Users Guide For

TXL Wizard

By Softinterface and RMS Vision System

Version 1.x
Windows 95/98/2000/XP

Contents

TXL Wizard Users Manual 3

What is TXL Wizard? 3
TXL Wizard Features 3
'How Do I Use It? 4
STEP 1 4
STEP 2 5
STEP 3 5
Using the TXL Wizard 6
FILE MARKING METHOD 6
RECIPE FILE METHOD 10
What You Will Need To TXL Wizard 13
Installation and Un-installation 13
Installing 13
Uninstalling 13
Troubleshooting 13

Using TXL Wizard From the Command Line 14

Using The Command Line 14
Command Line Examples 15
All Command Line Switches 16

TXL Wizard Users Manual

What is TXL Wizard?

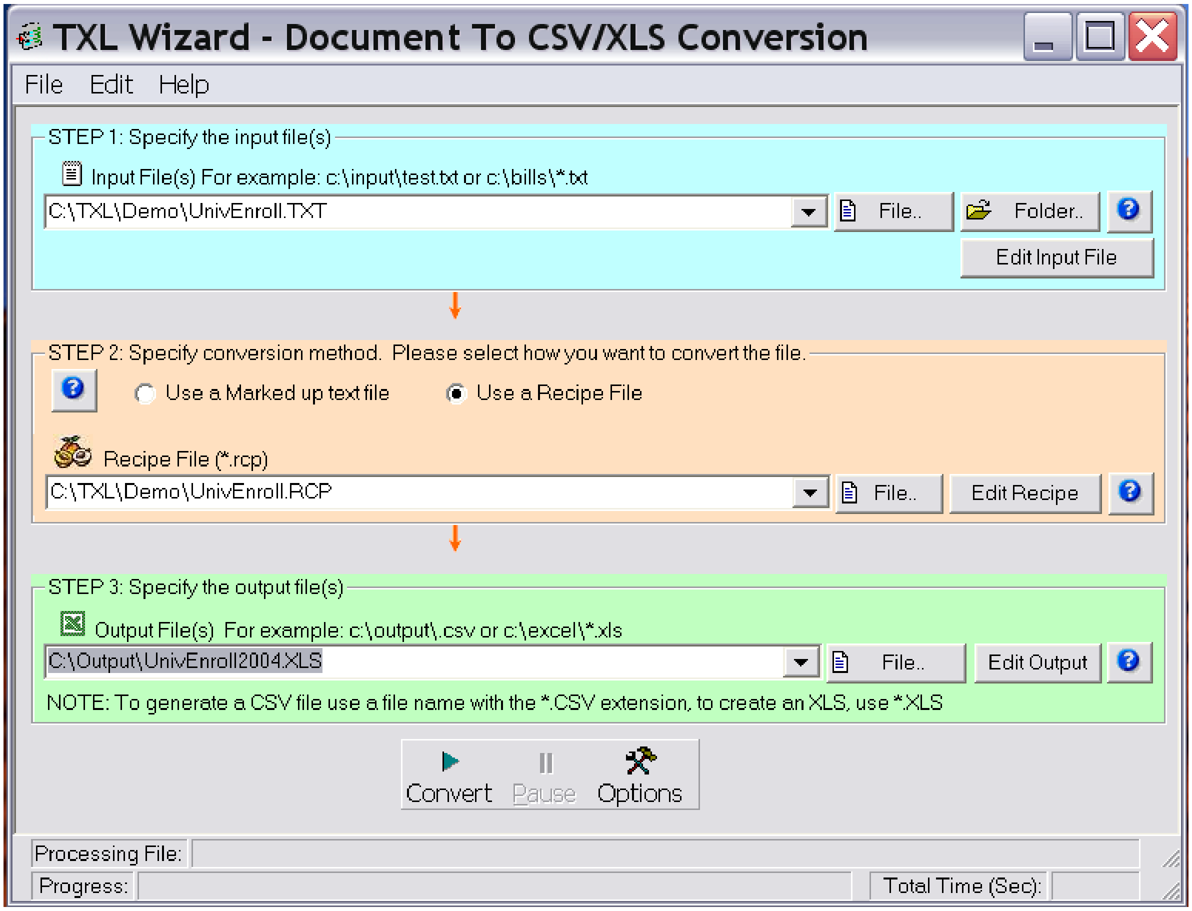
TXL Wizard is a software application designed to easily transform text files into either Excel or CSV (comma delimited text file) file types. It does this by using one of two methods, the "File Marking" method or the "Recipe File" method. You'll find both methods easy to learn. Once mastered, TXL Wizard becomes a software tool with enormous leverage.

