Process to convert data in various formats to TSV (Tab Separated Values)
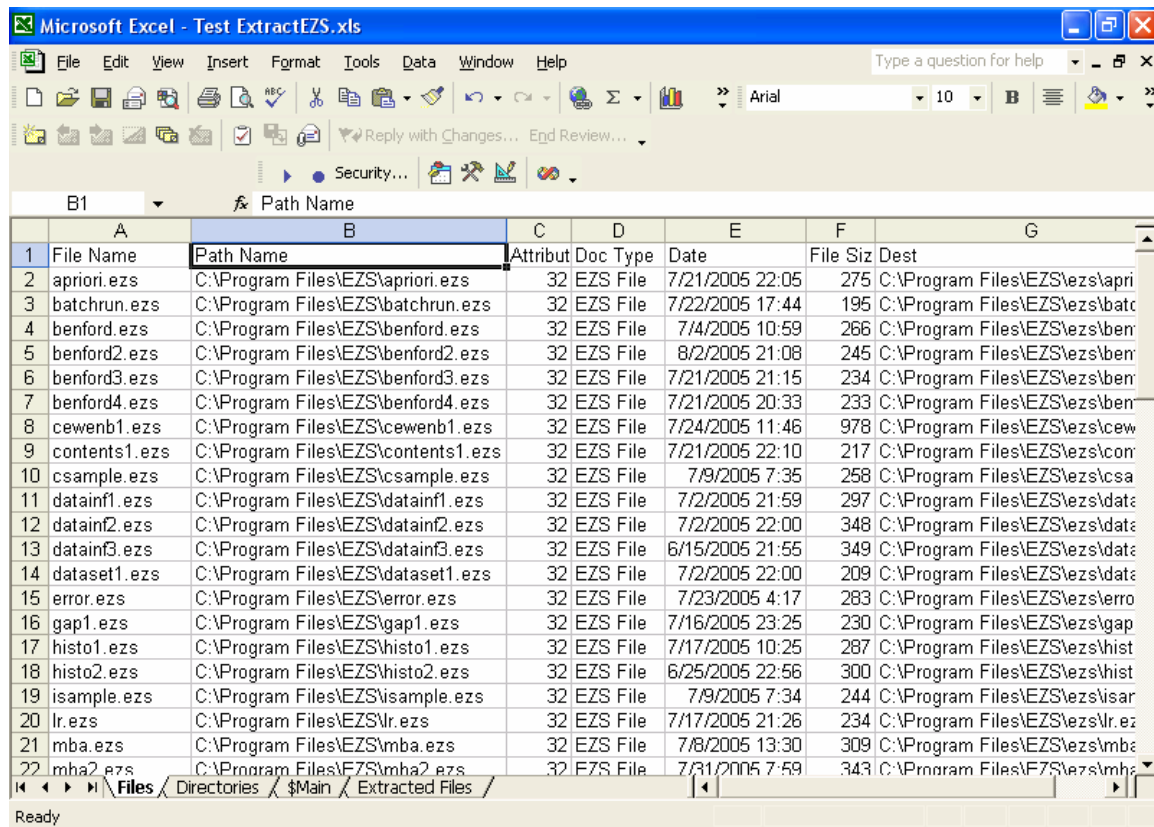
Process to convert data in various formats to TSV (Tab Separated Values)

Exporting from MS-Excel

There are at least two methods for exporting data from MS-Excel. Perhaps the simplest (and most direct) method is to open MS-Excel and select the sheet to be copied. Then select the area to be copied and use either Ctrl-C or Edit | Copy to place the contents of the sheet on the Clip Board.

Open a Word Pad document and paste the clip board onto it, using either Ctrl-V or Edit | Paste. Save the document as a text file (not RTF). You may optionally choose a file extension of other than ".txt", e.g. (".tsv" or ".tab").

The second method to export data from MS-Excel, exports a sheet at a time using the MS-Excel menus. Select the sheet which contains the data, then File|Save As and file type of Text (tab delimited) (*.txt).
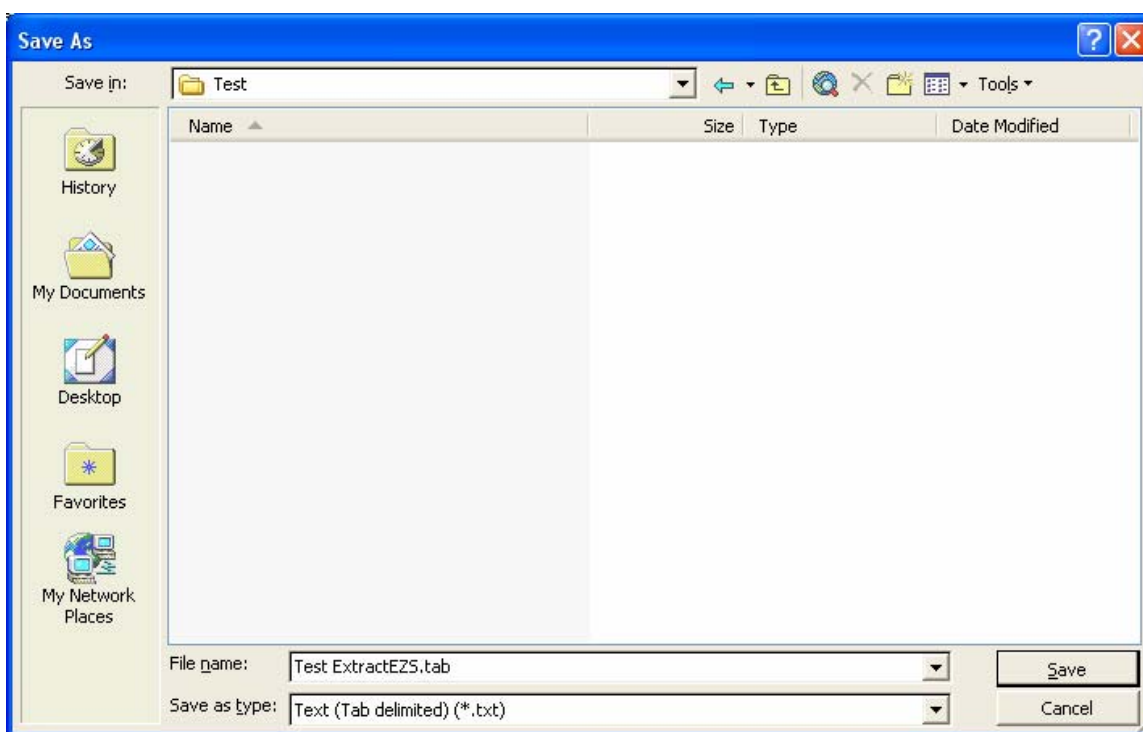

