the code "libraries" (another coincidental term) that Access needs to run. Fortunately, if you know about it, it is relatively easy to check. Open the module basEndNote. Click on Tools > References. You will see the following dialog box:



Make sure the same boxes are checked and that they are in the same order, which you can change with the Priority buttons to the right side of the list. If you run a different

version of Access, your list may differ, but try to match what you have to this list as best you can. The list, by the way, is VERY long, so you may have to do some searching to find the boxes you need to check.

Using basEndnote

I have tried to make this code as automatic as possible. As written, it will accommodate EndNote versions 3 - 6. All you have to do is tell it which version you want by including the version number when you run it, for example:

```
ExportReferencesToEndNote(6).
```

The number in the parentheses is your version number. The code will automatically strip the CRs out of the Access References table and create a Tab Delimited file suitable for import into EndNote with no further editing needed.

Also, after starting the code, a small dialog box pops up that allows you to choose to "pad" the Tab Delimited file with "dummy" records to fill in the gaps in record numbering created by deleting references from your old library. To get this to work, you must have the References table structured exactly as shown, and have the "Record Number" field contain the record numbers from your old library. If this is the case, then the code can create dummy records in the output file that will fill in the number gaps. Then, after you delete the dummy records from your newly imported EndNote library, your library will be numbered just as it was originally.

This assumes, of course, that you will be importing into a new library. If you import into an existing library, the numbers will be off by the number of records already in the library.

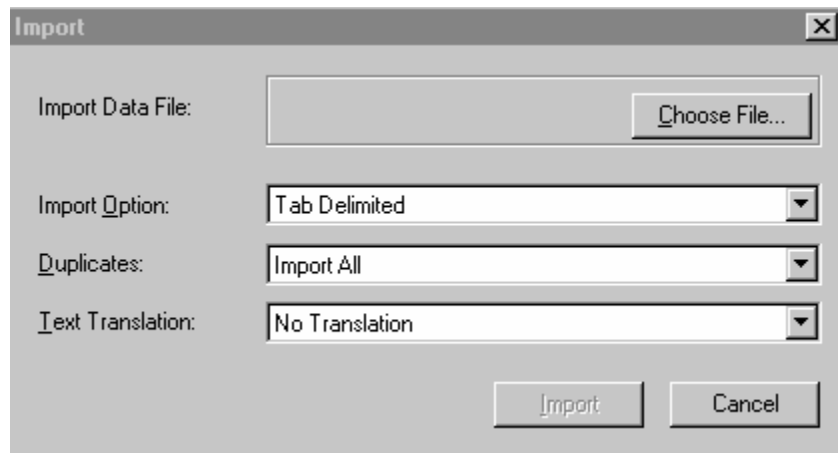
Note, however, that the code is written so as not to attempt padding with dummy records if the Record Number field of any of the References in the table doesn't contain a number. A pop-up dialog will warn if this is the case. Also, if there are duplicate numbers, you will probably get erroneous results but there will be no warning.

To get your old record numbers into the Record Number field, you can adapt the method shown below (Appendix A) or just follow the method in Appendix A as written and then use Access to fill the Record Number field from the Label field. Remember, though, that the Record Number field is a "number" data type and the "Label" field is a "text" data type, so you will have to convert the text to numbers.

Of course this technique of padding with dummy records is useless if your data did not come from EndNote to begin with. In that case, you needn't use it unless you want to specify record numbers in advance.

Importing the Tab Delimited File into EndNote

This final step couldn't be easier. Open the EndNote library into which you want to import the references. Open the Import dialog (File > Import):



