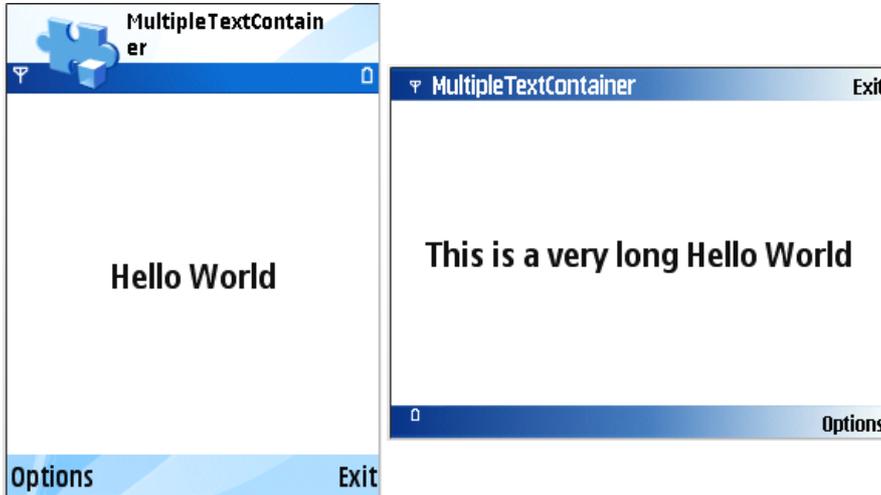


How to define and choose scalable text

This article explains how to define and choose scalable text in S60 2nd FP3 or newer. Scalable text allows the developers to display different variant of the "same text" depending on the screen resolution. For example, a simple label may display more text in landscape mode, compared to the portrait mode.

The two pictures below show an example of scalable text. The shorter text is displayed in the portrait mode because the space is limited. The longer one is displayed in the landscape mode because there is more space.



Defining scalable text

A Scalable text is normally defined in the resource file (.rss) or localization file (.loc). Each variant is separated with 0x0001. For example:

```
#define Str_Text_Example1 "This is a very long Hello World"<0x0001>"Hello World"
```

The example above defines a scalable text that contains two variant, i.e. "This is a very long Hello World" and "Hello World".

There can be two or more variants defined in the scalable text. For example:

```
"This is a very long Hello World"<0x0001>"Long Hello World"<0x0001>"Hello World".
```

The scalable text above contains three variants.

Using scalable text in control

Using scalable text in a control, such as label or list box, is straightforward. All S60 controls are aware of scalable text. They will choose the longest variant that can fit on the selected area.

Example:

```
HBufC* labelString = StringLoader::LoadLC(R_TEXT_EXAMPLE1);
iLabel1->SetTextL(*labelString);
CleanupStack::PopAndDestroy(); // labelString
```

Choosing scalable text manually

If the scalable text is going to be displayed not in S60 control, the variant has to be chosen manually. For example, the method

`CFbsBitGc::DrawText()` is not aware of scalable text.

In this case, choosing the variant has to be done manually using `AknUtils::ChooseScalableText()`. This method chooses the variant based on two parameters, i.e.:

- The font used to display the text.
- The maximum width of the area (in pixel) where the text is going to be displayed.

Example:

```
CWindowGc& gc = SystemGc();
gc.Clear( aRect );

// Select the font.
const CFont* font = CCoeEnv::Static()->NormalFont();
gc.UseFont(font);

// Load the scalable text.
HBufC* labelString = StringLoader::LoadLC(R_TEXT_EXAMPLE1);

// Choose the best variant based on the string, font and pixel width.
TPtrC outputText = AknTextUtils::ChooseScalableText(
    *labelString, *font, Rect().Width());

// Draw the text to the screen.
gc.DrawText(outputText, TPoint(10, 20));

CleanupStack::PopAndDestroy(); // labelString
```

Internal links

- [Scalable UI](#)
- [Change screen orientation of UI application](#)