Select, File | Save As, with file type as (Text, Tab Delimited), and specify the file name to save as having an extension of "tab".

Process to convert data in various formats to TSV (Tab Separated Values)

When you save the file, you will receive a message that the selected file type doesn't support multiple sheets – Click OK.
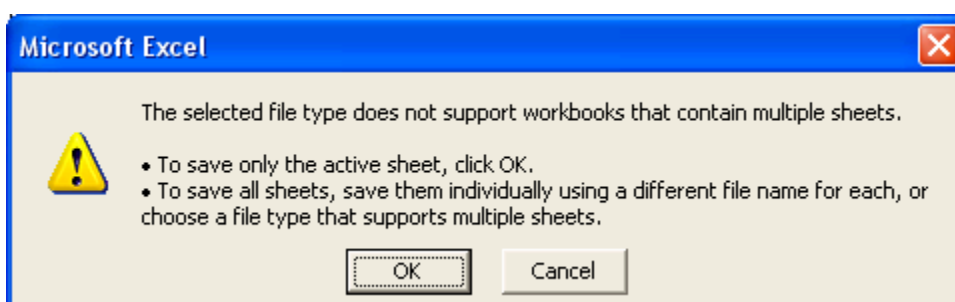
You are then prompted if you want to keep the workbook in this format – select No.
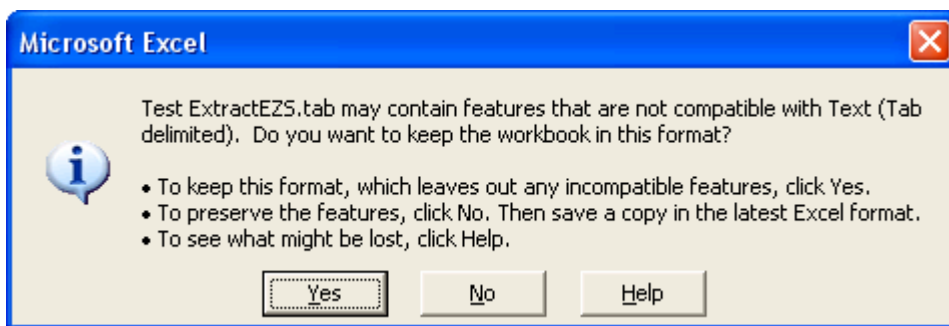
Process to convert data in various formats to TSV (Tab Separated Values)