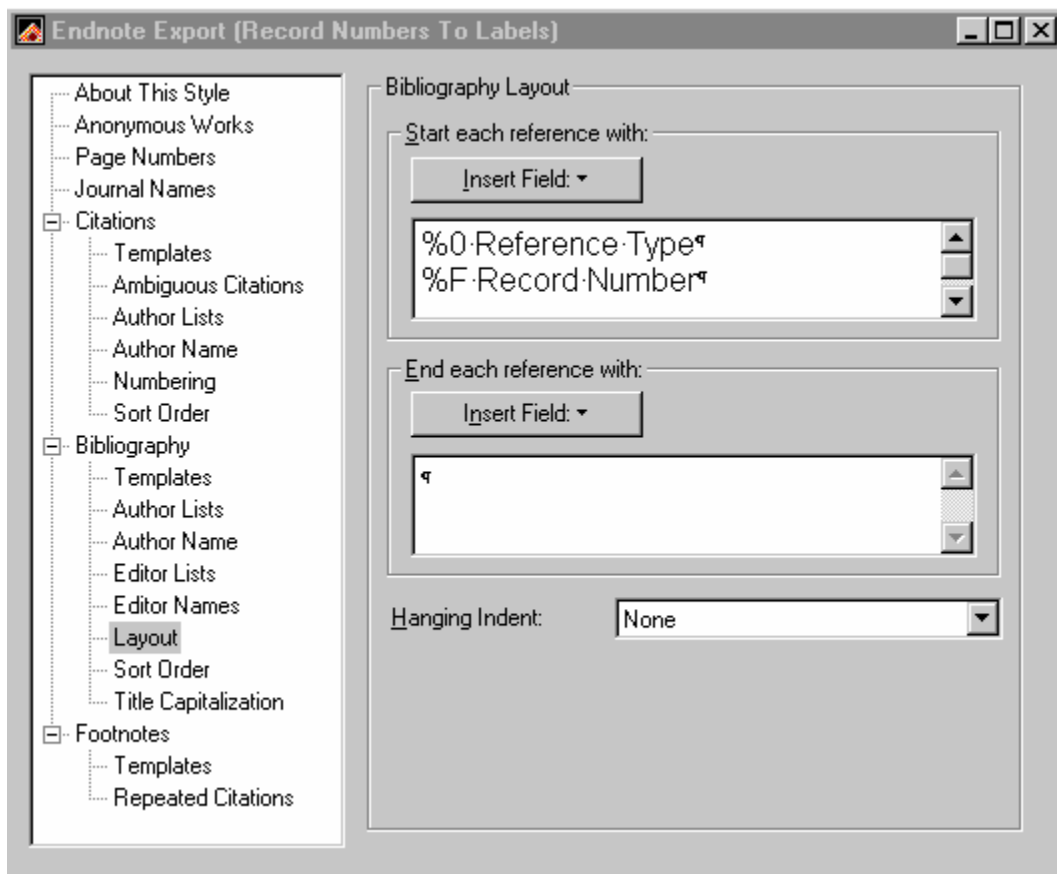
Select "Tab Delimited" in the "Import Option" box. Click on the "Choose File" button to select your new file. Click the file to select it, then click the "Open" button and the import process will begin.

APPENDIX A: How (and why) to copy record numbers to another field in EndNote.

There is no built-in way to copy record numbers to another field. What you have to do is to export your EndNote library to a text file using a modified version of the "EndNote Export.ens" tagged output style. By modifying the style you can export record numbers "tagged" with one of the recognized fields. Then re-import the file into a new library and the record numbers from the old library will now be found in the chosen field of the new library. While you can transfer the record numbers to any field, or to more than one field, this example uses the Label field. Here are the steps:

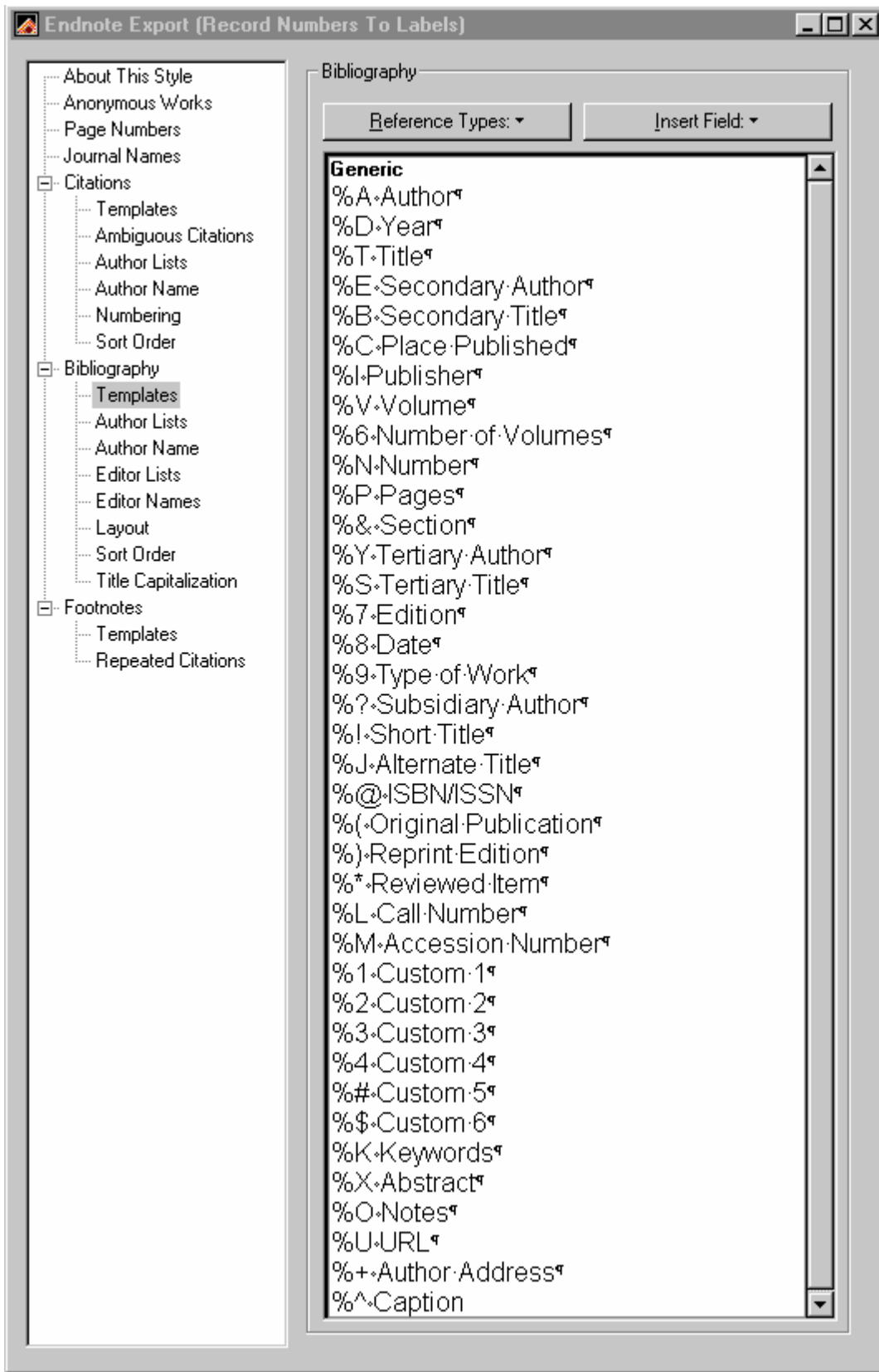
Make a copy of the EndNote Export.ens style and name it something like "EndNote Export (Record Numbers to Labels).ens". You definitely don't want to get this style mixed up with the original style. (See cautionary note below).

Edit the new export style. In the Bibliography Layout window, below the "%0 Reference Type" line, add a new line as shown below:



The %F tag will tag the Record Number for the Label field. You can also add a text marker to each Label. For example, if you want each number to be preceded with your initials (mine are RWG), the new line would become, "%F RWGRecord Number". If you use a different text prefix in different libraries, you can identify the originating library.

Now, go to the Bibliography > Templates window delete the line, "%F Label".



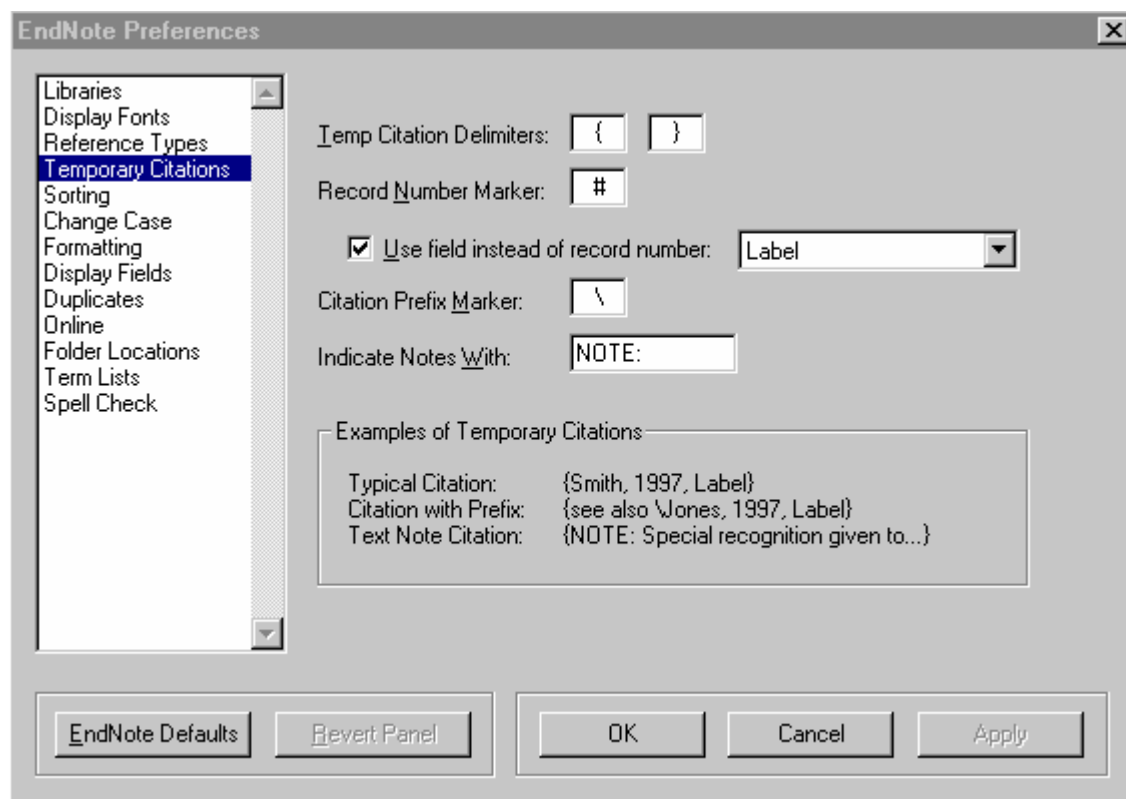
Save the changes to the style. Export your library as described above and then re-import it using the Endnote Import option, also described above.