At SoftInterface, Inc. we are constantly enhancing and improving our products. Please visit our web site to see what's new and tell us what you would like to see in our products (WWW.SoftInterface.COM). ***Also, it is important to register your products to ensure you have the latest version and support.***

TXL Wizard Features

- Powerful, fast, recipe driven, and easy to use.
- Automatically reads text reports, and creates MS Excel spreadsheets or CSV files.
- Do one or thousands of files and folders, set and forget.
- Automatically creates the report per specs saved in recipes.
- Generating a new report with different data is a single click process.
- Compatible with future changes in input format with minor recipe change.
- New recipe creation/editing typically takes a few seconds to a few minutes depending on complexity.
- Built in help for training new users.
- Delivered with example data for training.
- Command line able.
- Eliminates the possibility of human error when moving data between text and XLS/CSV files.

How Do I Use It?



STEP 1: Specify the input file(s)

Input File(s) For example: c:\input\test.txt or c:\bills\*.txt

C:\TXL\Demo\UnivEnroll.TXT

STEP 2: Specify conversion method. Please select how you want to convert the file.

○ Use a Marked up text file ● Use a Recipe File

Recipe File (*.rcp)

C:\TXL\Demo\UnivEnroll.RCP

STEP 3: Specify the output file(s)

Output File(s) For example: c:\output\.csv or c:\excel\*.xls

C:\Output\UnivEnroll2004.XLS

NOTE: To generate a CSV file use a file name with the *.CSV extension, to create an XLS, use *.XLS

Processing File:

Progress:

Total Time (Sec):

Using 'TXL Wizard' is a 3 step process, as shown in the picture above. Simply review each step below. The order of each step is not critical, but you must do each step.

Note that the Help buttons found throughout the graphical user interface will bring up a help page for the appropriate step with a single click.

STEP 1

First select the input text file(s) you are going to process. Specify the full path and file name within the "Input File" text box. If you want to specify a whole folder of files, use the wildcard (e.g. d:\inputfiles\*.txt).

Rather than typing in the full path, you may browse to a file or folder. For a single file click the File button, for an entire folder click the Folder button.

Once you've selected an input file, you can easily view and edit it by clicking the **Edit Input File** button Edit Input File. The document will open in Notepad.

See also:

STEP 2

STEP 3

STEP 2

Decide what method you will use to transform the text file, the "File Marking" method or the "Recipe File" method.

The "File Marking" method requires that you modify the input file with special character sequences typed directly into the original document. This may be easiest choice if you only expect to process a given text file once.

The "Recipe File" method can be used to process similar text files and requires recipe specification only once. A separate recipe file is created which can then be applied to multiple documents. If you anticipate having many files with similar layout, the "Recipe File" method is the appropriate choice. Keep in mind, similar layout does not mean each input text file must be exactly the same. The number of rows in a given table can be different for example, and the same recipe will work. What must be similar between files for the recipe to work, are specific text strings that the program uses to denote the start and end of a table.

See also:

STEP 1

STEP 3

STEP 3

Here you must choose the file or folder in which to place the processed files. Rather than typing in the full path, you may browse to a file or folder. For a single file click the File button, for a folder click the Folder button.

You must also choose between creating Excel (XLS) files or comma delimited files (CSV). To do this simply use the appropriate extension (*.CSV or *.XLS) in the "Output File" text box.

The CSV files generated can be opened directly within Excel without any issues. Also, and perhaps most importantly, creating a CSV file can be done much faster than creating an Excel file.

Excel output files on the other hand, although generated more slowly, can save you the extra step of converting the CSV file to an Excel file. Also, output Excel files can optionally be pre-existing which will allow for the preservation of previously saved formatting information. That means you can transfer text file data directly into a pre-formatted Excel file.

Finally, click the **Convert Button** to initiate the conversion process.

