Changes in Nokia Belle and Qt 4.x

This article describes a set of changes introduced by the Nokia Nokia Belle release and Qt 4.x, which may impact some 3rd party applications. The aim of this article is to support developers in their review of application's behavior on Nokia Belle and their corrective actions.

Introduction

As with any new OS release, Nokia Belle's new features and especially its significant UI changes, have required that a number of APIs be changed, in some cases these changes resulting in binary compatibility breaks which affect existing applications. Aside from these approved API breaks, we are also documenting some accidental breaks (bugs!) which will however be fixed in the PR 1.1 release for the Nokia Belle devices, which is the same release which will be available soon for the Symbian^3 and Symbian Anna devices.

The changes are affecting various Symbian OS components but there are also changes in the Qt framework itself. This article covers all significant changes in Belle, Belle FP1, Qt 4.7.4 and Qt 4.8.

A note on firmware versions: Nokia 700, Nokia 701 and Nokia 603 were released with what we call Nokia Belle PR 1.0 firmware (some variants with PR 1.01). The next firmware release for these devices, PR 1.1, is the common Nokia Belle release that was distributed as a firmware update to all Nokia Belle compatible devices (including Nokia N8 & friends). So while "PR" is a term reserved for firmware, it does help us in this case to name two successive Nokia Belle releases as well.

In the article below you will see references to PR 1.1, as in "expected to be fixed in PR 1.1". You should read this as: the problem may still affect a few of the devices which have not been updated to PR 1.1 but the main release, the one covering the entire Symbian^3 set of devices, has the issue fixed already.

Please note that this page is currently in draft mode and may be updated frequently. Make sure to add it to your watch list (eye icon) to ensure that you're receiving update notifications.

Changes in Nokia Belle

Avkon UI changes

Background:

The new Belle UI has impact in the way screen real estate is used by applications. The cumulated size of status pane and control pane result now in a bigger client area in portrait more (wider but less tall in landscape mode). Toolbars are merged with the control pane ... Softkeys are now icons, with the conversion being done in most cases automatically but also with an option to customize the icons shown.

Impact for affected applications:

Applications designed with S60 5th edition layout in mind (or even older) may end-up not drawing the full client area, or not utilizing the available real estate to its maximum. Applications which are relying on hardcoded assumptions about the layout and available real estate may end-up crashing in various places inside own logic. Applications attempting to read/change softkeys dynamically may also be affected Text toolbar buttons are no longer supported, will show-up as “...” in Belle toolbar

Solution/Workaround

Applications must be tested on Belle devices and changes must be considered in case by case bases. A porting guide is available in Nokia Belle Developer Library. Developers starting new applications are advised to build their UI using the Qt Quick Components, which offers the same Belle look and feel in a more productive and future proof package.

Remember that Nokia Belle [FP1] is available on various devices, with differences not only in resolution (360x640 vs 640x480) but also in DPI (Nokia N8 vs Nokia E7 or Nokia E6). For your application to easily and seamlessly support all these devices you should review the Scalability topic in the SDK documentation and device specific guidance, such as Prepare your application for Nokia E6.

New memory allocator implementation

Background:

The previous heap allocator uses a first-fit allocation algorithm, where free cells are linked in a chain and when an allocation request is issued, the list will be traversed until a large enough free cell is found. When freeing cells, consecutive free areas will be combined. The new memory allocator called the High Performance Allocator, is a combination of three allocation algorithms, namely Doug Lea allocator, Slab allocator and paged allocator.

Impact for affected applications:

The solution is binary compatible with respect to the public/SDK APIs but the internals of the API have been changed significantly and any applications using non-public interfaces (RHeap class changed into a base class, instantiating it is no longer valid) RHeap has been modifled as follows:

- Several public inline methods have been moved to RAllocator or removed.
- RAllocator must be dervied as most of protected RHeap methods are removed. If application derived RHeap and calls these, break occurs.
- RHeap::__DbgGetAllocFail() and User::__DbgGetAllocFail() functions added allowing to access that variable.

(C) Copyright Nokia 2013. All rights reserved.

- The value of enums ECellAlignment, EFreeCellSize and EAllocCellSize changed as the implementation has changed in the new heap allocator.

Also new allocator consumes more memory for storing internal metadata and this may become visible in borderline cases (apps close to using 99,99% of their max heap with the old allocator), when apps will see memory allocation failures when run under the new memory allocator.