You can cancel this dialog (as the file has already been saved).
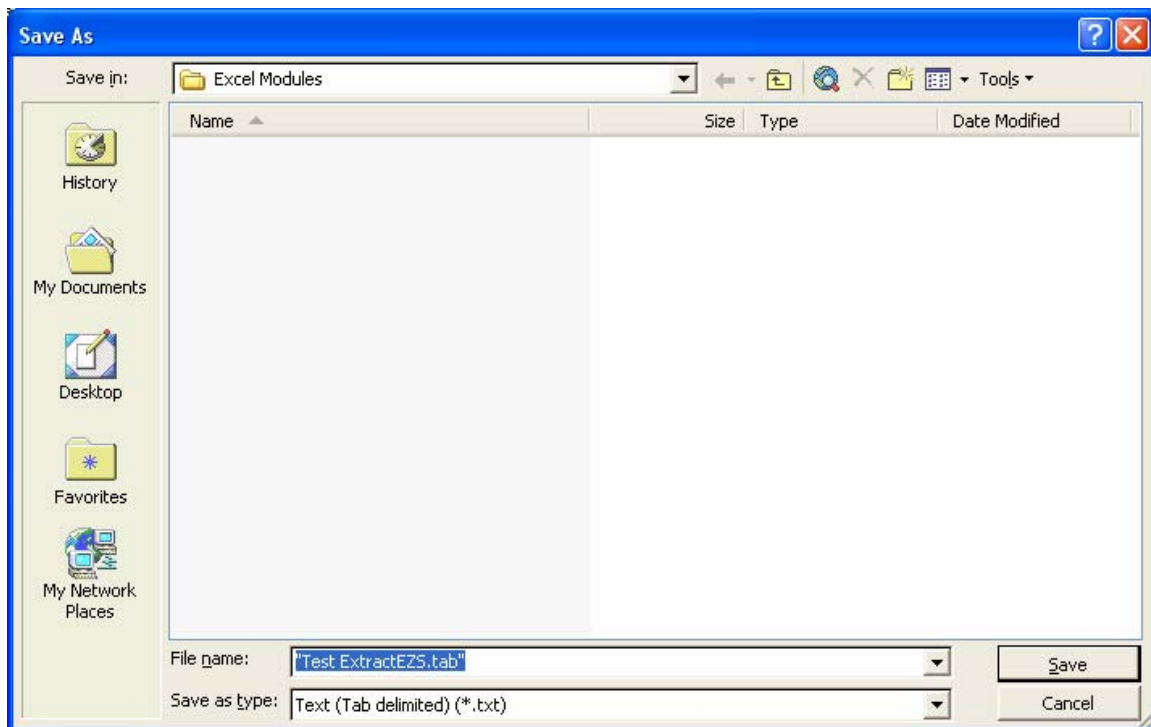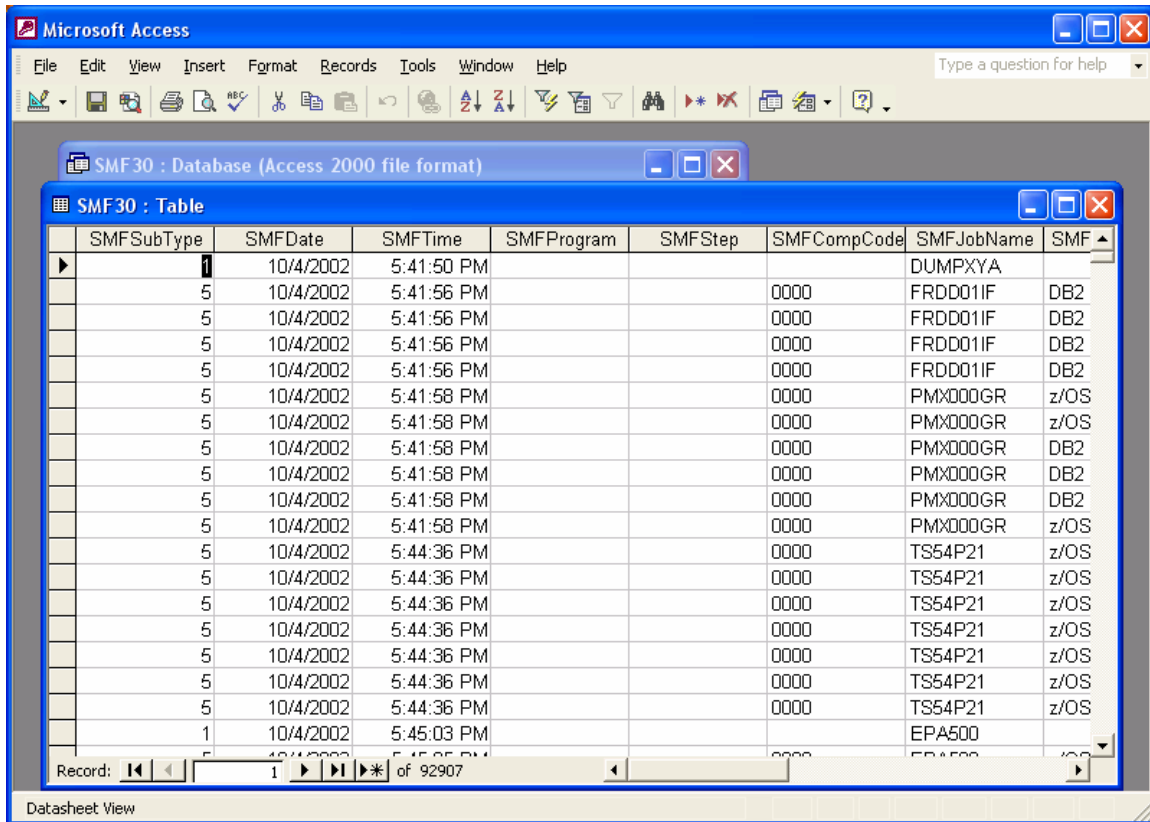
The Excel sheet has been converted to tab separated file format.

Exporting from MS-Access

To convert a table stored in MS-Access to tab separated file format, you will need to open the table in MS-Access. Select the database and the table to be exported.

Process to convert data in various formats to TSV (Tab Separated Values)

Select File | Export and choose the text file format:

Process to convert data in various formats to TSV (Tab Separated Values)

Process to convert data in various formats to TSV (Tab Separated Values)