The reason you might want to do this is to enable to you to use the Label field instead of record numbers in your temporary citations. This procedure will make your library backwards compatible with your documents. However, you will have to change the Temporary Citations in your documents to the format EndNote expects for Labels.

Change {Author, year #Record Number} to {Author, year, Label}.

Example: {Smith, 2002 #1502} becomes {Smith, 2002, 1502}.

With any documents in which Record Numbers were inserted into the temporary citations, you "Search and Replace" the " #" (space - #) in the documents with ", " (comma - space). Of course, if you added text to your Label fields, you need to include it here as well: Replace then becomes ", RWG" (using the example above). Then change your EndNote preferences (Edit > Preferences > Temporary Citations) to use Labels instead of record numbers when formatting your documents:



Using Labels instead of Record Numbers in your temporary citations offers the great advantage that you can always format your documents from any library containing the reference. You can transfer them in and out of Access or between EndNote libraries and still be able to format your documents. If a document was formatted in EndNote 6 using the Label field, you can also create a new EndNote library from the traveling library saved in document, and then reformat it with another style or add new citations if needed. You don't need the original library from which the document was formatted. You wouldn't be able to do that if you are using Record Numbers in temporary citations because Record Numbers are changed every time they are moved from library to library. The Label field will always stay the same unless you edit it.

Cautionary Note! If you create a third generation EndNote library by using the "EndNote Export (Record Numbers to Labels).ens export style, you will transfer the new record numbers to the Label field and lose all your old record numbers. Never use this style more than once for any given set of references.