Solution/Workaround

Application launcher code was changed so that initial max heap size (as defined in the MMP file) is incremented automatically up to the next 1MB, in order to allow the increased memory consumption.with as Some applications may be still affected by the changes if they use directly the changed the internal APIs and would require code change to remove the source of the problem.

Changes in Profiles Engine Wrapper API

Background:

In an effort to make the Settings application more user friendly and to simplify the amount of options provided, the Outdoor and Pager profile were removed from both UI and the API implementation.

Impact for affected applications:

Applications which were expecting that these two profiles exist and attempted to set them via the MProEngEngine() interface will fail.

Solution/Workaround

No solution exists. Applications must avoid setting these undefined profiles.

Printing Support API is no longer available

Background:

The PrintingServices collection (sf\os\graphics\printingservices) provides a framework for providing printer drivers. However, this framework has not been used in any device and has not been actively developed or maintained so this collection is now being deprecated with the intention to remove it from future releases. It is important to note, however, that the Graphics Device Interface API continues to provide printing support to printer devices. Printing is treated as drawing to a graphics context, with the printer represented by a specialized graphics device.

The PrintingSupport component (sf\mw\appsupport\printingsupport) provides the framework for printing to printer devices. It also supports a framework for print previews. The implementation supports printers using PictBridge (USB), Bluetooth, DPOF and UPnP (WLAN). The printing application can be started from any application using AIW (Application Interworking) plug-in.

Impact for affected applications:

A few applications may be affected, but they are likely enterprise apps developed in partnership with Nokia. Public notification was published at http://www.developer.nokia.com/Community/Discussion/showthread.php?209951-Removal-of-PrintingServices-component-call-for-comments

Solution/Workaround

This is a planned break.

Screensaver API capability requirements changed

Background:

New requirement introduced in Symbian Anna asks for the user to be able to place emergency calls from the built-in screensaver (e.g. BigClock). The new feature was implemented in the ScreenSaver app as part of it the set of capabilities required by the app was increased by adding the NetworkControl capability.

Impact for affected applications:

All the ScreenSaver plug-ins are now required to have the NetworkControl capability in order to be loaded. The issue it is likely to affect affects all published screensavers plug-ins. Since the capability is from Restricted range developers can’t simply add it to the project since Nokia Publish cannot sign an application having this capability.

Solution/Workaround

The error was reported against the ScreenSaver application and the designed solution was contested. Now the emergency dial code was moved in Big Screen and ScreenSaver is NetworkControl free. The fix is due in PR 1.1 release of Nokia Belle. All screensaver apps are affected until this fix is deployed.

Remote Storage FW MountMan API removed

Background:

This API provided control functions that can be used to mount and unmount remote storages, make queries about active mounts, their connection properties (server name etc.) and connection state (connected/disconnected). It can also be used to set the connection state of some active mount. The removal is part of the planned restructuring of the remotestorage package

Impact for affected applications:

(C) Copyright Nokia 2013. All rights reserved.

This API has platform classification which means it has not been documented as part of the public developer offering. If there are however third party applications using it, they will be affected by this change.

Solution/Workaround

The affected applications will have to drop whatever feature they have implemented which has a dependency on this API.

Toolbar API changes cause redraw issues

Background:

Side effect of the Belle UI related changes in Avkon.

Impact for affected applications:

Due to the way the size and position of controls are calculated, in some cases the applications may see that their toolbar buttons are not rendered in the new Belle toolbar (between the softkeys) until the softkeys themselves are being repainted as a result of external event (user tapping the button).

Solution/Workaround

Fix expected in PR 1.1 of Nokia Belle

WSERV 86 panic

Background:

There is a behavioral change in CWsWindowBase::CommandL() and EWsWinOpEnableAdvancedPointers. This is caused due to a change to the window server client library where it panics a client if it tries to enable advanced pointer events after the window has been activated.

Impact for affected applications:

Application attempting to use multi-touch on Symbian^3 devices may be affected by this defect.

Solution/Workaround

Fix expected in PR 1.1 of Nokia Belle.

Phonebook Fetch UI API break

Background:

Root-cause unspecified

Impact for affected applications:

The implementation of CPbkMultipleEntryFetchDlg is broken and the contact selection dialog cannot be launched.