Select the "Advanced" button, in order to specify the field separators.
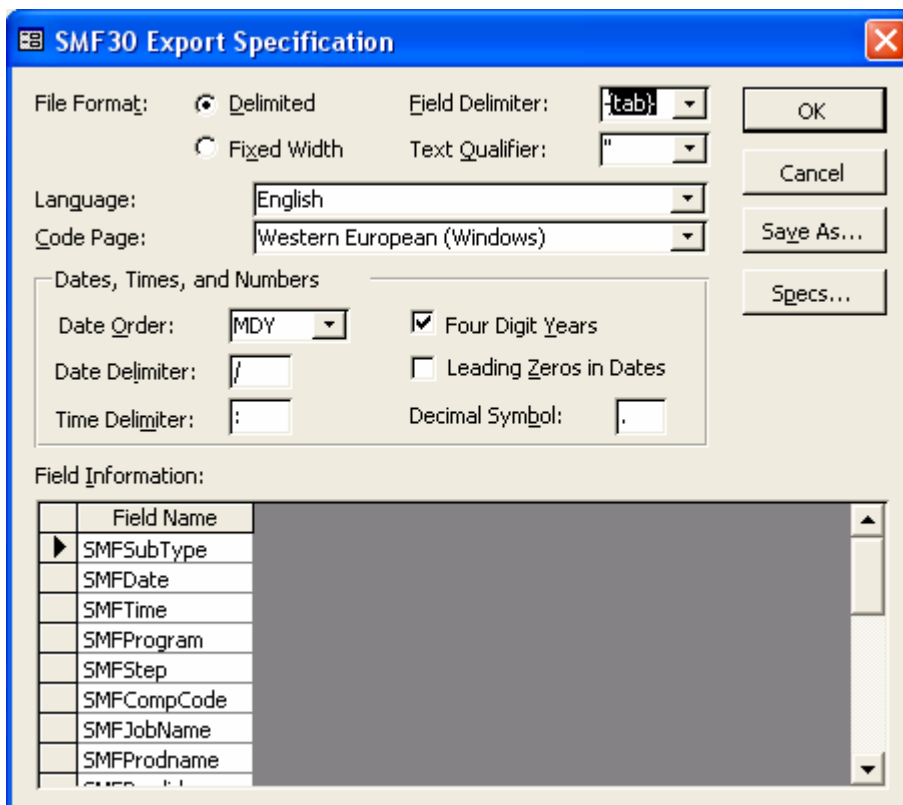
Process to convert data in various formats to TSV (Tab Separated Values)

Specify that the first row is to contain the field names

Process to convert data in various formats to TSV (Tab Separated Values)

Provide the file name for the tab separated file to be saved:

Save it, and you're finished:
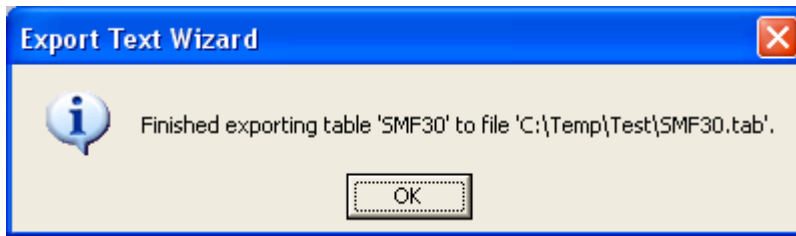
Converting a flat file

There are various ways to convert a flat file to a tab separated file. For small files of less than 65,000 rows, the data can be imported into MS-Excel, column titles inserted into the first row, and then save the Excel worksheet as described above.

For larger files, they may be imported into MS-Access, and then exported as a tab separated file.

Another alternative is to use a scripting language such as ScriptBasic to read the external flat file and write to the output tab separated file.

An example of a script basic routine to convert a text file is as follows:

```
vbTab = "\t"
vbCrLf = "\n"
on error goto cantopenin
Open "c:\\cloop\\ndc\\data\\firms.txt" For Input As 1
on error goto cantopenout
Open "c:\\test\\model\\g2.tab" For Output As 2
On Error GoTo erreoj
obuff = "seqno" & vbtab & "lblcode" & vbtab & "firmname" &
vbcrlf
Print #2, obuff
While Not EOF(1)
Line Input #1, buffer
seqno = mid(buffer,1,7)
lblcode = mid(buffer,8,6)
firmname = mid(buffer,14,65)
obuff = seqno & vbTab & lblcode & vbTab & firmname & vbcrlf
Print #2, obuff
deleterec:
Wend
Close 1
Close 2
Print "normal eoj...\n"
```

```
Stop
cantopenin:
Print "cant open input...\n"
Stop
cantopenout:
Print "cant open output...\n"
Stop
erreoj:
Print "error eoj...\n"
Stop
```

Exporting from MySQL database

Tab separated files may be created from a MySQL database by using a Select Into statement. (Select expr into outfile 'filename'. The default file options can be used, i.e. there is no need to specify fields terminated by '\t' and lines terminated by '\n'.

Exporting from SQLite database

Tab separated files may also be created from a SQLite database using any of the following methods:

1. The SQLite query dialog available from www.ezrstats.com
2. The SQlite utility routine command line sqlite.exe which may be downloaded from www.sqlite.org
3. Running sqlite.exe as a DOS command and redirecting the output to a file

Exporting from ODBC data source

Open Excel, and Select Data | Import Data | New Data Base Query

MS-Excel will ask for the desired datasource:

Process to convert data in various formats to TSV (Tab Separated Values)
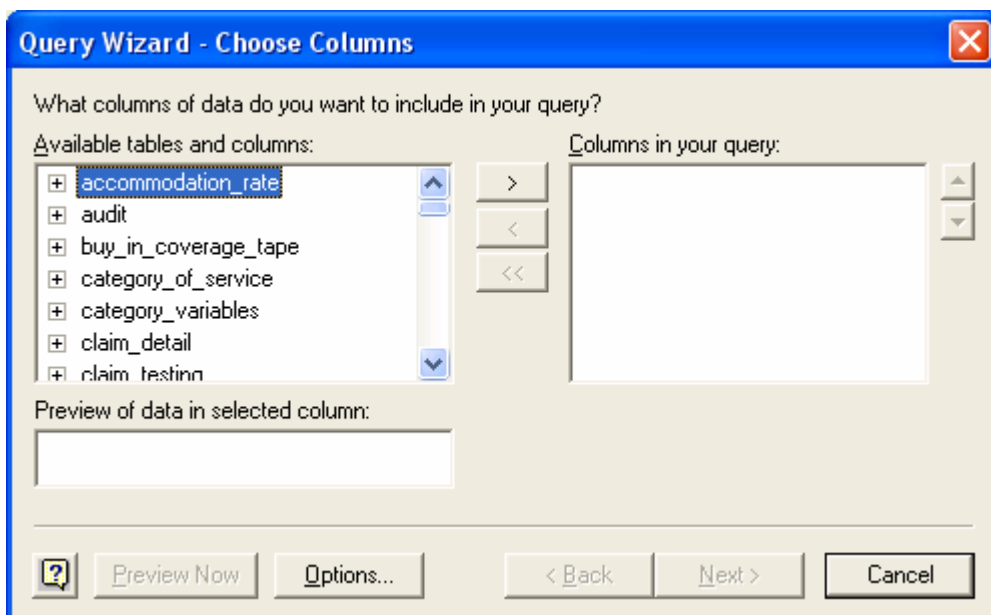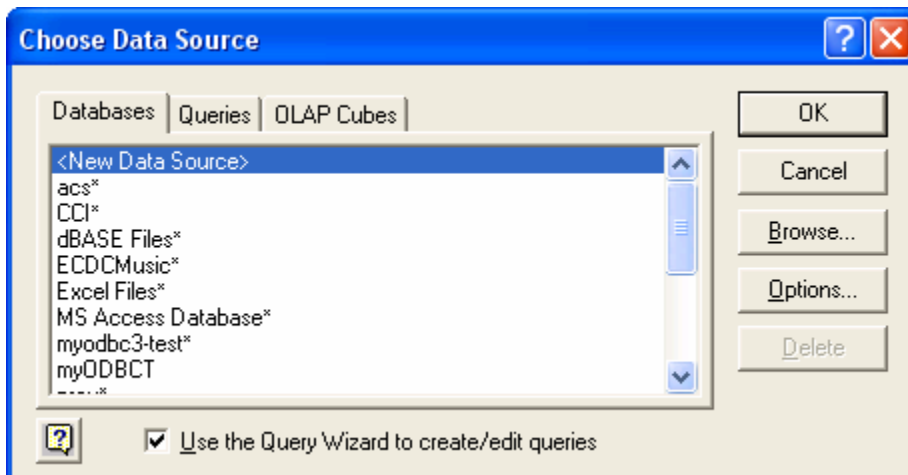
Process to convert data in various formats to TSV (Tab Separated Values)

Query Wizard - Choose Columns

What columns of data do you want to include in your query?

Available tables and columns:

```
☐ audit
    sysid
    actMonth
    segcount
    access
    cc
    annlam
```

Columns in your query:

```
userid
jobname
actyear
tpind
```

Preview of data in selected column:

[Preview Now] [Options...]   < Back   Next >   Cancel

Query Wizard - Filter Data

Filter the data to specify which rows to include in your query.
If you don't want to filter the data, click Next.

Column to filter:

```
userid
jobname
actyear
tpind
```

Only include rows where:

userid

begins with ▼   BZB ▼

⦿ And   ○ Or
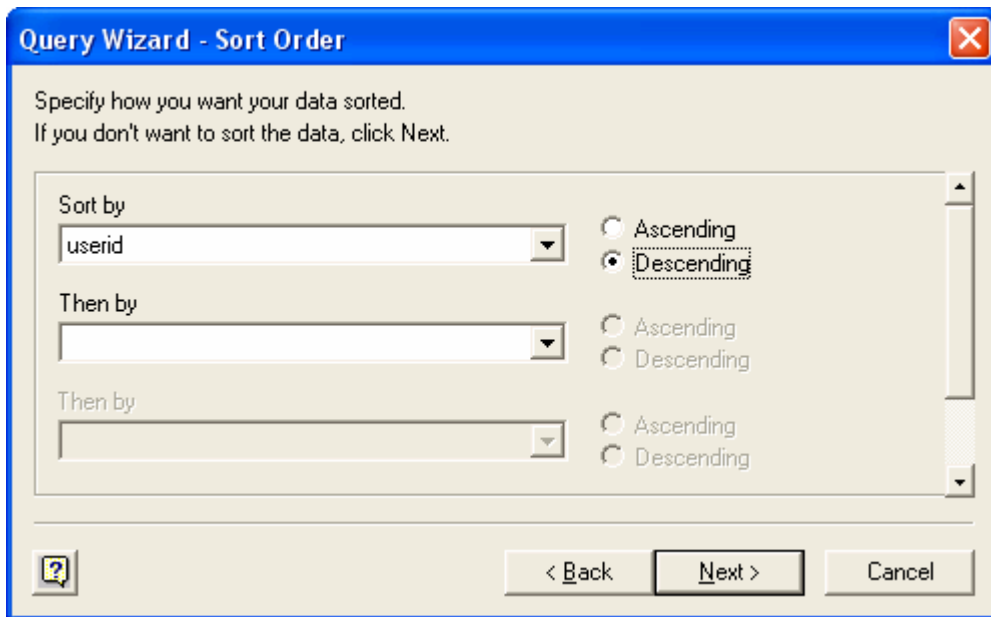
○ And   ○ Or

○ And   ○ Or

< Back   Next >   Cancel

Process to convert data in various formats to TSV (Tab Separated Values)

Query Wizard - Sort Order

Specify how you want your data sorted.
If you don't want to sort the data, click Next.