See also:

STEP 1

STEP 2

EXAMPLE FILES

To quickly get you up and running, several example text files were included with this software. You may run the samples and open them. This allows you to examine the markings and the recipes used to generate the output. They are found in the Examples folder located in the application folder (root directory). If a default installation was selected that path is C:\Program Files\SoftInterface, Inc\TXL Wizard\Examples.

EXAMPLE 1 (Marked Up): Select Example1.TXT as the input file. Example1.TXT has been marked up, so choose "Use Marked up text file". Finally specify any output file you like, for example C:\MyOutput\Output.**CSV**. To create an excel file specify C:\MyOutput\Output.**XLS**.

EXAMPLE 2 (Recipe): Select Example2.TXT as the input file. Then select "Use a Recipe File" and select Example2.RCP as the recipe file. Finally specify any output file you like, for example C:\MyOutput\Output.**CSV**. To create an excel file specify C:\MyOutput\Output.**XLS**.

Using the TXL Wizard

You should carefully read the brief material below before using the TXL Wizard. This will help you understand how the application works. Reading this material should take no more than 10 to 15 minutes. Running the examples only a few minutes more.

If your input files are not already in text format, you must convert your native documents to text. For example to go from PDF to XLS you must first convert the PDF to text. PDF to text conversion is done with 'Convert Doc'.

Once in text format, there are two main methods to translate files, namely the *File Marking* method, and the *Recipe File* method.

FILE MARKING METHOD

This method involves inserting special markers within the text file to control the translation process.

Marker	Purpose
>>n	Table Start, where n is the table number
<<n	Table End
--------	Single Column Field. Copies data to one Excel Column.
=========	Multiple Column Field. Copies data to many

	Excel columns

Summary of Markers Table

Using a combination of these simple markers, many complex data parsing tasks can be achieved.

To copy a text table to Excel, insert two control lines before the top row of the data. The two control lines you should insert are the Table Start marker followed by the required field markers. Then after the last data row of the table, insert the Table End Marker as shown below:

```
>>1                                     Table Start
Marker (n=1)
------------ ------------- ----------- Field (or Column)
Template
xxxxxxxxxxxx yyyyyyyyyyyyy zzzzzzzzzzz Text data to
import to Excel
xxxxxxxxxxxx yyyyyyyyyyyyy zzzzzzzzzzz Text data to
import to Excel
. .
. .
. .
<<1                                     Table End Marker
```

Short Cut Markers:
In the case where only a single type of field marker is required, you can use a short cut marker instead of typing many repeated "-" (dash) or "=" (equal) characters:

1. Single Filed Only: - (a single "-" at the beginning of the first column means all "–" to the end of line)

Sample 1 and Sample 2 are exactly equivalent

Sample 1	Sample 2
>>1	>>1
-	------------------------
aaaa bbbb cccc dddd	aaaa bbbb cccc dddd
aaaa bbbb cccc dddd	aaaa bbbb cccc dddd
. .	. .
. .	. .
<<1	<<1

2. Multi-Field Only: = (a single "=" at the beginning of the first column means all "=" to the end of line)

Sample 3 and Sample 4 are exactly equivalent

Sample 3	Sample 4
>>1	>>1
=	================
aaaa bbbb cccc dddd	aaaa bbbb cccc dddd
aaaa bbbb cccc dddd	aaaa bbbb cccc dddd

```
..                                   . .
..                                   . .
<<1                                  <<1
```

The following example illustrates how and where to use the file markers.

Example: Importing a text file to Excel

Assume you have a text file that looks something like this:

```
xxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxx

xxxxxxx yyyyyyy zzzz wwwwwwwwwwww
        yyyyyyy      wwwwwwwwwwww
xxxxxxx yyyyyyy zzzz wwwwwwwwwwww

xx yyyyyy zzzzzzzzzzzz wwwwwwwwwwwwww
xx        zzzzzzzzzzzz wwwwwwwwwwwwww
xx        zzzzzzzzzzzz wwwwwwwwwwwwww
```

It is possible to fit the above text (represented by the x's, y's, z's, and w's data columns) correctly into 3 tables. Each table may have different number of rows and columns.