Solution/Workaround

Fix expected in PR 1.1 of Nokia Belle.

SIP Providers API is deprecated

Background:

SIPProviders API is a Symbian OS component that was prototyped to support SIP high level APIs using the comms framework. S60 provides a more complete implementation for this functionality in the IPAppServices API and as a result the SIPProviders API is now marked as deprecated and will be removed in future releases of the platform.

Impact for affected applications:

Should there be any application with dependencies on this component, such application would have to migrate to using the IPpAppServices API.

Solution/Workaround

This is a planned change. Going forward developers are encouraged to only use IPAppServices APIs if they are looking for High Level SIP APIs.

Connection Settings API

Background:

Support for ECmCellularDataUsageConfirm value and iUsageOfWlan field is removed from struct TCmGenConnSettings. The structure is defined in cmgenconnsettings.h and it's used in cmmanager.h by RCmManager::ReadGenConnSettingsL() and RCmManager::WriteGenConnSettingsL(). The following values/fields are no longer supported: • ECmCellularDataUsageConfirm value in TCmCellularDataUsage • iUsageOfWlan field in TCmGenConnSettings • TCmUsageOfWlan enum The reason for these changes are new cellular data and WLAN radio on of toggles for Nokia Belle, thus the Connection Settings API must reflect this change. For backwards compatibility the values are kept in the interface but shall not be used anymore.

(C) Copyright Nokia 2013. All rights reserved.

Impact for affected applications:

This CR is binary and source compatible but introduces functional break as described above. Users of ReadGenConnSettingsL() should check their code. ECmCellularDataUsageConfirm is no longer returned by CMManager, so some dead code can be removed by clients. Field iUsageOfWlan is not used outside ipconnmgmt package. WriteGenConnSettingsL() is not used outside ipconnmgt package. This change is binary and source compatible but introduces functional break as described above.

Solution/Workaround

This is a planned change. Corrective actions must be taken on application side.

Changes in Nokia Belle FP1

Deprecation of Access Point Engine API

Background:

Access Point Engine API was deprecated since S60 3.1 and it cannot be guaranteed to work post Belle FP1 update.

Impact for affected applications:

Any application which had dependency on Access point Engine API will cease to work as secure Wi-Fi access point cannot be created.

Solution/Workaround

Connection settings API should be used to handle the access points in place of Access Point Engine API.

TRAP Harness Addition

Background:

'Due to touch improvements in Belle FP1, additional TRAP harness is added. The addition is part of the changes made to the window server as well as to the control environment framework. In Belle FP1, this adds additional check for applications' Cleanup stack and thereby triggers the failure during application lifetime.

Impact for affected applications:

In Nokia Belle FP1, applications could crash with the panic E32USER-CBase 71 when any additional items are left on Cleanup stack.

Solution/Workaround

Applications have to be stricter in the usage of Cleanup stack and have to ensure that any item that is pushed onto the cleanup stack inside a TRAP harness is popped out before leaving the TRAP harness.

QSystemNetworkInfo::networkName() requires ReadDeviceData and WriteDeviceData capabilities

Background:

While the System Info API from the Qt Mobility module is documented to require both *ReadDeviceData* and *WriteDeviceData* capabilities, not all the method calls from this API would actually require it and up until Nokia Belle Feature Pack 1 apps could call e.g. the *networkName()* method successfully without having these apabilities. Due to changes in Nokia Belle Feature Pack 1's middleware the API does require now both ReadDeviceData and WriteDeviceData capabilities in order to be executed successfully.

Impact for affected applications:

Those applications which are calling the QSystemNetworkInfo::networkName() API without having the ReadDeviceData capability will malfunction/panic while having ReadDeviceData alone is not enough to actually read the data.

Solution/Workaround:"

The required capabilities must be added to the projects' *.pro file and a new version of the application published.

Pop-up lists and single tap handling

Background:

Depending on the parameters used in constructing the pop-up list controls based on Avkon's CAknPopupList and related classes, in Nokia Belle FP1 you may notice that the popup elements are no longer single tap enabled and it may now be required for the user to tap twice on the desired selection in order to trigger the popup closing.

Impact for affected applications:

User experience may be slightly degraded although the application remains functional.

 (C) Copyright Nokia 2013. All rights reserved.