Sort by
[userid] ○ Ascending ● Descending
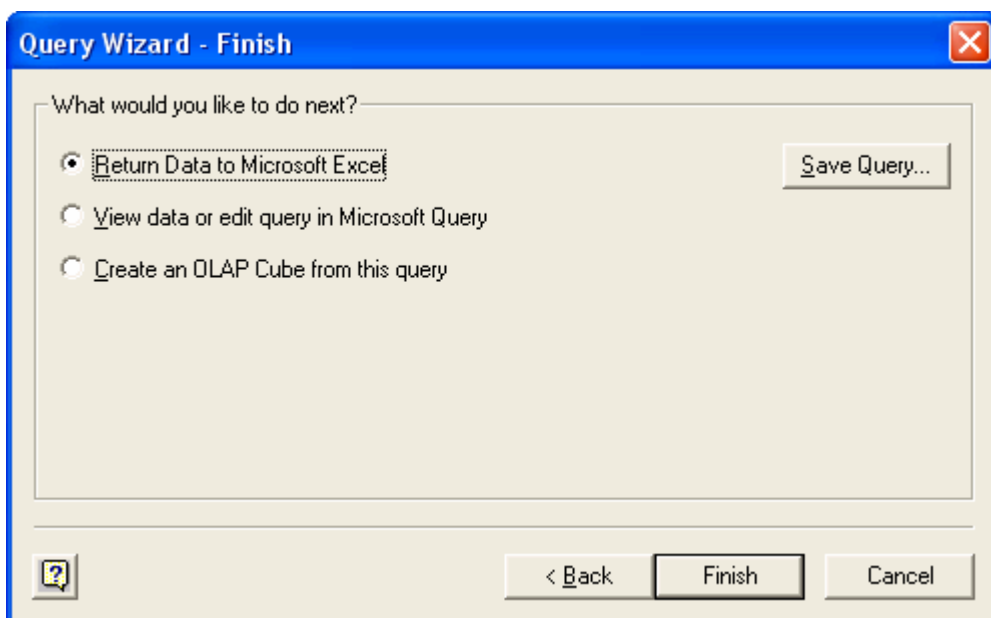
Then by
[ ] ○ Ascending ○ Descending

Then by
[ ] ○ Ascending ○ Descending

< Back   Next >   Cancel

Query Wizard - Finish

What would you like to do next?

● Return Data to Microsoft Excel   Save Query...

○ View data or edit query in Microsoft Query

○ Create an OLAP Cube from this query

< Back   Finish   Cancel

Process to convert data in various formats to TSV (Tab Separated Values)
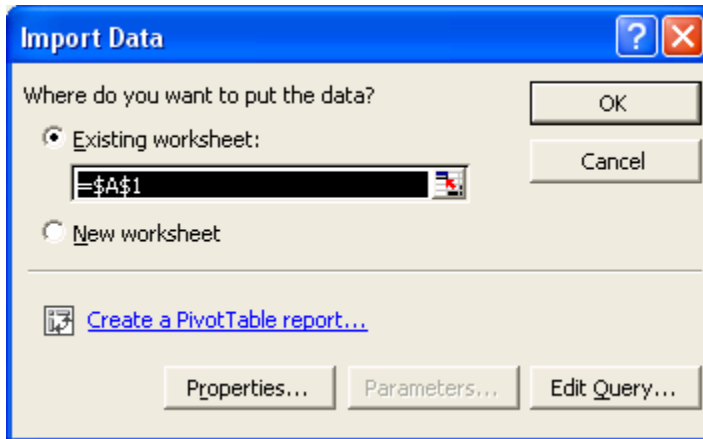
The data is returned as an Excel Sheet.
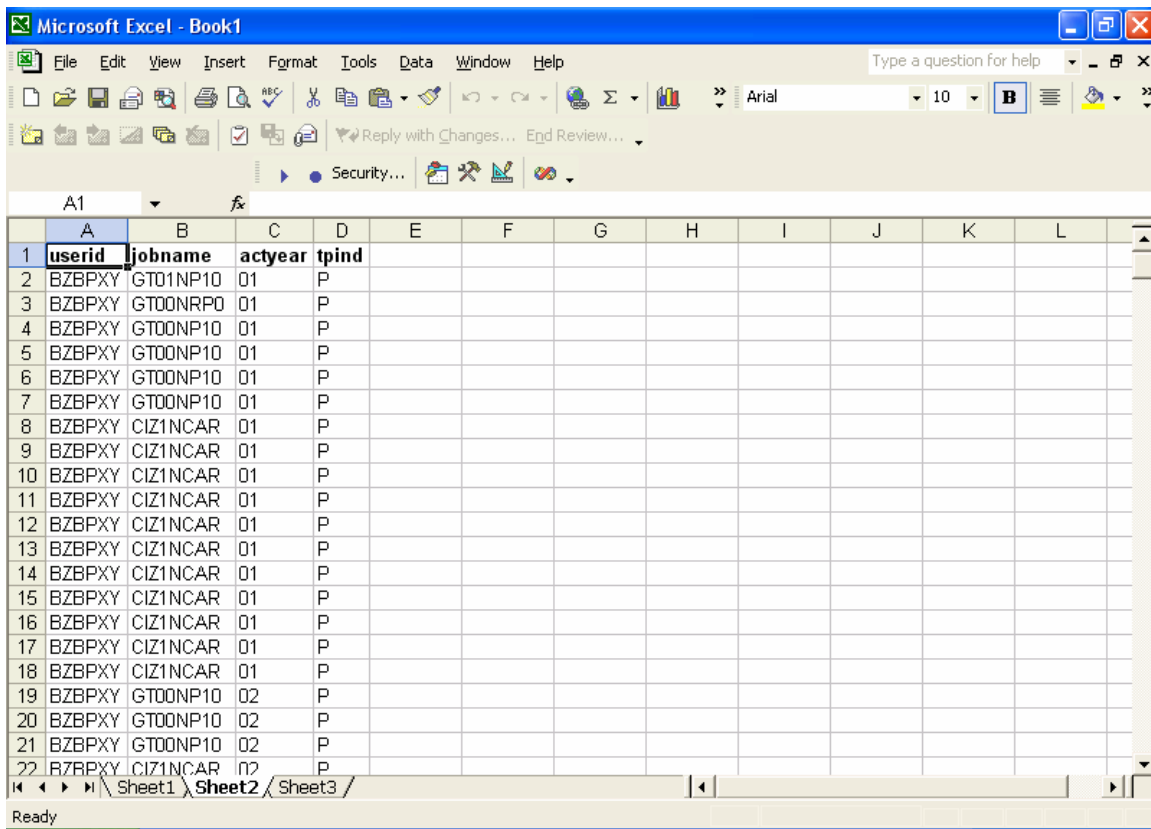
Converting Comma Separated Values (CSV)