APPENDIX B: basEndNote source code

Note that “remarks” are preceded by a single quote (apostrophe mark). Any text to the right of this mark is generally not part of the code. There are exceptions to this rule, however, like when the apostrophe is used as a single quote inside a double quote string.

```
Option Compare Database
```

```
Option Explicit
```

```
Function ExportReferencesToEndNote(ENVersion As Integer) As Integer
```

```
    On Error GoTo ErrorTrap
```

```
    Dim strPath As String
```

```
    Dim strFileName As String
```

```
    ' set your path name on the next line
```

```
    strPath = "C:\Program Files\Endnote\" 'Type your desired path using the shown format
```

```
    strFileName = "ImportToEndNote.txt" 'name of Tab Delimited export file
```

```
    Dim varTextFile, varFileName
```

```
    Dim db As Database
```

```
    Dim rst As Recordset
```

```
    Dim intXX As Integer
```

```
    Dim intChoice As Integer
```

```
    Dim strChoice As String
```

```
    Dim strWrite As String
```

```
    Dim ReferenceCount As Integer
```

```
    Dim strRst As String
```

```
    If Trim(strPath) = "" Then strPath = "C:\" 'Default path
```

```
    If Trim(strFileName) = "" Then strFileName = "Default.txt" 'Default file name...
```

```
    strFileName = Trim$(strPath) & Trim$(strFileName) 'combine the two names
```

```
DoCmd.Hourglass True
```

```
'OK, before going any farther, let's get rid of the CRs from the Table
```

```
If Not StripCRs(ENVersion) Then GoTo Finish 'this line calls the function StripCRs
```

```
'Put the required info at the top of the Tab Delimited file
```

```
Set varTextFile = CreateObject("Scripting.FileSystemObject")
```

```
Set varFileName = varTextFile.CreateTextFile(strFileName, True)
```

```
    varFileName.WriteLine "*Generic" 'required 1st line by EndNote
```

```
    varFileName.WriteLine strHeaderRow(ENVersion) 'puts the names of the fields in the second row of the file
```

```
Set db = CurrentDb
```

```
DoCmd.Hourglass False
```

```
If DCount("*", "References", "Nz(References.[Record Number])=0") > 0 Then
```

```
    strChoice = "One or more records do not contain a valid number in the 'Record Number' " & Chr$(13) & Chr$(10) & _
```

```
        "field. Therefore, you can't pad the Tab Delimited file with dummy records" & Chr$(13) & Chr$(10) & _
```

```
        "in order to match the record numbers in your old library. " & Chr$(13) & Chr$(10) & _
```

```
        Chr$(13) & Chr$(10) & _
```

```

        "Do you wish to proceed without adding dummy records?"
    If MsgBox(strChoice, vbYesNo, "Missing Record Numbers") = vbNo Then
        GoTo Finish
    Else
        strChoice = vbNo 'We're proceeding with the export, but without dummy records
    End If
Else
    strChoice = "Do you want to pad the Tab Delimited file with dummy records" & Chr$(13) & Chr$(10) & _
        "in order to match record numbers to your old library?" & Chr$(13) & Chr$(10) & Chr$(13) & Chr$(10) & _
        "CAUTION! This process depends on the old record numbers being in the " & Chr$(13) & Chr$(10) & _
        "'Record Number' field of your References table."
    intChoice = MsgBox(strChoice, vbYesNo, "Pad with dummy records?")
End If
If intChoice = vbNo Then
    strRst = "Select References.* From References;"
Else
    strRst = "Select References.* From References Order By References.[Record Number];"
End If

DoCmd.Hourglass True
Set rst = db.OpenRecordset(strRst, dbOpenForwardOnly)
If rst.RecordCount = 0 Then
    MsgBox "Nothing to export. No file created..."
    DoCmd.Hourglass False
    GoTo Finish
End If
ReferenceCount = 1
Do Until rst.EOF
    If intChoice = vbYes Then
        Do Until ReferenceCount >= rst.Fields("[Record Number]")
            varFileName.WriteLine "Generic" & Chr$(9) & "Delete"
            ReferenceCount = ReferenceCount + 1
        Loop
    End If
    strWrite = rst.Fields(0).Value
    For intXX = 1 To rst.Fields.Count - 3 'Fields count starts with 0. We don't want the last two.
        strWrite = strWrite & Chr$(9) & rst.Fields(intXX).Value
    Next intXX
    varFileName.WriteLine strWrite
    rst.MoveNext
    ReferenceCount = ReferenceCount + 1
Loop
varFileName.Close
DoCmd.Hourglass False
Finish:
MsgBox "Finished"

```

```

Exit Function
ErrorTrap:
  DoCmd.Hourglass False
  MsgBox "Error #" & Err.Number & ": " & Err.Description, vbExclamation, "ExportReferencesToEndNote"
  Resume Finish
End Function

Public Function StripCRs(ENVersion As Integer) As Boolean
  On Error GoTo ErrorTrap
  Dim db As Database
  Dim rst As Recordset
  Dim intXX As Integer
  Dim strDelete As String 'this is the string we want to get rid of
  Dim strReplace As String 'this is the string we want to put in place of strDelete

  strDelete = Chr$(13) & Chr$(10)      'These are the CR/LF characters that have to stripped from the file
  strReplace = " "                    'We are just going to put a space in place of the CR/LFs
                                      'Any other characters could be substituted instead.

  Set db = CurrentDb
  DoCmd.Hourglass True
  If ENVersion = 3 Then
    Set rst = db.OpenRecordset(SqlString3, dbOpenDynaset) 'SqlString3 is a function (below)
  Else
    Set rst = db.OpenRecordset(SqlString4To6(ENVersion), dbOpenDynaset) 'SqlString4To6 is a function
  End If
  With rst
    If .RecordCount = 0 Then GoTo Finish
    .MoveFirst
    Do Until .EOF
      For intXX = 0 To .Fields.Count - 1
        If .Fields(intXX).Type = dbMemo Or .Fields(intXX).Type = dbText Then
          If InStr(.Fields(intXX).Value, Chr$(13)) > 0 Then
            .Edit
            .Fields(intXX).Value = ReplaceStr(rst.Fields(intXX).Value, strDelete, strReplace)
            .Update
          End If
        End If
      Next intXX
      .MoveNext
    Loop
  End With
Finish:
  DoCmd.Hourglass False
  StripCRs = True
  Exit Function
ErrorTrap:

```

```

DoCmd.Hourglass False
MsgBox "Error #" & Err.Number & ": " & Err.Description, vbExclamation, "StripCRs"
StripCRs = False
End Function

Function ReplaceStr(strRaw As String, strOld As String, strNew As String) As String
' This function accepts a string (strRaw) containing characters (strOld) that are to be replaced
' by other characters (strNew). The changed string is returned.
Dim intPosit As Variant
Dim strLeft As String
Dim NumOldChars As Integer

NumOldChars = Len(strOld)
If strRaw = "" Then GoTo Finished
Loop1: 'loop1 removes the unwanted characters from the back end of the string without replacement
If Right$(strRaw, NumOldChars) = strOld Then
strRaw = Left$(strRaw, Len(strRaw) - NumOldChars)
GoTo Loop1
End If
Loop2: 'loop2 finds and replaces the characters that are not at the back end of the string
intPosit = InStr(strRaw, strOld)
If intPosit > 0 Then
strLeft = strLeft + Left$(strRaw, intPosit - 1) + strNew
strRaw = Mid$(strRaw, intPosit + NumOldChars)
GoTo Loop2
Else
strRaw = strLeft + strRaw
End If
Finished:
ReplaceStr = strRaw
End Function

Function SqlString3() As String
Dim strTemp1 As String
Dim strTemp2 As String
Dim strTemp3 As String
Dim strTemp4 As String
Dim strTemp5 As String
Dim strTemp6 As String
strTemp1 = "SELECT References.[Reference Type], " & _
"References.Author, " & _
"References.Year, " & _
"References.Title, " & _
"References.[Secondary Author], " & _
"References.[Secondary Title], " & _

```

```
"References.[Place Published], " & _
"References.Publisher, " & _
"References.Volume, " & _
"References.[Number of Volumes], " & _
"References.Number, " & _
"References.Pages, " & _
"References.[Tertiary Author], " & _
"References.[Tertiary Title], "
```

```
strTemp2 = "References.Edition, " & _
"References.Date, " & _
"References.[Type of Work], " & _
"References.[Subsidiary Author], " & _
"References.[Alternate Title], " & _
"References.[ISBN/ISSN], " & _
"References.[Custom 1], " & _
"References.[Custom 2], " & _
"References.[Custom 3], "
```

```
strTemp3 = "References.[Custom 4], " & _
"References.[Accession Number], " & _
"References.[Call Number], " & _
"References.Label, " & _
"References.Keywords, " & _
"References.Abstract, " & _
"References.Notes, " & _
"References.URL "
```

```
strTemp4 = "FROM [References] " & _
"WHERE (((References.[Reference Type]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.Author) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.Year) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.Title) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Secondary Author]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Secondary Title]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Place Published]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.Publisher) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.Volume) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Number of Volumes]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.Number) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.Pages) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Tertiary Author]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Tertiary Title]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.Edition) Like '*' & Chr$(13) & '*')) "
```

```
strTemp5 = "OR (((References.Date) Like '*' & Chr$(13) & '*')) " & _
```

```
"OR (((References.[Type of Work]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Subsidiary Author]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Alternate Title]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[ISBN/ISSN]) Like '*' & Chr$(13) & '*')) "
```

```
strTemp6 = "OR (((References.[Custom 1]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Custom 2]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Custom 3]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Custom 4]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Accession Number]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Call Number]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Label]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Keywords]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Abstract]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Notes]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[URL]) Like '*' & Chr$(13) & '*'))"
```

```
SqlString3 = strTemp1 & strTemp2 & strTemp3 & strTemp4 & strTemp5 & strTemp6 & ";"
```

```
End Function
```

```
Function SqlString4To6(ENVersion As Integer) As String
```

```
Dim strTemp1 As String
Dim strTemp2 As String
Dim strTemp3 As String
Dim strTemp4 As String
Dim strTemp5 As String
Dim strTemp6 As String
```

```
strTemp1 = "SELECT References.[Reference Type], " & _
"References.Author, " & _
"References.Year, " & _
"References.Title, " & _
"References.[Secondary Author], " & _
"References.[Secondary Title], " & _
"References.[Place Published], " & _
"References.Publisher, " & _
"References.Volume, " & _
"References.[Number of Volumes], " & _
"References.Number, " & _
"References.Pages, " & _
"References.Section, " & _
"References.[Tertiary Author], " & _
"References.[Tertiary Title], "
```

```

strTemp2 = "References.Edition, " & _
"References.Date, " & _
"References.[Type of Work], " & _
"References.[Subsidiary Author], " & _
"References.[Short Title], " & _
"References.[Alternate Title], " & _
"References.[ISBN/ISSN], " & _
"References.[Original Publication], " & _
"References.[Reprint Edition], " & _
"References.[Reviewed Item], " & _
"References.[Custom 1], " & _
"References.[Custom 2], " & _
"References.[Custom 3], "

strTemp3 = "References.[Custom 4], " & _
"References.[Custom 5], " & _
"References.[Custom 6], " & _
"References.[Accession Number], " & _
"References.[Call Number], " & _
"References.Label, " & _
"References.Keywords, " & _
"References.Abstract, " & _
"References.Notes, " & _
"References.URL, "
If ENVersion = 6 Then
    strTemp3 = strTemp3 & "References.[Author Address], " & _
"References.Caption "
Else
    strTemp3 = strTemp3 & "References.[Author Address] "
End If

strTemp4 = "FROM [References] " & _
"WHERE (((References.[Reference Type]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.Author) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.Year) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.Title) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Secondary Author]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Secondary Title]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Place Published]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.Publisher) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.Volume) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Number of Volumes]) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.Number) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.Pages) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.Section) Like '*' & Chr$(13) & '*')) " & _
"OR (((References.[Tertiary Author]) Like '*' & Chr$(13) & '*')) " & _