Solution/Workaround:

See Custom Avkon pup-up list for an example on how a subclass of CAknPopupList can be used to control the way the pop-up will handle the key and tap events, in order to achieve the desired single tap selection effect.

Nokia Browser 8.2 and WAP/WML support

Background:

Nokia Belle FP1 brings a new version of the Nokia Browser for Symbian, version 8.2. This browser release bring new HTML5 features but at the same time drops support for the legacy WAP 2.0 standard. For a full list of supported standards please see the Standards support matrix for Nokia mobile browsers in the recently updated Web Developer Library.

Impact for affected applications:

Mobile WAP sites as well as hybrid applications which are relying on WMLand WMLScript to be parsed via the Browser Control API, will no longer be rendered.

Solution/Workaround:

These sites and applications must migrate to using HTML.

Changes in Nokia Belle FP2

Browser Control API

Background:

Implementation of CBrCtlInterface::PageInfoLC() has a bug in it. Calling it with parameters TBrCtlDefs::EPageInfoFocusedNodeUrl or TBrCtlDefs::EPageInfoContent will return null. Moreover as the function is of LC type, null is pushed to cleanupstack.

Impact for affected applications:

When querying TBrCtlDefs::EPageInfoFocusedNodeUrl or TBrCtlDefs::EPageInfoContent, data is not returned. Crash results.

Solution/Workaround:

None

Changes in Qt 4.7.4

The Qt 4.7.4 bundle (including not only Qt but also WebKit, Qt Quick 1.1, Qt Mobility 1.2 and Qt Quick Components 1.1) is also part of the Nokia Belle offering. Applications using its components may also experience some behavior changes, as described below.

(See also List of compatibility issues in the Qt 4.7.4 Release Bundle)

Changes in Qt Mobility's QML bindings

Background:

The QML bindings released with Qt Mobility 1.1.3 were never considered final, and they should have been in fact removed or clearly marked as experimental. In Qt Mobility 1.2 these bindings have been touched, either for the purpose of fixing implementation bugs or in order to finalise the API design QML Maps was re-implemented, developers should import version 1.2 (version 1.1 can now be imported but has 1.2 implementation underneath) Signals have been removed from QML DeviceInfo, QML NetworkInfo, QML DocumentGalleryModel The BatteryLevel property has been removed from QSystemInfo See more detailed info about the changes at
http://developer.qt.nokia.com/wiki/Qt_4_7_4_Compatibility_Issues#660af2586b3a959587a018029d083504

Impact for affected applications:

The change in QML Maps is likely the one with the more significant impact when considering the number of affected apps, but hopefully the impact will not be as significant. The rest of the changes impact less significant features, likely used only be specialised utility apps, but in this case the impact is critical (app stops working).

Solution/Workaround

This is a planned change. With Qt Mobility 1.2.1 an effort was made to attenuate the impact of the changes compared with what was proposed in 1.2.0 but no further action will be taken. Affected applications must be modified to remove the references to the non-existing components.

Strict QML import version control

Background:

Import version number must be correct. Previously a higher version number statement was ignored, and the latest version actually available was used. Example: In Qt 4.7.3: "import QtMultimediaKit 1.2" is accepted, but the latest version (1.1) is used In Qt 4.7.4: "import QtMultimediaKit 1.2" produces an

(C) Copyright Nokia 2013. All rights reserved.

error “module “QtMultimediaKit” version 1.2 is not installed import QtMultimediaKit 1.2”

Impact for affected applications:

It is expected that applications have correct versions in use. If any applications are affected this is an application level error.

Solution/Workaround

This is a planned change. Corrective actions must be taken on application side.

Qt Quick Components import changes in Symbian

Background:

“import Qt.labs.components.native 1.0” will not work anymore after components 1.1 has been installed on device. 1.1 is pre-installed on newer Belle releases. Use “import com.nokia.symbian 1.0” instead.

Impact for affected applications:

This should not be an issue for 3rd party apps since “com.nokia.symbian 1.0” was already the recommended/used import.

Solution/Workaround

This is a planned change. Corrective actions must be taken on application side.

Orientation detection in QWidget –based Symbian application

Background:

Using QApplication::desktop()->availableGeometry() during the resize event to detect orientation will produce wrong results. The recommended method is to connect to QApplication::desktop()‘s workAreaResized(int) signal and call QApplication::desktop()->availableGeometry() in the connected slot function.