The process is very simple:

Step 1: Edit the text file (or a copy of it) by inserting Markers to define the start and end of table rows and columns as follows:

FILE CONTENTS	COMMENTS
>>1	Table 1 start marker
--------------------------	Import as one Column
xxxxxxxxxxxxxxxxxxxxxxxxxx	First data row
xxxxxxxxxxxxxxxxxxxxxxxxxx	.
xxxxxxxxxxxxxxxxxxxxxxxxxx	.
xxxxxxxxxxxxxxxxxxxxxxxxxx	Last data row
<<1	Table 1 end marker
>>2	Table 2 start marker
------- ------- ---- ------------	Import as 4 Columns
xxxxxxx yyyyyyy zzzz wwwwwwwwwwww	First data row
yyyyyyy wwwwwwwwwwww	.
xxxxxxx yyyyyyy zzzz wwwwwwwwwwww	Last data row
<<2	Table 2 end marker
>>3	Table 3 start
-- ------ ------------ --------------	Import 4 Columns
xx yyyyyy zzzzzzzzzzzz wwwwwwwwwwwwww	First data row
xx zzzzzzzzzzzz wwwwwwwwwwwwww	.
xx zzzzzzzzzzzz wwwwwwwwwwwwww	Last data row
<<3	Table 3 end marker

Step 2: Select "Use a Marked up text file", then click convert. This will launch Microsoft Excel and copy the data into the correct cells.

If the TXL Wizard Option *Clear sheet contents before processing* check box is checked then the existing data and formatting in the output file will be cleared before processing. If the *Clear sheet contents before processing* option has been unchecked, and the Excel file name selected actually exists, the corresponding cells will be overwritten with all other cells and formatting information preserved.

In many cases, you can make the TXL software automatically determine the columns. For example we can alternatively mark table 2 as follows:

```
>>2                                   Table 2 start
===================================== Import as multiple (4)
Columns
xxxxxxx yyyyyyy zzzz     wwwwwwwwwwww . First data row
yyyyyyy                  wwwwwwwwwwww .
xxxxxxx yyyyyyy zzzz     wwwwwwwwwwww . Last data row
<<2                                   Table 2 end
```

- Sometimes you may need to mix the column markers to achieve a desired effect. For example:

```
>>1                                Table 2 start
==================== ------------- Creates 3 + 1 Columns
xxxxxxx yyyyyyy zzzz ww wwwwwwwwww . First data row
        yyyyyyy      ww wwwwwwwwww .
xxxxxxx yyyyyyy zzzz ww wwwwwwwwww . last data row
<<1                                Table 2 end
```

This will import the data into Excel in 4 columns (3 columns for XYZ data, followed one column for the "w" data in a single column. Note that the "w" data is not split into 2 columns since it has been marked with a Single column marker "-" instead of the "=".

- If you wish to import the spaces between the z's and the w's into a separate column in Excel, you should do this instead:

```
>>1                                    Table 2 start
==================== --- ------------- Creates 5 Columns
xxxxxxx yyyyyyy zzzz     ww wwwwwwwwww      . First data row
        yyyyyyy          ww wwwwwwwwww      .
xxxxxxx yyyyyyy zzzz     ww wwwwwwwwww      . last data row
<<1                                    Table 2 end
```

Sometimes the columns are not separated by any spaces; rather they are determined by exact location. For example the table:

```
>>1
-- -- ----
MM/DD/YYYY
01/09/2001
02/15/2001
02/17/2003
```

```
.
.
<<1
```

As marked above, the second data row will copy "01/", "09/" and "2001" into Excel cells A2, B2, C2, respectively, with the slash marks being included. When being used for exact location, the dash markings should be interpreted as meaning the exact number of characters represented by the dashes plus one additional character at the end of each column, with the exception of the last column. The additional character is not automatically added onto the last column.

If you wanted "01", "09" and "2001" without the slashes, do this:

```
>>1
-  -  ----
MM/DD/YYYY
01/09/2001
02/15/2001
02/17/2003
.
.
<<1
```

Here is another example to illustrate this concept. If your file markings on your input file look like this:

```
>>1
-    --   ---  ---- ----
1234/5678/1234/5678/1234
1234/5678/1234/5678/1234
1234/5678/1234/5678/1234
. .
. .
<<1
```

Your output in A1:E2 will look like this:

```
12    567   1234   5678/   1234
12    567   1234   5678/   1234
. .
```

RECIPE FILE METHOD

The file marking method is very simple and works quite well in most cases.

