



ID	CS001412	Creation date	June 9, 2009
Platform	S60 3rd Edition S60 5th Edition	Tested on devices	Nokia 5800 XpressMusic
Category	Symbian C++	Subcategory	Location Based Services

Keywords (APIs, classes, methods, functions): TPositionInfoBase, TPosition

Reviewer Approved



Overview

This snippet demonstrates two ways of using TPositionInfoBase:

- getting the TPosition class from TPositionInfoBase
- showing TPosition in textual format

Note: The Location capability is included in the self-signing capabilities of S60 3rd Edition, Feature Pack 2 and newer platforms.

MMP file

The following libraries and capabilities are required:

```
CAPABILITY    Location
LIBRARY       lbs.lib
```

Header

```
#include <LbsPositionInfo.h>
#include <LbsPosition.h>
```

CS001412_-_Getting_position_data_from_TPositionInfoBase

```
// Degrees sign delimiter used in formatting methods
_LIT(KDelimDegree, "\xb0"); // "°" symbol

// Dot delimiter used in formatting methods
_LIT(KDelimDot, "\x2e"); // "." symbol

// Plus sign delimiter used in formatting methods
_LIT(KDelimPlus, "\x2b"); // "+" symbol

// Minus sign delimiter used in formatting methods
_LIT(KDelimMinus, "\x2d"); // "-" symbol

// Quotation sign delimiter used in formatting methods
_LIT(KDelimQuot, "\x22"); // "\"" symbol

// Apostrophe sign delimiter used in formatting methods
_LIT(KApostrophe, "\x27"); // "'" symbol

const TInt KDegreeLength = 19;
```

Source

```
void CMyClass::PositionUpdated(TPositionInfoBase& aPosInfo)
{
    // Check if position information class type is TPositionInfo
    if (aPosInfo.PositionClassType() & EPositionInfoClass)
    {
        // Cast the TPositionInfoBase object to TPositionInfo
        TPositionInfo* posInfo = static_cast<TPositionInfo*>(&aPosInfo);

        // Get position
        TPosition position;
        posInfo->GetPosition(position);

        // Convert positions to the descriptors
        TBuf<KDegreeLength> latitudeDegr;
        GetDegreesString(position.Latitude(), latitudeDegr);

        // Convert positions to the descriptors
        TBuf<KDegreeLength> longitudeDegr;
        GetDegreesString(position.Longitude(), longitudeDegr);
    }
}

void CMyClass::GetDegreesString(
    const TReal64& aDegrees, TBuf<KDegreeLength>& aDegreesString) const
{
    const TReal KSecondsInMinute = 60.0;
    const TInt KNumWidth = 3;

    // If the aDegree is a proper number
    if ( !Math::IsNaN(aDegrees) )
    {
        // Integer part of the degrees
        TInt intDegrees = static_cast<TInt>(aDegrees);
```

CS001412_-_Getting_position_data_from_TPositionInfoBase

```
// Positive float of the degrees
TReal64 realDegrees = aDegrees;

// Convert to positive values
if ( intDegrees < 0 )
{
    intDegrees = -intDegrees;
    realDegrees = -realDegrees;
}

// Minutes
TReal64 realMinutes = (realDegrees - intDegrees) * KSecondsInMinute;

// Integer part of the minutes
TInt intMinutes = static_cast<TInt>(realMinutes);

// Seconds
TReal64 realSeconds = (realMinutes - intMinutes) * KSecondsInMinute;
TInt intSeconds = static_cast<TInt>((realMinutes - intMinutes) * KSecondsInMinute);

// Check the sign of the result
if ( aDegrees >= 0 )
{
    aDegreesString.Append(KDelimPlus);
}
else
{
    aDegreesString.Append(KDelimMinus);
}

// Add the degrees
TInt64 value = intDegrees;
aDegreesString.AppendNum(value);

// Add the separator
aDegreesString.Append(KDelimDegree);

// Add the minutes
value = intMinutes;
aDegreesString.AppendNum(value);

// Add the separator
aDegreesString.Append(KApostrophe);

// Add the seconds
value = intSeconds;
aDegreesString.AppendNum(value);

// Add the separator
aDegreesString.Append(KDelimQuot);

// Add the separator
aDegreesString.Append(KDelimDot);

// Get six last digits
realSeconds -= intSeconds;
realSeconds *= 1000;

// Add the seconds
aDegreesString.AppendNumFixedWidth(static_cast<TInt>(realSeconds),
EDecimal, KNumWidth);
```

```
}  
}
```

Postconditions

Position data is read from `TPositionInfoBase`.

See also

- [CS001376 - Retrieving GPS location](#)
- [S60 Platform Location Example](#)