Impact for affected applications:

This break has been introduced between Qt 4.6 and 4.7, so it already applies to Symbian Anna.

Solution/Workaround

QWidget is deprecated for mobile use, developers are strongly advised to rewrite their applications using Qt Quick.

Changes in Qt 4.8.x

Qt Quick Components Window anchoring changes in Symbian

Background:

Any child items should not be anchored to the Window root item directly, but refer to parent instead:

Wrong (child anchoring refers to the Window id):

```
Window {
    id: root
    Statusbar {
        anchors.top: root.top
    }
    ToolBar {
        anchors.bottom: root.bottom
    }
}
```

Correct (child anchoring refers to the parent):

```
Window {
    id: root
    Statusbar {
        anchors.top: parent.top
    }
    ToolBar {
        anchors.bottom: parent.bottom
    }
}
```

Impact for affected applications:

Applications must obey this rule already now in order to ensure that their application is future proof..

Solution/Workaround

(C) Copyright Nokia 2013. All rights reserved.

This is a planned change. Corrective actions must be taken on application side.

Qt 4.8.x drops dependency on Open C

Background:

When porting Qt to the Symbian platform it was decided that it would use the OpenC (PIPS & Co.) component to speed up the porting effort. This decision has had a cost in performance and stability, in particular in the networking and file system access areas. In the Qt 4.8 release for Symbian the Open C dependency is dropped and the corresponding backends (including networking) have been reimplemented using Symbian's native APIs.

Impact for affected applications:

Applications using the APIs from the QtNetwork module should not see a negative impact from this changes. However, applications developed with the early Qt releases (4.6) and still using the code exemplified in sym_iap_util.h will malfunction since the call to *setdefaultif()* has no effect on the Qt application any more.

Solution/Workaround:

Applications must remove the legacy implementation and use the Bearer Management API instead.

Key press event handling

Background:

As you can see from QTBUG-17545 the key handling implementation is incorrect before Qt 4.8 and QKeyEvent::text() returns an empty string when the Enter key is pressed.

Impact for affected applications:

Depending on application's internals, this change on API return may not be handled properly, causing the application to misbehave/crash.

Solution/Workaround:

Since the application is expected to run correctly on both 4.7.x and Qt 4.8, developers should be aware of this data compatibility change and ensure that the situation is correctly handled in the context.

Performance considerations on flushing a file stream

Background:

Several C++ file operations are supposed to flush file stream handles. In previous release (pre-Qt 4.8) stream flush did not flush all the way to disk. This may have caused incorrect behavior if several processes were reading and writing to same files. In Qt 4.8 this has been fixed. The file operations that are supposed to flush file, do it. This however may have a performance penalty on Symbian devices. If application does a lot of file flushing, it may be a lot slower in Nokia Belle Feature Pack 1/Qt 4.8.

Impact for affected applications:

Applications may appear to be sluggish when run on Qt 4.8

Solution/Workaround:

Avoid unintentional or redundant flush requests.

For instance, look at the following code snippet.

```
QString fname("testfile.txt");
QFile file(fname);
if (!file.open(QIODevice::WriteOnly))
    return;
QTextStream fileStream(&file);
for (int i=0; i < 1000; i++) {
    fileStream << "Test file content line " << i << endl;
}
file.close();
```

The C++ manipulator function “endl” causes a file stream to flush. Running this code causes 1001 flushes, the last one coming from QFile::close. Running this code on Qt 4.8 for Symbian is very slow. If application is running slow on Qt 4.8, the first thing to try is to remove unnecessary flushes. To fix the code above, change the line doing the actual write to the next one.

```
fileStream << "Test file content line " << i << "\n";
```

This makes the code above to flush the file stream only once, when the "close" method is called, which is an obvious improvement in this context.

NFC: Accessing target that supports multiple access methods

Background:

Due to a bug in AccessMethod enum definition it is not possible to have different access flags bitwise OR’d together.

(C) Copyright Nokia 2013. All rights reserved.

Qt Mobility 1.2.1 (and earlier) had a bug that it never set the TagTypeSpecificAccess access flag even though target supported it. Qt Mobility 1.2.2 has this fixed and TagTypeSpecificAccess flag is set if target supports it.