To convert a coma separated value file, generally, it should be first imported into MS-Excel and then saved as a tab separated file.

Converting Dbase Files

Dbase files can also be converted using MS-Excel, once an ODBC datasource has been established for them. They can also be imported into MS-Access and then converted from MS-Access as described above.

Tab Separated Values (TSV): a format for tabular data exchange

TSV is a very simple textual data format which allows tabular data to be exchanged between applications that use different internal data formats. This document briefly explains this standardized format and gives simple examples of using it.

Content:

- The TSV format
- Not to be shown as is!
- Importing TSV data to MS Excel
- Generating TSV data
- Exporting data in TSV format from Excel
- Converting from TSV format to HTML table format

The TSV format

The Tab Separated Values has been officially registered as an Internet media type (MIME type) under the name `text/tab-separated-values`. Note that major media type `text` means that the data is in textual format which can be viewed and edited using a normal text editor; in practice however this is seldom done - instead, the format is programmatically written and read.

The abbreviation TSV (or tsv) is commonly used, and usually the file name suffix for TSV files is "`.tsv`".

The registry entry is very short, since the format is very simple, but the description is paraphrased here in plain English (as opposite to the partly formalized notation in the registration):

- A file in TSV format consists of lines.
- Each line contain fields separated from each other by TAB characters (horizontal tab, HT, Ascii control code 9).

- "Field" means here just any string of characters, excluding TABs. The point is simply that TABs divide a line into pieces, components.
- Each line must contain the same number of fields.
- The first line contains the names for the fields (on all lines), i.e. column headers.

The last rule could be taken just as *allowing* the labeling of columns that way. Mostly if you wish to include just tabular data as such, it probably works OK. But it's usually recommendable to use some labeling there.

Since TAB is used as a separator between fields, a field cannot contain a TAB. However, TABs usually don't appear in data items that you wish to tabulate, so this is seldom a restriction. There are various other formats which are very similar to TSV but use a different separator, such as Comma Separated Values (CSV). Note that commas, spaces, and other characters often used as separators in such formats appear rather often in data to be tabulated, at least in header fields. Special conventions can be made to allow the inclusion of the separator into data fields, but the extreme simplicity of the data format is then lost.

The example given in the registration is the following, where <TAB> denotes a TAB character:

```
Name<TAB>Age<TAB>Address
Paul<TAB>23<TAB>1115 W Franklin
Bessy the Cow<TAB>5<TAB>Big Farm Way
Zeke<TAB>45<TAB>W Main St
```

Not to be shown as is!

Although TSV is a text format, a TSV file is *not* expected to appear in a nice tabular format when displayed or printed as such, e.g. by an editor. In special cases it might do that, if tab stops are set suitably in the environment and if the fields in a column have roughly the same width. But even the simple example probably fails when embedded into an HTML document (inside a PRE element, which is the best shot):

```
Name     Age      Address
Paul     23       1115 W Franklin
Bessy the Cow  5       Big Farm Way
Zeke     45       W Main St
```

