Qt & Application Signing/zh-hans

This article needs to be updated: If you found this article useful, please fix the problems below then delete the {{ArticleNeedsUpdate}} template from the article to remove this warning.

Reasons: hamishwillee
(02 Sep 2011)
Please (partially) re-translate Qt & Application Signing to include changes since last revision before 20110902. At high level the changes are:

- Developer Certificates are now granted to both individuals and companies through the site - as a result Open Signed Online is no longer need and has been discontinued.
- Open Signed Offline and Open Signed Online are no longer used to describe process for getting DevCerts. Instead instructions on how to get a devcert are given in the user guide: User guide: Symbian Signed.
- User guide: Symbian Signed is now a high level important document
- Symbian Signed is no longer run through Symbian Foundation

Note, I have made some of the needed changes, simply deleting links to Open Signed Offline/Online references. I am not certain that the text still makes sense in Chinese.

Symbian平台上的Qt应用配置在数字签名的安装包中（被称为"SIS"文件）。本文为你的QT应用开发和发布选择合适的数字签名。接着，文章讲解如何用Qt Creator 或qmake来对你的应用进行签名。

注意：配置Qt应用一文讲解了签名前如何配置安装包。

何谓签名选项？

签名选项中包含Symbian Signed(Express Signed和Certified Signed)（开发伙伴认证签名，有 以及Self-Signing（自签名）。每一种选项都于应用安装时向用户给出不同的警示，同时提供不同的访问能力- 有关签名选项所赋予的能力(capability) 方面的更多信息请参阅：Capabilities (Symbian Signed)。

以下章节将对这些选项展开讨论。

Symbian Signed

Symbian Signed是一个测试程序及为软件签名的项目。经过Symbian Signed签名过的应用程序内置营销商防篡改数字证书，其中包含的二进制文件授权添加很多功能项，使应用程序可以在无报警的情况下安装到Symbian终端上。像Handango和诺基亚Ovi商店等许多发行渠道要求应用程序必须经过Symbian签名。

很多应用程序既可以通过Express Signed或者Certified Signed路径实现Symbian签名。大部分Qt应用都可以通过Express Signed做签名，很少应用程序需要得到Certified Signed所授予的受限功能或终端制造商功能。

不管通过哪一种路径，应用程序都要符合Symbian Signed测试标准。区别在于：每一个Certified Signed应用都由某家测试机构测试提供，而Express Signed应用只是偶尔得到审核（如果不能通过审核，该开发者将需在以后的测试中使用Certified Signed）。因而，Express Signing的价格要便宜许多，但由于其信任度较低，只能给用户及系统添加功能。如果需要的话，Certified Signed应用则还能获得一些受限的功能项。

开发者证书

仅使用用户功能项的那些Qt应用程序可以在开发期间使用自签名，并不需要开发者证书。

Developer certificates(开发者证书，DevCerts)是由Symbian Signed颁发的有限证书，目的是让大家能在开发期间使用一些更为高级的功能项。用某张DevCert签名的应用被绑定于某台特定移动终端（或一批终端），并会在安装时显示一条警告信息，提示该应用并不获信任。

自签名

自签名是一种以开发者自己创建的证书对应用进行签名的方案。由于平台对这类应用并无任何信任基础，因而：

- 需使用"user capabilities" （用户能对该应用的安装风险进行决策的那些"capabilities"，即`NetworkServices`, `LocalServices`, `Location`, `ReadUserData`, `WriteUserData`, 及`UserEnvironment`）之外的"capabilities"的自签名应用无法被安装。
  - 注：一些早期S60 3rd edition手机(pre FP2)上使用`Location`的应用不能被自签名。
- 终端用户在安装应用时被告知该应用为"不获信任的"，然后又被告知需要哪些用户capabilities，询问其是否希望继续安装过程。
- 许多发行渠道不接受自签名应用。例如，Ovi商店就要求你的应用都经过Symbian Signed签名。

默认地，*Qt Creator IDE* 和*qmake*对应用进行自签名。假设你并不希望通过某些应用商店发行应用，也不在意出现警告信息，自签名可以成为某种可接受的签名方案。'

选对签名选项

针对测试/开发我应选用何种选项？

大部分Qt应用仅要求"user capabilities"，因而能在开发过程中使用自签名（*Qt Creator*默认使用一个自签名证书，因此你无需任何作为！）

针对发行我应选用何种选项？

(C) Copyright Nokia 2013. All rights reserved.

需要通过各种应用商店发行的Qt应用或那些不能出现"信任警示"的Qt应用都必须经过Symbian Signed签名。大部分的Qt应用都能使用*Express Signed'路径（*Certified Signed *Signed用于需要用到限制capabilities的场合，或者，如果该应用无法满足测试标准并需要申请免测的话）*。

仅需要*user* capabilities的应用，及安装警告无伤大雅的那些应用，可以用自签名。

针对免费软件签名我应选用何种选项？

仅需*user capabilities*的应用可以用自签名。

DevCerts 可用于需要*system capabilities*的免费软件的签名，假定这些应用并不使用智能安装程序（Smart Installer）。因而，这些应用必须内嵌Qt二进制文件，或者是单独发行的。 '