Impact for affected applications:

Above mentioned issues make it difficult to handle targets that support e.g. NDEF and tag type specific access in Qt Mobility 1.2.2. In this case, checking of access flags returns only TagTypeSpecificAccess due to AccessMethod [doc.qt.nokia.com] enum definition bug. This means that following application side workaround is needed to access NdefMessage:

Solution/Workaround:

Instead of having normal NDEF access check:

```
if (accessMethods.testFlag(QNearFieldTarget::NdefAccess) {
// Read or write NDEF messages
}
```

application should try for NDEF access even if TagTypeSpecificAccess is specified:

```
if (accessMethods.testFlag(QNearFieldTarget::NdefAccess) || accessMethods.testFlag(QNearFieldTarget::TagTypeSpecificAccess)) {
// Try to read or write NDEF messages
}
```

For more details, please see the QTMOBILITY-2024 bug report.

Interpretation of directory paths

Background:

In earlier versions there was some inconsistency in interpreting Symbian directory paths either as relative or as absolute paths. In Qt 4.8 this has been fixed. This means, however, that some paths may get interpreted differently in Qt 4.8 than in 4.7.4, and application behavior may change. One such case is using drive letter as a path.

Let’s have file path, say, “E:”. This is a relative path. If the Symbian application has not set the current working directory to be anything else, this path then refers to the private directory of the application – “E:\private\1234abcd”. In Qt 4.7.4 path “E:” was interpreted in some functions as if it was an absolute path, referring to the root directory “E:\”.

Impact for affected applications:

Impacted applications may malfunction due toe trying to access resources at incorrect locations in the file system.

Solution/Workaround:

Review your code and ensure that no assumptions are made about the file paths used.

QNetworkRequest – problem with Pipelined Request and “Transfer-Encoding: chunked"

Background:

This has been a known issue in Qt platforms prior to version 4.8 but is highlighted here. The problem has been fixed in some platforms (see QTBUG-19480 and QTBUG-20924) and will be also in Symbian, in a future Qt version.

The problem occurs if pipelining is enabled in the QNetworkRequest [qt-project.org] and server reply contains “Transfer-Encoding: chunked” and “Connection: close” headers, and if the request is not retry-able.

This causes the consequent request to fail as Qt tries to send the next pipelined request using the same socket.

Impact for affected applications:

Network requests will fail.

Solution/Workaround:

Workaround for the problem: if pipelining is disabled in the request, a new socket will be used. Note that sockets are only reused for requests that have pipelining enabled.

QNetwork – default value for Content-Type header field Changed

Background:

An application must always define itself the Content-Type header field in any HTTP request. Qt has a fallback for a missing Content-Type header field which in Qt 4.7 is defined as “application/x-www-form-urlencoded” while in Qt 4.8 it has been changed to “application/octet-stream”.

Impact for affected applications:

An application that has previously relied on Qt's default Content-Type header field may now have its request fail with server dependent HTTP error.

 (C) Copyright Nokia 2013. All rights reserved.

Solution/Workaround:

To avoid this problem applications must add relevant Content-Type header field themselves, see QNetworkRequest::setHeader and QNetworkRequest::setRawHeader.

Root QML object

Background:

All applications that are meant to be used both in portrait and landscape orientations, or in landscape only, and are using Qt Quick Components must use Window or one of the inherited elements as their root element. Otherwise application layout will be broken when device is rotated to landscape orientation. Size and coordinates of the Window element must not be explicitly set; Window will automatically take the size of the screen in current orientation. If the application uses different layout definitions for portrait and landscape modes, it should bind its layout state to property Window.inPortrait. Binding it to any other orientation related property may cause screen flicker when orientation is changed.

Applications locked to portrait mode can use also other root elements. In that case, the root element size must be explicitly set or the application UI may not become visible.

Applications wishing to do their own custom orientation handling must not use Window as their root element. To be compatible both with Qt 4.7 and Qt 4.8, such applications should set the attribute Qt::WA_SymbianNoSystemRotation for the QDeclarativeView from their cpp code. Then, those applications are free to set width, height, and rotation properties of their root element as they like.

Please see Window component for more info.

Impact for affected applications:

Previously validated UI design will not be rendered correctly any more.

Solution/Workaround:

See the above discussed considerations.

