Archived:Controlling the keyboard repeat rate

Archived: This article is archived because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

Overview

This article shows how to change the keyboard repeat rate on Symbian devices using Symbian C++ code in Qt applications.

Description

Keyboard repeat rate in Symbian devices can be controlled using functions from the `RWsSession` class:

```
void GetKeyboardRepeatRate( TTimeIntervalMicroSeconds32 &aInitialTime,
                            TTimeIntervalMicroSeconds32 &aTime );

TInt SetKeyboardRepeatRate( const TTimeIntervalMicroSeconds32 &aInitialTime,
                            const TTimeIntervalMicroSeconds32 &aTime );
```

Because the repeat rate is a global setting, the WriteDeviceData capability is required for `SetKeyboardRepeatRate()`. Also, applications modifying the repeat rate should reset back to the default values whenever switched to background or terminated.

Solution

```
CCoeEnv::Static()->WsSession().SetKeyboardRepeatRate( 200000, 50000 );
```

Virtual mouse cursor in Qt applications

On non-touch Symbian devices, the following code will enable a virtual mouse cursor, controlled by the navigation keys:

```
QApplication::setNavigationMode( Qt::NavigationModeCursorAuto );
```

The speed of the cursor movement can be controlled by changing the keyboard repeat rate as above.

For example, to set the initial delay to 200 milliseconds, and subsequent repeats to 50 milliseconds:

```
#ifdef Q_OS_SYMBIAN
#include <coemain.h>
#include <w32std.h>
#endif

int main( int argc, char *argv[] )
{
  QApplication a( argc, argv );
  ...
#ifdef Q_OS_SYMBIAN

  QApplication::setNavigationMode( Qt::NavigationModeCursorAuto );

  CCoeEnv::Static()->WsSession().SetKeyboardRepeatRate( 200000, 50000 );

#endif
  ...
}
```

In the **.pro** file, add the `WriteDeviceData` capability and the required libraries:

```
symbian {
  TARGET.CAPABILITY += WriteDeviceData
  LIBS += -lcone -lws32
}
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.