Using TABs in HTML is unreliable, but more fundamentally, tabbing with TAB is in itself not structural. It just takes to some predefined tab stop; typically the tab stops are set at 1st, 9th, 17th, etc. character position. This means that the data would look OK by accident only. And when using TSV, you are *not* supposed to try to "fix" things by using multiple TABs; that would mean that there are *empty fields* between the TABs, defeating the very idea of TSV.
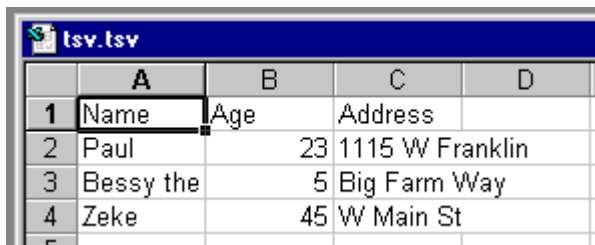
What happens if you ask your Web browser to show a TSV file, e.g. by following a link to a file containing our example? It really depends on many things; it's a matter of handling media types, which can be a bit confusing. It most essentially depends on the settings of your browser, but also on how the server sends it. But in any case, you as a

user have at least as one option the possibility of *saving the file* onto local disk, to be processed separately using some suitable program (e.g. MS Excel). So TSV is one way of making tabular data available on a network in a universally accessible way, though it may require some skills on the user side to know what could be done with the data and do something useful with it.

Importing TSV data to MS Excel

We could give a TSV file as input to a spreadsheet application, like MS Excel. Such programs generally accept TSV format data. The data can then be viewed, printed, and manipulated in tabular format. And you can save the data in MS Excel's "native" format (which cannot be processed by other programs unless they have been specifically written to process it) or in TSV format, or in a few other formats. Our simple example data would get displayed roughly as follows:

Cells with long content may *look* truncated. This is Excel's feature and often useful when manipulating spreadsheets. The presentation, both on screen and on paper, can be tuned as needed, using various tools in Excel.

When you have Excel in use, you can open the File menu, then select Open, and pick up the file to be opened. For finding the file, you may need to set value of the "Files of type:" setting (which is effectively a file type filter) to "All files (*.*)". (The details may vary depending on the version and configuration of the program.)

But you can make things easier by setting things up so that your system (we're assuming some flavor of Windows here) automatically recognizes `.tsv` files as something to be opened in Excel, when you click on their icons. Assuming there is no such association for `.tsv` in your system yet, you'll be prompted for one when you first click on the icon of a `.tsv` file. When prompted for "Description of '.tsv' files:", you could reply e.g. "text/tab-separated-values". Then pick up Excel from the list of programs and click OK.

When TSV data is imported into Excel, some data transformation and interpretation may take place. For example, some strings might be interpreted as dates and converted to a different date format, and Excel generally treats a string of digits as a number and right-justifies it. In such cases, "opening by clicking" could be inadequate, since it might not give you the option of specifying how the data is to be interpreted. Explicitly opening the file in Excel normally lets you use "Text import wizard" where you can specify the interpretation of each column.

Generating TSV data

Process to convert data in various formats to TSV (Tab Separated Values)

For example, if you have a program of your own (say, a Fortran or C program, or a Perl script) that writes out some tabular data, and you'd like to be able to open (or let others open) the data in Excel, the TSV approach is rather simple. You just need to know how to write data in TSV format, which is *a lot* easier than trying to generate MS Excel format. Usually you can just use TAB characters in a string used to format the output, but see next paragraph as regards to editors; in C and Perl for example, you can alternatively use the notation `\t` inside a character constant or string constant.

And, of course, you could type TSV data "by hand", using a text editor. Note that typing the TAB character may require some extra trick, since many editors process it as command-like, instead of entering it into data as a character. For example, in some modes Emacs turns TABs to sequences of spaces, which won't do in TSV; to prevent this, type control-Q before hitting the TAB key.

As an example, here is Fortran 77 code that writes a matrix is TSV format:

```
      parameter(m=42)
      parameter(n=5)
      real a(m,n)
... code that writes data to matrix a ...
      do 200 i=1,m
 200     print 210, (a(i,j),j=1,n)
 210     format(F10.6,4('    ',F10.6))
```

The code is not particularly elegant, since the number of columns is hard-wired into the `format` statement, which causes the first number to printed as such and 4 more with a tab character before each of them. Between the apostrophes (single quotation marks), there is exactly one TAB character and nothing more; the appearance on your browser may vary.

Exporting data in TSV format from Excel

When you save a spreadsheet in Excel (using Save in the File menu, or using control-S), it normally uses by default its own internal data format. You can override this by using Save As... (in the File menu) instead and selecting the desired format. Among the available formats, you can pick up "Text (Tab delimited) (*.txt)", which means TSV. (Here, too, you may encounter some variation between versions of Excel.) Note that the default file name suffix will be .txt when; you may wish to override this, reserving .txt for plain text files.

Saving as TSV generally *loses formatting information*. Quite often this is *desirable*, since if you e.g. need to E-mail some data or put a table onto the Web, information about the specific font faces, cell widths, and other presentational details are not needed and can cause problems.

You could use this possibility for the job of making data available on the Internet, or in an intranet. If you have the data in Excel format, it will be less universally accessible than e.g. TSV format. Although there are Excel viewers available free of charge for Windows,

Process to convert data in various formats to TSV (Tab Separated Values)

not everyone wants to install such software just to casually view some tabular data, and not everyone uses Windows at all. And even if people have Excel installed, they might be unwilling to open documents from unknown origin, in fear of macro viruses for example. So TSV might be a good idea at least as an *alternative*. You could always make the data available both in Excel format and as TSV, especially since saving as TSV may remove some formatting information.