Conversion from one native character set to another native character set

The following code snippet illustrates the steps to be followed to convert from one native character set to another native character set.The code converts a string from the ISO-8859-1 character set to the ISO-8859-2 character set.

Steps:

- Specify the non-Unicode character set being converted to Unicode. In this case it is ISO-8859-1.
- Convert the text from the native character set to Unicode.
- Specify the non-Unicode character set being converted from Unicode. In this case it is ISO-8859-2
- Convert the text from Unicode to the native character set

```
#include <e32base.h>
#include <e32cons.h>
#include <charconv.h>
#include <f32file.h>
#include <string.h>
#include <stdlib.h>
LIT8(KNativeText, "abcd");

LOCAL_C int ConversionFunction()
{
   char *inbuf = "abcd";
   char *outbuf = NULL;
   int inbytes,outbytes;
   inbytes = outbytes = 4;

   TInt retVal = KErrNone;
   outbuf = (char*) malloc(outbytes);
   RFs fileSession;
   User::LeaveIfError(fileSession.Connect());
   CleanupClosePushL(fileSession);

  //Allocates and constructs a CCnvCharacterSetConverter object
  CnvCharacterSetConverter* conv = CCnvCharacterSetConverter::NewLC() ;
  TPtrC8 remainderOfForeignText((const TText8*) inbuf,inbytes);
  TBuf16<256> UnicodeText;
  TBuf8<256> outputBuffer;
  TInt numberOfUnconvertibleCharacters = 0;
  TInt indexOfFirstByteOfFirstUnconvertibleCharacter = 0;
  //Specifies the character set(ISO-8859-1) to convert to UNICODE
  CCnvCharacterSetConverter::TAvailability avail = conv->PrepareToConvertToOrFromL (KCharacterSetIdentifierIso88591, fileSession);
 if(CCnvCharacterSetConverter::ENotAvailable == avail)
 {
   CleanupStack::PopAndDestroy(2);  //conv, fileSession
   return KErrGeneral;
 }
 //Convert text encoded in the ISO-8859-1 character set into the Unicode character set (UCS-2).
 retVal = conv->ConvertToUnicode(UnicodeText, remainderOfForeignText, numberOfUnconvertibleCharacters, indexOfFirstByteOfFirstUnconv
 if(retVal < 0 && (retVal != CCnvCharacterSetConverter::EErrorIllFormedInput))
 {
  CleanupStack::PopAndDestroy(2);  //conv, fileSession
  return retVal;
 }
 //Specifies the character set(ISO-8859-2) to convert from UNICODE
 avail = conv->PrepareToConvertToOrFromL(KCharacterSetIdentifierIso88592, fileSession);
 if(CCnvCharacterSetConverter::ENotAvailable == avail)
 {
   CleanupStack::PopAndDestroy(2);  //conv, fileSession
   return KErrGeneral;
 }
 //Convert text encoded in the Unicode character set (UCS-2) to ISO-8859-2
 retVal = conv->ConvertFromUnicode(outputBuffer,UnicodeText, numberOfUnconvertibleCharacters, indexOfFirstByteOfFirstUnconvertibleCh
 if(retVal < 0 && (retVal != CCnvCharacterSetConverter::EErrorIllFormedInput))
 {
   CleanupStack::PopAndDestroy(2);  //conv, fileSession
   return retVal;
 }
 TInt outputbufferLength = outputBuffer.Length();
 strncpy(outbuf, (const char*) outputBuffer.Ptr(),outputbufferLength);
 outbuf = outbuf + outputbufferLength;
 CleanupStack::PopAndDestroy(2); //conv, fileSession
 return retVal;
}
```


(C) Copyright Nokia 2013. All rights reserved.

 (C) Copyright Nokia 2013. All rights reserved.