# Changing locale-specific information by loading the appropriate DLL

In Symbian OS locale-specific information is built as a DLL. The user can change the locale information related to a specific country by loading the appropriate DLL.

The convention followed in naming the DLL is as follows:

**elocl.language_index** - The list of languages supported by Symbian OS is enumerated in TLanguage. Each value in the TLanguage enumeration uniquely identifies a language. TLanguage enumeration can be found in the e32Const.h header file.

**Examples:**

ELangEnglish and DLL name is elocl.01.

ELangFrench and DLL name is elocl.02.

ELangGerman and DLL name is elocl.03.

ELangFinnish and DLL name is elocl.09.

Symbian OS provides the TLocale class to set the system-wide locale settings and to retrieve system-wide locale setting. The TLocale class provides methods for setting the following information: calendar settings, country code, currency format, date and time formatting, numeric values, time zone information, and units of distance.

In Symbian OS most applications do not need to change the locale settings; they just use the system setting instead.

The user can load the appropriate locale-specific DLL to fetch the locale data. The function given below is a way to fetch the locale-specific information.

```
void LoadLanguageDLL()
{
  //Finnish
  _LIT(KDllName, "elocl.09");
  RLoader loader;
  TInt r = loader.Connect();
  if(KErrNone == r)
  {
    //Load the language variant DLL
    TInt size = KNumLocaleExports * sizeof(TLibraryFunction);
    TPtr8 functionListBuf((TUint8*) data, size, size);
    r = loader.SendReceive(ELoadLocale, TIpcArgs(0, KDllName, &functionListBuf ) );
    if(KErrNone == r)
    {
     ((TLibFn)data[FnLocaleData])(&iLocale);
     retVal = (TText*) aLocale.Ptr();
    }
    loader.Close();
  }
}
```

Setting the locale by writing a program in POSIX is easier compared to the locale settings in Symbian OS. In POSIX the user does not need to be aware of the locale information, whereas in Symbian OS the user should be aware of it.

Example: The code below sets the monetary information of Finland using POSIX:

```
#include <stdio.h>
#include <locale.h>
int main()
{
  //before fetching the monetray information it is assumed that locale is set to finnish
  struct lconv* localeinfo = localeconv();
  return 0;
}
```

The lconv structure has the complete monetary information.

The code below sets the monetary information of Finland using Symbian C++:

```
#include <e32std.h>
void doExampleL()
{
  TLocale locale;
  locale. SetCurrencySymbolPosition(ELocaleAfter);
  locale. SetCurrencySpaceBetween(ETrue);
  locale. SetCurrencyDecimalPlaces(2);
  locale. SetCurrencyTriadsAllowed(ETrue);
  locale. SetNegativeCurrencyFormat(TLocale::ELeadingMinusSign);
  //similarly other monetary information should be set individually.
}
```


(C) Copyright Nokia 2013. All rights reserved.

 (C) Copyright Nokia 2013. All rights reserved.