



ID	...	Creation date	June 22, 2009
Platform	S60 3rd Edition, FP1, FP2 S60 5th Edition	Tested on devices	Nokia 5800 XpressMusic
Category	Qt for Symbian	Subcategory	

Keywords (APIs, classes, methods, functions): XQInstaller::install(), XQInstaller::remove().

Overview

This code snippets shows how to install/uninstall application silently. The XQInstaller can be used to silently install/uninstall the SISX without user intervention.

This snippet requires TrustedUI capabilities. Self-signing is not possible because a Developer Certificate is needed.

Note: This snippet can not work on emulator, so need to test it on real device.

Preconditions

- Download and Install latest version [Qt for Symbian - Installation packages](#) which has links on how to install the latest version
- Go through this article: [How to use Mobile Extension APIs](#)

Headers file

```
#ifndef SILENTINSTALLUNINSTALL_H
#define SILENTINSTALLUNINSTALL_H

#include <QtGui/QMainWindow>
#include "ui_SilentInstallUninstall.h"
#include "xqinstaller.h"

class SilentInstallUninstall : public QMainWindow
{
    Q_OBJECT

public:
    SilentInstallUninstall(QWidget *parent = 0);
    ~SilentInstallUninstall();

public Q_SLOTS:
    //slots to receive action of menu trigger.
    void installHelloWorldAction();
};
```

Silent_installation/uninstallation_of_application_in_Qt

```
void uninstallHelloWorldAction();
//slots to receive signal from XQInstaller.
void installationComplete();
void installationError();
void uninstallationComplete();
void uninstallationError();

public:
    //new functios
    void SilentInstall();
    void SilentUninstall();
    void createMyMenu();

private:
    Ui::SilentInstallUninstallClass ui;

    QAction* menu_installHelloWorldAction;
    QAction* menu_uninstallHelloWorldAction;
    QAction* menu_exitAction;

    XQInstaller installer;
bool result;
};

#endif // SILENTINSTALLUNINSTALL_H
```

.pro file

```
symbian:LIBS += -lswinstcli \
    -lcommdb \
    -lapparc \
    -lefsrv \
    -lapgrfx
symbian:TARGET.CAPABILITY += TrustedUI
```

Source

```
#include "SilentInstallUninstall.h"
#include <QMessageBox>
#include <QDir>

SilentInstallUninstall::~SilentInstallUninstall()
{
}

SilentInstallUninstall::SilentInstallUninstall(QWidget *parent)
    : QMainWindow(parent)
{
    setupUi(this);
    setWindowTitle("Silent InstallUninstall");
    createMyMenu

// Create the installer class
    installer = new XQInstaller(this);
}
```

Headers file

Silent_installation/uninstallation_of_application_in_Qt

```
void SilentInstallUninstall::SilentUninstall()
{
    (connect(this, SIGNAL(applicationRemoved()), this, SLOT(uninstallComplete())));
    (connect(this, SIGNAL(error()), this, SLOT(uninstallError())));

    /* ED8788DC is UID of helloworld
       * convert UID from hex to decimal
       */
    QString uid{"0xED8788DC"};
    bool ok;
    uint appIdToUInt(&ok, 16);

    /* Uninstall the application with the given UID
       * Or result = installer->remove(3985082588);
       */
    result = installer->remove(appId);

    if (!result)
    {
        emit installationError
    }
}

void SilentInstallUninstall::SilentInstall()
{
    // Connect to the applicationInstalled() signal
    (connect(this, SIGNAL(applicationInstalled()), this, SLOT(installationComplete())));
    (connect(this, SIGNAL(error()), this, SLOT(installationError())));

    // Install HelloWorld.SISX from private path
    QString fileName{cleanPath(QDir::currentPath() + QLatin1Char('/') + "HelloWorld.SISX")};
    result = installer->install(fileName.replace("/", "\\"));
    if (!result)
    {
        emit installationError
    }
}

void SilentInstallUninstall::createMyMenu()
{
    menu_installHelloWorld = new QAction(tr("Install HelloWorld"), this);
    (connect(menu_installHelloWorldAction, SIGNAL(triggered()), this, SLOT(installHelloWorldAction())));

    menu_uninstallHelloWorld = new QAction(tr("Uninstall HelloWorld"), this);
    (connect(menu_uninstallHelloWorldAction, SIGNAL(triggered()), this, SLOT(uninstallHelloWorldAction())));

    menu_exit = new QAction(tr("Exit"), this);
    (connect(menu_exitAction, SIGNAL(triggered()), this, SLOT(close())));
}

void SilentInstallUninstall::installHelloWorldAction()
{
    SilentInstall
}
}
```

Silent_installation/uninstallation_of_application_in_Qt

```
void SilentInstallUninstall::uninstallHelloWorldAction()
{
    SilentUninstall
}

void SilentInstallUninstall::installationComplete()
{
    QMessageBox::information(0, "Congrats!!!", "Application is installed sucessfully");
}

void SilentInstallUninstall::installationError()
{
    QMessageBox msgBox
    msgBox("Error in installation.");
    msgBox(QMessageBox::Critical);
    msgBox.standardButtons(QMessageBox::Ok);

    switch (installer->error())
    {
    case XQInstaller::OutOfMemoryError:
    {
        setInformativeText(msgBox.enough memory");
    }
    break;
    case XQInstaller::AlreadyInUseError:
    {
        setInformativeText(msgBox.installer is already in use");
    }
    break;
    case XQInstaller::UserCancelError:
    {
        setInformativeText(msgBox.installer cancelled by the user.");
    }
    break;
    case XQInstaller::PackageNotSupportedError:
    {
        setInformativeText(msgBox.package not supported");
    }
    break;
    case XQInstaller::SecurityFailureError:
    {
        setInformativeText(msgBox.security failure");
    }
    break;
    case XQInstaller::MissingDependencyError:
    {
        setInformativeText(msgBox.missing dependency");
    }
    break;
    case XQInstaller::NoRightsError:
    {
        setInformativeText(msgBox.rights ");
    }
    break;
    case XQInstaller::BusyError:
    {
        setInformativeText(msgBox.installer is busy");
    }
    break;
    case XQInstaller::AccessDeniedError:
    {
        setInformativeText(msgBox. Access denied"
```

Silent_installation/uninstallation_of_application_in_Qt

```
}
break;
case XQInstaller::UpgradeError:
{
    setInformativeText(mMessageBox. Error while upgrading"
}
break;
case XQInstaller::UnknownError:
{
    setInformativeText(mMessageBox. "Unknown error.");
}
break;

default:
break;
}

    QMessageBox.
}
```

Postconditions

The code snippet is expected to install/uninstall helloworld application silently.

Code Example

- The [Code Example](#) will show how to install/uninstall application silently. The application is tested on Nokia 5800 XpressMusic.

Related Links

- [Get list of installed applications and its UID in Qt](#)