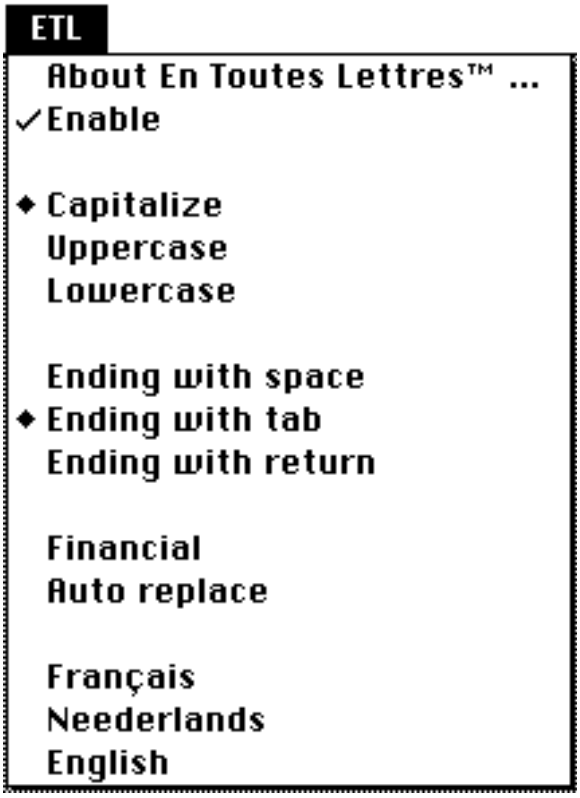

About En Toutes Lettres™…	select this item to get © information and basic help
Enable	this item must be checked to make the translation of numbers to letters available
Capitalize	select this item to capitalize the resulting text: 124 -> One Hundred Twenty Four
Uppercase	select this item to display the resulting text in uppercase: 124 -> ONE HUNDRED TWENTY FOUR
Lowercase	select this item to display the resulting text in lowercase: 124 -> one hundred twenty four
Ending with space	select this item if you want the converted text being ended by a space
Ending with tab	select this item if you want the converted text being ended by a tabulation
Ending with return	select this item if you want the converted text being ended by a carriage return
Financial	select this item to add the currency sign at the end of the text
Auto replace	select this item to make ETL delete the original number you entered

Holding down the option key while selecting a translator name forces ETL to use the clipboard content instead of the characters entered with the keyboard. This of course doesn't work with some applications …

Programmer Interface:

If you are a developer, you may be interested by the fact that you can call En Toutes Lettres™ translators from within your application or any other piece of code.
This is achieve thru the Gestalt selector 'ETLE'. The `response` parameter will be filled with a pointer on a variable length `GestaltInfo` record. (See next page).
You will find in the TransInfo array, the pointers to the translators themselves.
The translators may be identified by name or by ID. The IDs are defined in the International Utilities Manager of Inside Mac. (US = 0, France = 1, etc)
A translator is a one routine stand-alone code resource of type 'LGDF'.

```
typedef     pascal void (*LgdfProcPtr)(char *text,char *result)   ;
```

The ETL code calls the translator passing in `text` the number in ASCII to be converted and in result a pointer on a buffer of 255 characters - C string - to be filled by the LGDF code.

```
#include     <GestaltEqu.h>
typedef      pascal void (*LgdfProcPtr)(char *text,char *result)  ;
typedef struct TransInfo  {
       short        fIntlID      ;
       Str32        fName        ;
       LgdfProcPtr  fProc        ;
}      TransInfo    ;
typedef struct GestaltInfo {
       short        fVersion           ;
       short        fNTranslators      ;
       short        fCurFormat         ;
       short        fCurEnding         ;
       Boolean      fDoFinancial       ;
       Boolean      fDoAutoReplace     ;
       Boolean      fEnable            ;
       TransInfo    fTranslators[1]    ;
}      GestaltInfo;
typedef      GestaltInfo  *GestaltInfoPtr    ;
main() {
       GestaltInfo  *response    ;

             Gestalt('ETLE',(long *)&response);
}
```