```

```
"OR ((References.[Tertiary Title]) Like '*' & Chr$(13) & '*') " & _
"OR ((References.Edition) Like '*' & Chr$(13) & '*') "
```

```
strTemp5 = "OR ((References.Date) Like '*' & Chr$(13) & '*') " & _
"OR ((References.[Type of Work]) Like '*' & Chr$(13) & '*') " & _
"OR ((References.[Subsidiary Author]) Like '*' & Chr$(13) & '*') " & _
"OR ((References.[Short Title]) Like '*' & Chr$(13) & '*') " & _
"OR ((References.[Alternate Title]) Like '*' & Chr$(13) & '*') " & _
"OR ((References.[ISBN/ISSN]) Like '*' & Chr$(13) & '*') " & _
"OR ((References.[Original Publication]) Like '*' & Chr$(13) & '*') " & _
"OR ((References.[Reprint Edition]) Like '*' & Chr$(13) & '*') " & _
"OR ((References.[Reviewed Item]) Like '*' & Chr$(13) & '*') "
```

```
strTemp6 = "OR ((References.[Custom 1]) Like '*' & Chr$(13) & '*') " & _
"OR ((References.[Custom 2]) Like '*' & Chr$(13) & '*') " & _
"OR ((References.[Custom 3]) Like '*' & Chr$(13) & '*') " & _
"OR ((References.[Custom 4]) Like '*' & Chr$(13) & '*') " & _
"OR ((References.[Custom 5]) Like '*' & Chr$(13) & '*') " & _
"OR ((References.[Custom 6]) Like '*' & Chr$(13) & '*') " & _
"OR ((References.[Accession Number]) Like '*' & Chr$(13) & '*') " & _
"OR ((References.[Call Number]) Like '*' & Chr$(13) & '*') " & _
"OR ((References.Label) Like '*' & Chr$(13) & '*') " & _
"OR ((References.Keywords) Like '*' & Chr$(13) & '*') " & _
"OR ((References.Abstract) Like '*' & Chr$(13) & '*') " & _
"OR ((References.Notes) Like '*' & Chr$(13) & '*') " & _
"OR ((References.URL) Like '*' & Chr$(13) & '*') " & _
"OR ((References.[Author Address]) Like '*' & Chr$(13) & '*'))"
If ENVersion = 6 Then
    strTemp6 = strTemp6 & " OR ((References.Caption) Like '*' & Chr$(13) & '*'))"
End If
```

```
SqlString4To6 = strTemp1 & strTemp2 & strTemp3 & strTemp4 & strTemp5 & strTemp6 & ";"
```

```
End Function
```

```
Function strHeaderRow(ENVersion As Integer) As String
```

```
Dim strTemp As String
```

```
Select Case ENVersion
```

```
Case 3
```

```
strTemp = "Reference Type" & Chr$(9) & "Author" & Chr$(9) & "Year" & Chr$(9) & "Title" & Chr$(9) & _
"Secondary Author" & Chr$(9) & "Secondary Title" & Chr$(9) & "Place Published" & Chr$(9) & _
"Publisher" & Chr$(9) & "Volume" & Chr$(9) & "Number of Volumes" & Chr$(9) & "Number" & Chr$(9) & _
"Pages" & Chr$(9) & "Tertiary Author" & Chr$(9) & "Tertiary Title" & Chr$(9) & _
"Edition" & Chr$(9) & "Date" & Chr$(9) & "Type of Work" & Chr$(9) & "Subsidiary Author" & Chr$(9) & _
"Alternate Title" & Chr$(9) & "ISBN/ISSN" & Chr$(9) & "Custom 1" & Chr$(9) & "Custom 2" & Chr$(9) & _
```

```
"Custom 3" & Chr$(9) & "Custom 4" & Chr$(9) & "Accession Number" & Chr$(9) & _  
"Call Number" & Chr$(9) & "Label" & Chr$(9) & "Keywords" & Chr$(9) & "Abstract" & Chr$(9) & _  
"Notes" & Chr$(9) & "URL"
```