However, when you have a large number of files with similar table structures (but with different data) it may not be desirable to mark each text file. This is the case where you should consider the Recipe Method.

A Recipe File is a text file that contains saved table and column-parsing information. The recipe can later be applied to unmarked data files.

CREATING A RECIPE FILE

One simple way to create (or edit) a recipe is to use the Recipe Editor from the edit menu or a simple text editor such as Windows Notepad.

- Recipe File name: *.rcp where * is any legal file name

- Example Recipe file to illustrate file format

```
FILE CONTENTS                   COMMENT
[Header]                       ;Header Section ID
Product=TXL-B                  ;Do not change
FileType=RECIPE                ;Do not change
FileFormat=1.1                 ;Do not change

[Table 1]                      ;Table 1 Section ID
ReadStartLine=INCOME STATEMENT ;Find row containing "INCOME
STATEMENT"
ReadStartLineOffset=0          ;table starts at the same row
ReadEndLine=BALANCE SHEET      ;Find row containing "Balance
Sheet"
ReadEndLineOffset=-1           ;table 1 ends one row before end
key
Parsing=------------------ ==================== ;Column
Template
WriteColumn=A                   ;Start column in Excel

[Table 2]
ReadStartLine= BALANCE SHEET
ReadStartLineOffset=0
ReadEndLine= END OF REPORT
ReadEndLineOffset=-1
Parsing=-------------- ===============
WriteColumn=A
```

The above Recipe file will work correctly for the example text below:

- Example Data (for the above recipe example):

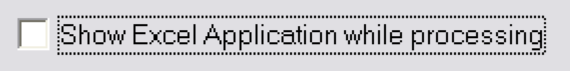
```
INCOME STATEMENT


ACCOUNT             First   Second
                    Quarter Quarter

Revenues:

Sales Products      $1,000  $1,000,000
Sales Services      $1,000  $1,000,000
Sales Misc          $1,000  $1,000,000

Expenses:

Payroll             $300    $300
Travel Expenses     $300    $300
```

```
Other               $1,000  $1,000


BALANCE SHEET

Assets:

Account        1st Q  2nd Q
xxxxxx xx      yyyyyy zzzzzz
xxxxx xxxx xxx yyyyyy zzzzzz
xxxxxxxxxx

        END OF REPORT
```

TXL WIZARD OPTIONS

Clicking the **Options** button brings up a page offering three tabbed options.

1. **General**

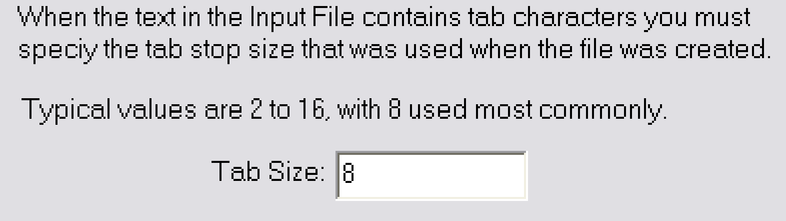


□ Show Excel Application while processing

If *Show Excel Application while* processing is checked, Excel will become briefly visible while TXL Wizard is processing

2. **Input Text File**

When working with tab delineated files, you can adjust the default stop size of the tab. Enter a number between 2 and 16 to specify tab size based on your system settings. Eight is the most common tab size.



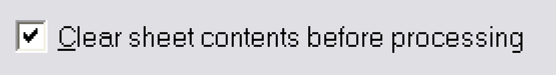
When the text in the Input File contains tab characters you must speciy the tab stop size that was used when the file was created.

Typical values are 2 to 16, with 8 used most commonly.

Tab Size: 8

If your columns are not copying to Excel correctly from a tab delineated text file, try adjusting tab size.

3. **Output Excel File**



☑ Clear sheet contents before processing

When checked, output sheet data and formatting will be cleared before processing. If unchecked the contents will not be cleared before processing, and cells will be overwritten if the input and output files share cells in common. For example, if the output file has data in cells A1:F20 before processing, and the input file has data in cells A1:F6 then after processing the data in A1:F6 will be from the input sheet, with all data in cells A7:F20 remaining unaltered.

Note that this option is only applicable to Excel files. Output CSV files will always be cleared before processing.