Applications may not obey the specified orientation lock

Background:

With earlier Qt releases if a QML-based application wanted to have a predefined screen orientation locked and thus not have its views rotate when the phone orientation changed, it would use the method setOrientation from QmlApplicationViewer and it would specify the desired orientation mode.

```
QmlApplicationViewer viewer;
viewer.setOrientation(QmlApplicationViewer::ScreenOrientationLockPortrait);
viewer.setMainQmlFile(QLatin1String("qml/TestApp/main.qml"));
```

Qt Quick Components for Symbian also define an API for setting the orientation of the view, namely the *orientationLock* property in the Page component. Until Qt 4.8 the QML setting did not override the orientation specified in the C++ code (with QmlApplicationViewer's setOrientation) but since Qt 4.8 the QML setting has precedence and its default value (PageOrientation.Automatic) will basically cause the orientation lock choice to be ignored.

Impact for affected applications:

Apps designed to work in a specific orientation mode will now ignore this design choice and their view will rotate according to phone's position. Depending on the layout parameters used at design time the application may have an undesired look and feel when displayed in the wrong orientation.

Solution/Workaround:

The application would have to be updated and the orientation lock should be specified using Qt Quick Component's Page.orientationLock property.

Qt applications must set their own user agent for HTTP requests

Background:

Whether using the HTTP stack through direct calls to QNetworkRequest or via the WebKit, 3rd party Qt applications must ensure that they set their own user agent strings which is to be used by Qt Framework for all the requests issued by the application. This will ensure that the servers receiving the requests, as well as the proxy servers filtering the traffic in the mobile networks, are able to identify and validate the particular app which is issuing the request or the platform from which this request was issued.

In Qt 4.7.4 a change was introduced so that the default minimal user agent sting used by the Qt framework includes the work "Nokia" and this was intended to fix a particular known issue affecting data traffic from Qt applications when using the SFR operator's network. Unfortunately this change was accidentally dropped from the Qt 4.8 release.

It was also signalled to us that 3rd party APIs providers may introduce API changes in which they ask not only to be able to identify the application but also the device from which the request is made, which makes it relevant for the apps using such APIs to include the device model info in the custom user agent.

Impact for affected applications:

The impact on 3rd party applications may differ a lot in the context. While some may continue to work without problems, even without setting own user agent, others will experience failure to establish a working data connection (see above SFR known issue) while others may experience functional breaks

(C) Copyright Nokia 2013. All rights reserved.

due to changes in the behaviour of 3rd party API providers.

Solution/Workaround:

See the solutions provided at Set User-Agent Header in a Qt Application (HTTP stack, WebKit) and Qt default user agent gets rejected (Known Issue) (Qt Quick) and ensure that your application's user agent string includes:

- application name and version
- keywords "Nokia" and "Qt"
- device model

QML Video and Camera elements and Qt Quick Components 1.1

Background:

The new rotation handling of Qt Quick Component 1.1 conflicts with QML Video and Camera elements. The video or the camera viewfinder can be misplaced in the screen. This happens more often in landscape orientation, also with applications that lock into landscape. Having no rotation in the application does not help.

Impact for affected applications:

The problem happens only if application uses Qt Quick Components 1.1 mixed with the above mentioned multimedia components.

Solution/Workaround:

The workaround for the problem is to disable the new rotation handling of Qt Quick Components 1.1 and to use the old rotation handling. This is done by setting QDeclarativeView property “orientationMethod” to value 1. The old rotation handling has different rotation effects. The new orientation handling must be disabled in the beginning of application before setting the main QML file. It cannot be dynamically switched on or off during later execution. Setting the property in older environments does nothing. The work-around is safe in all environments.

If you use application with helper class QmlApplicationViewer generated by Qt SDK application wizard, disabling is done like this:

```
QmlApplicationViewer viewer;
viewer.setProperty("orientationMethod", 1);
viewer.setMainQmlFile(QLatin1String("qml/MyApp/main.qml"));
```

With QDeclarativeView the same thing is done in a similar fashion:

```
QDeclarativeView view;
view.setProperty("orientationMethod", 1);
view.setSource(QUrl::fromLocalFile(QLatin1String("qml/MyApp/main.qml")));
```

Please see also the bug QTMOBILITY-2055 for more details.

Changes in tools

Changes in On Device Debugging (ODD) configuration

Background:

Starting with Nokia Belle FP1 the CODA configuration has been changed and a new macro has to be added to project's *.pro (or directly to the MMP file if you are developing a Symbian C++ project). Without this macro being present at build time CODA will fail to load and execute your executable.