Case 4, 5

```
strTemp = "Reference Type" & Chr$(9) & "Author" & Chr$(9) & "Year" & Chr$(9) & "Title" & Chr$(9) & _  
"Secondary Author" & Chr$(9) & "Secondary Title" & Chr$(9) & "Place Published" & Chr$(9) & _  
"Publisher" & Chr$(9) & "Volume" & Chr$(9) & "Number of Volumes" & Chr$(9) & "Number" & Chr$(9) & _  
"Pages" & Chr$(9) & "Section" & Chr$(9) & "Tertiary Author" & Chr$(9) & "Tertiary Title" & Chr$(9) & _  
"Edition" & Chr$(9) & "Date" & Chr$(9) & "Type of Work" & Chr$(9) & "Subsidiary Author" & Chr$(9) & _  
"Short Title" & Chr$(9) & "Alternate Title" & Chr$(9) & "ISBN/ISSN" & Chr$(9) & _  
"Original Publication" & Chr$(9) & "Reprint Edition" & Chr$(9) & "Reviewed Item" & Chr$(9) & _  
"Custom 1" & Chr$(9) & "Custom 2" & Chr$(9) & "Custom 3" & Chr$(9) & "Custom 4" & Chr$(9) & _  
"Custom 5" & Chr$(9) & "Custom 6" & Chr$(9) & "Accession Number" & Chr$(9) & _  
"Call Number" & Chr$(9) & "Label" & Chr$(9) & "Keywords" & Chr$(9) & "Abstract" & Chr$(9) & _  
"Notes" & Chr$(9) & "URL" & Chr$(9) & "Author Address"
```

Case 6

```
strTemp = "Reference Type" & Chr$(9) & "Author" & Chr$(9) & "Year" & Chr$(9) & "Title" & Chr$(9) & _  
"Secondary Author" & Chr$(9) & "Secondary Title" & Chr$(9) & "Place Published" & Chr$(9) & _  
"Publisher" & Chr$(9) & "Volume" & Chr$(9) & "Number of Volumes" & Chr$(9) & "Number" & Chr$(9) & _  
"Pages" & Chr$(9) & "Section" & Chr$(9) & "Tertiary Author" & Chr$(9) & "Tertiary Title" & Chr$(9) & _  
"Edition" & Chr$(9) & "Date" & Chr$(9) & "Type of Work" & Chr$(9) & "Subsidiary Author" & Chr$(9) & _  
"Short Title" & Chr$(9) & "Alternate Title" & Chr$(9) & "ISBN/ISSN" & Chr$(9) & _  
"Original Publication" & Chr$(9) & "Reprint Edition" & Chr$(9) & "Reviewed Item" & Chr$(9) & _  
"Custom 1" & Chr$(9) & "Custom 2" & Chr$(9) & "Custom 3" & Chr$(9) & "Custom 4" & Chr$(9) & _  
"Custom 5" & Chr$(9) & "Custom 6" & Chr$(9) & "Accession Number" & Chr$(9) & _  
"Call Number" & Chr$(9) & "Label" & Chr$(9) & "Keywords" & Chr$(9) & "Abstract" & Chr$(9) & _  
"Notes" & Chr$(9) & "URL" & Chr$(9) & "Author Address" & Chr$(9) & "Caption"
```

Case Else

```
End Select  
strHeaderRow = strTemp  
End Function
```