What You Will Need To Use TXL Wizard

The minimum hardware and software requirements for this software is shown below:

- IBM or compatible PC/AT (Pentium or higher CPU) with 16 MB of memory and one hard disk drive with 3 MB of space
- VGA or SVGA display adapter
- Microsoft Windows 95, Windows 98, Windows 2000, or Windows XP.

Installation and Un-installation

Installing

When you run the setup program to install TXL Wizard on your computer, you will be able to specify where on your hard drive to place the installation. It is preferred that you install it in the suggested directory for consistency (although not required).

Run the Setup.EXE that came with the TXL Wizard media. You may do this by clicking the *Start* button from the taskbar and select the *Run...* menu option. Then type the path and location of the Setup.EXE program. For example:

```
A:setup
```

Then press ENTER. Follow the installation instructions on the screen.

Uninstalling

It is highly suggested that you uninstall TXL Wizard before upgrading to a newer version of the product.

To uninstall TXL Wizard click the *Start* button from the taskbar and select *Settings* then *Control Panel*.

Within the control panel select the *Add/Remove Programs* icon. Double click on the TXL Wizard entry in the list box or click the *Add/Remove* button to uninstall.

All files copied during the installation will be removed (only if other programs are not currently dependent on them). Furthermore, if files have been added to the installation directory (i.e. program files you created) then the uninstall wizard will report that not all directories could be deleted. You will have to manually remove these files.

Troubleshooting

If you encounter problems your first troubleshooting steps should be to:

- Review this manual

- Frequently Asked Questions
- Bug Lists
- Latest Patches/Downloads
- E-Mail questions to TXL_Support@SoftInterface.COM

We hope that CD ease of use and this documentation will provide all the help you need.

Using TXL Wizard From the Command Line

Using The Command Line

TXL Wizard can be run without a user interface, using command line arguments, much like in the good old DOS days. Alternatively you can initiate command line execution by going to the operating system's command prompt or from within Windows from the START-RUN menu and typing it in there. Furthermore, the command line interface can be used from Batch files (files with a *.BAT extension).

Important: To use the command line successfully you must do one of the following:

1. Set the **path** in your environment variables to the location of TXL.EXE; or
2. Copy TXL.EXE to the same folder as the batch file.

Otherwise, you'll see the following error: *" TXL.EXE is not recognized as an internal or external command, operable program or batch file."*

If you installed TXL Wizard and accepted the default location, the call to the executable file will appear something like this, including the double quotes: *"C:\Program Files\Softinterface, Inc\TXL Wizard\txl.exe"*. Job specific data would then follow. For the sake of brevity, all examples below show the call to the executable file as simply **TXL.EXE.**

The syntax for the command line for specifying a *conversion task* is:

TXL.EXE /I{Input File(s)} /**O**{Output File(s)} [/**R**{Recipe File}] [/**S**] [/**G**] [/**V**] [/**?**]

Items above enclosed in square brackets “[ ]” are optional, all other ‘switches’ are required. Therefore /I, /O should always be specified on the command line. *There is one exception, however, and that is when /G (Save to original folder) is specified, the /O switch is not necessary.*

NOTE: Not specifying a recipe file implies that you have used the "File Marking Method" on the input files.

See also: All Command Line Switches

Command Line Examples

NOTE: You will need to supply the entire path to the executable file which we have condensed to **TXL.EXE** in these examples. Your command line text will likely need to include the following "*C:\Program Files\SoftInterface, Inc\TXL Wizard\txl.exe*". This path assumes you accepted the default location when installing TXL Wizard.

EXAMPLE 1: To process an input file D:\MyFiles\Input.TXT that has been marked up using the "File Marking Method" and place the results in an Excel file C:\Output\Done.XLS use the following syntax. NOTE: Once you've successfully created your command line, you can remove the /V switch.

```
TXL.EXE /I D:\MyFiles\Input.TXT  /O C:\Output\Done.XLS  /V
```

EXAMPLE 2: Process a marked up input file D:\MyFiles\Input.TXT and create output file C:\Output\Done.CSV by using the following syntax.

```
TXL.EXE /I D:\MyFiles\Input.TXT  /O C:\Output\Done.CSV  /V
```