Impact for affected applications:

If the macro is missing, the IDE will display the following type of output:

```
Executable file: 104061 2012-05-07T18:59:29 C:\QtSDK\Symbian\SDKs\SymbianSR1Qt474\\epoc32\release\armv5\udeb\delTest.exe
 Connecting to 'COM12'...
 Connected.
 Launching: delTest.exe
 Launch failed: Command answer [command error], 1 values(s) to request: 'C|10|Processes|start|""|"delTest.exe"|[""]|[]|true'
#0 {"Code":-46,Format="* Cannot debug 'C:\\sys\\bin\\delTest.exe': ensure the DEBUGGABLE[_UDEBONLY] flag is set in its MMP\n (permis
 Error: '* Cannot debug 'C:\sys\bin\delTest.exe': ensure the DEBUGGABLE[_UDEBONLY] flag is set in its MMP
 (permission denied)' Code: -46
 Finished.
```

Solution/Workaround:

As the error message indicates, the DEBUGGABLE macro would have to be added to your project's MMP file. In your Qt project this is done by adding the following line to your *.pro file:

```
symbian:MMP_RULES += "DEBUGGABLE"
```

Tip: The latest SDK updates releases (Qt SDK 1.2.1 and subsequent online only updates) have several fixes regarding on-device debugging on Nokia Belle FP1 devices. The above described problem should not be visible after applying these updates since the DEBUGGABLE macro will be automatically added by SDK's scripts to all the projects using the Symbian target.

(C) Copyright Nokia 2013. All rights reserved.

Note: After updating your device's firmware from Nokia Belle to Nokia Belle FP1, make sure to re-install CODA, using the package **CODA-1.0.6-for-S60v5-Anna-Belle-vFuture.sis** from c:\QtSDK\Symbian\sis\common\CODA\ . Failing to do so you will notice that any attempt to execute on-device debugging may cause a reboot of the phone.

The Remote Compiler is no longer available

Background:

The Remote Compiler tool was a set of SDK environments installed on a server and which could be accessed remotely by developers via a set a Qt Creator plug-in. This allowed for Qt applications to be compiled against SDK configurations not otherwise accessible to developers, in particular the Symbian SDK which could not be used on Linux or OS X.

Impact for affected applications:

Since the removal of the service developers will notice that while the Remote Compiler plugin still appears to be able to authenticate the user with Nokia Developer credentials, it cannot fetch configuration information from the non-existing server and thus the remote compiler service is no longer available.

Solution/Workaround:

Since the Symbian toolchain is officially supported on Windows only, the recommended solution is to install the Symbian SDK on Windows and use it locally. Alternatively you can try the experimental Symbian toolchain for Linux available on Nokia Developer Projects.

AnalyzeTool workarounds for Belle SDK

There are some bugs with AnalyzeTool on the Nokia Belle SDK. Below are some workarounds against v1.15.1:

Background

Function profiling does not work with GCCE compiler

Solution/Workaround

Do not build for armv5_urel_gcce4_4_1, but use armv5_urel_gcce instead. If you have the default sbs installation from Belle SDK (v 2.15.3), you need to do the following changes to your SDK

1. In **\epoc32\tools\sbs\lib\flm\standard.xml** of your SDK, add these lines:

   ```
   <param name='LINKEROPTION_GCCE' default=''/>
   <param name='OPTION_GCCE' default=''/>
   ```

   inside this interface:

   ```
   <interface name="Symbian.mmp" extends="base.flm" abstract="true">
   ```

2. Add these lines in **\epoc32\tools\sbs\lib\flm\e32abiv2defaults.mk** of your SDK

   ```
   LINKEROPTION:=$(addprefix -Wl$(CHAR_COMMA),$(LINKEROPTION_GCCE))
   OPTION_COMPILER:=$(OPTION_GCCE)
   ```

   inside this if clause:

   ```
   ifneq ($(TOOLCHAIN),RVCT)
   ```

For Carbide.c++ 3.2 users, there is new a version 1.15.2 available at the Carbide Portal. It fixes a problem where AnalyzeTool tried to build with RVCT despite GCCE target selected.

For more information see the discussion boards

Background:

Impact for affected applications:

Solution/Workaround:

See also

Camera key handling in Nokia Belle

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.