EXAMPLE 3: Uses a recipe to run an input file named D:\MyFiles\Input.TXT and place the output in a CSV file named C:\Output\MyOutputFile.CSV. The recipe file is located at D:\MyRecipes\aRecipe.RCP

```
TXL.EXE /I D:\MyFiles\Input.TXT  /O C:\Output\MyOutputFile.CSV  /R
D:\MyRecipes\aRecipe.RCP
```

EXAMPLE 4: To process a whole folder and subfolders in D:\MyFiles\ and put the results in CSV format within the c:\Output\ folder, using recipe file F:\MyRecipes\Solution.RCP, use the following syntax:

```
TXL.EXE /I D:\MyFiles\*.TXT  /O C:\Output\*.CSV  /R F:\MyRecipes\Solution.RCP /S
```

EXAMPLE 5: Uses a marked up input file named D:\MyFiles\Input.TXT with output going to C:\Output\MyOutputFile.XLS where data in the output file is not deleted prior to processing.

```
TXL.EXE /I D:\MyFiles\Input.TXT  /O C:\Output\MyOutputFile.XLS /C
```

EXAMPLE 6: Uses a marked up, tab separated file for input named D:\MyFiles\TabbedInput.TXT with output going to C:\Output\MyOutputFile.XLS with tab size set to 10.

```
TXL.EXE  /I D:\MyFiles\TabbedInput.TXT  /O C:\Output\MyOutputFile.XLS /T10
```

EXAMPLE 7: Uses a recipe file named D:\MyRecipes\Solution.RCP for input named D:\MyFiles\Input.TXT with output going to C:\Output\MyOutputFile.XLS and displays Excel while processing.

```
TXL.EXE  /I D:\MyFiles\Input.TXT  /O C:\Output\MyOutputFile.XLS  /R
F:\MyRecipes\Solution.RCP /E
```

EXAMPLE 8: Processes multiple marked up files all in a single folder found at D:\ManyFiles\*.txt. Uses the /G switch to place output in the same location with the same file names but in .CSV format.

```
TXL.EXE  /I D:\ManyFiles\*.TXT  /O D:\ManyFiles\*.CSV /G
```

All Command Line Switches

A full description for each switch is shown below:

Switch	Meaning
/**I**{Input File(s)}	Input File(s) to be converted. A single file including its complete path can be specified for a single file conversion (i.e. "D:\MyInputFiles\Single.TXT". If specifying a single file, make sure that the /O switch also specifies a single output file (i.e. "D:\OutputFiles\TargetOutput.CSV". Alternatively, you can specify whole directories by using the wildcard syntax (i.e. "D:\MyInputFiles\*.TXT"). If specifying a whole directory, make sure the /O also specifies a whole directory with a new target file extension. (i.e. "D:\OutputFiles\*.XLS"). See also /S for subfolders.
/**O**{Output File(s)}	Output File(s) to be created when converting the input file(s). Use ".CSV" to specify comma delimited file output. Use ".XLS" to specify an Excel file output If specifying a single input file, make sure that the /O switch also specifies a single output file (i.e.

	"D:\OutputFiles\SingleImage.CSV". Alternatively, you can specify whole directories by using the wildcard syntax (i.e. "D:\OutputFiles\*.CSV"). If specifying a whole directory, make sure the /I also specifies a whole directory with an input file extension. (i.e. "D:\MyImageFiles\*.TXT") This switch is ignored if the /G Switch is used.
/**R**{Recipe File}	Recipe file to use.
/**S**	Seek out and do all files found in the sub-folders specified in the /S switch. Recursive subdirectories. You may add this switch if you are processing whole directories of files. See /S, /O and /G for more details on selecting whole directories to be converted.
/**G**	Save files to their original folders. Use this in conjunction with the /O switch to place the converted files in the same folder as the original file. Cannot be used when converting a single, specified file from within a folder. Use the /O switch instead. Files will be saved with the same name as the original but in either .XLS or .CSV format. You cannot specify a different name for the output file when using the /G switch.
/**V**	Verbose mode. Specify this switch to display a message box indicating how the conversion went.
/**?**	Bring up the help file.
/C	When /C is used, output sheet content will not be cleared before processing.
/E	Show Excel while processing
/T#	Sets the size of the tab. # indicates the size and may be a number between 2 and 16. Initial default is set to 8, the most common tab stop size.

See Also: Using the Command Line