到本文撰写时为止，并无针对免费软件的Symbian Signed签名机制；但是请注意，各种应用商店有时会对应用签名进行赞助。

如通过应用商店发行应用，我应选用何种选项？

大多数应用商店要求这些应用通过Symbian Signed签名。需要注意的是：诺基亚Ovi商店也要求你内置Smart Installer。

如何对应用签名？

本节讲解如何用*Qt Creator*或以命令行命令*qmake*对你的应用进行签名。

需要注意的是，Qt应用通常是用诺基亚Smart Installer for Symbian来部署的，这意味着应用必须被两次签名，而且只能用*qmake*来创建用于发行的安装包。

开发阶段对应用签名

使用Qt Creator

如果你的应用仅需要一些基本的*user capabilities*，那么你无需任何作为！Qt Creator会自动地对应用的SIS文件进行适当的测试并以默认的自签名证书签名。该SIS文件中并不包括Qt二进制文件，也不包括smart installer – 假定你在安装自己的开发环境时已经安装了Qt。

如果你的应用需要*system* capabilities（或更高级别的），那么你就需要申请开发者证书。DevCerts 提供渐进式指南，为你讲解如何获得开发者证书，然后就可以用其来替换掉Qt Creator签名时所使用的证书，具体做法请见这里。

请注意，到撰写本文时为止，如果你的私钥具有密码，那么你就不能使用Qt Creator，具体请见QTCREATORBUG-950。

使用*qmake*

在命令行中，进入到应用项目目录（即含有.pro文件的目录）然后：

```
qmake                    //create make files
make release-gcce      //build application release version with the GCCE compiler
make sis                 //create the application SIS file (self signed with default certificate)
```

完成后，该应用目录中将含有以默认证书自签名后的*应用名字.sis*。如果你希望创建一个smart installer版本，那么你可以这么创建：

```
make installer-sis    //create the wrapper SIS file containing application SIS and smart installer (self signed)
```

如果你的应用需要一些*system*capabilities（或更高级别的）那么你就需要申请开发者证书。DevCerts 提供渐进式指南，为你讲解如何获得开发者证书，然后就可以用`QT_SIS_CERTIFICATE`, `QT_SIS_KEY`及`QT_SIS_PASSPHRASE`选项替换掉Qt Creator所使用的证书，如下所示（注意在这种情况下私钥中不能有密码）：

```
qmake                    //create make files
make release-gcce      //build application release version with the GCCE compiler
make sis QT_SIS_CERTIFICATE=yourcertificate.cer QT_SIS_KEY=yourkey.key
make installer-sis QT_SIS_CERTIFICATE=yourcertificate.cer QT_SIS_KEY=yourkey.key
```

对发行应用的自签名

将发行的那些应用通常以一个证书签名，该证书中包含作者的详细信息，也经常会包含Smart Installer（能保证用户安装应用后无需再作其它事情就能运行该应用）。

MakeKeys工具可用来创建证书 –MakeKeys Tutorial 向大家提供简洁的指导。请注意，MakeKeys在Symbian独立SDKs而并非Nokia Qt SDK (QTSDK-127)中提供。

创建证书后，如上文所述去创建smart installer *application_installer.sis*：

```
qmake
make release-gcce
make sis QT_SIS_CERTIFICATE=yourcertificate.cer QT_SIS_KEY=yourkey.key
make installer-sis sis QT_SIS_CERTIFICATE=publisherid.cer QT_SIS_KEY=publisherid.key
```

注意，当发行一个应用时，你或许还要对其SIS包定义中所用到的某些应用信息进行更新。文章Deploying a Qt Application对此有所讨论。

对应用作Symbian Signed签名

提交Symbian Signed的应用必须先用一个publisher ID签名（该证书由某家认证中心审核并保证你是该应用的供应者）并以某种特定方法打包，然后才能提交。文章Publisher ID (Symbian Signed)讲解了如何获取publisher ID。大多数情况下你需要，或希望，将Smart Installer包含其中。这意味着你需要对文件作两次签名。

(C) Copyright Nokia 2013. All rights reserved.

因而，对SIS文件进行Symbian Signed签名的过程是：

```
qmake                    //create make files
make release-gcce      //build application release version with the GCCE compiler
make sis QT_SIS_CERTIFICATE=yourPublisherID.cer QT_SIS_KEY=yourPrivateKey.key
```

然后如Express Signed (Symbian Signed)或Certified Signed (Symbian Signed)所述对*applicationname.sis* SIS文件进行打包并提交签名。获得已签名SIS文件后，请将其复制以覆盖未经签名的原SIS文件然后以下列操作创建*wrapper* SIS文件，并用你自己的publisher ID对其进行签名：

```
make installer-sis sis QT_SIS_CERTIFICATE=yourPublisherID.cer QT_SIS_KEY=yourPrivateKey.key
```

可以将该文件提交Express Signed (Symbian Signed)（wrapper无需再被certified signed），返回的文件就可以发行了。

请注意，当通过Symbian Signed发行某个应用时你还需要对其SIS包定义中所用到的一些应用信息进行更新。文章应用：部署一个Qt应用对此有讨论。

免费软件的签名

仅需*user capabilities*的免费应用可以如上文所述对其进行自签名。

常见问题解答

"smart installer" 对签名操作有何影响？

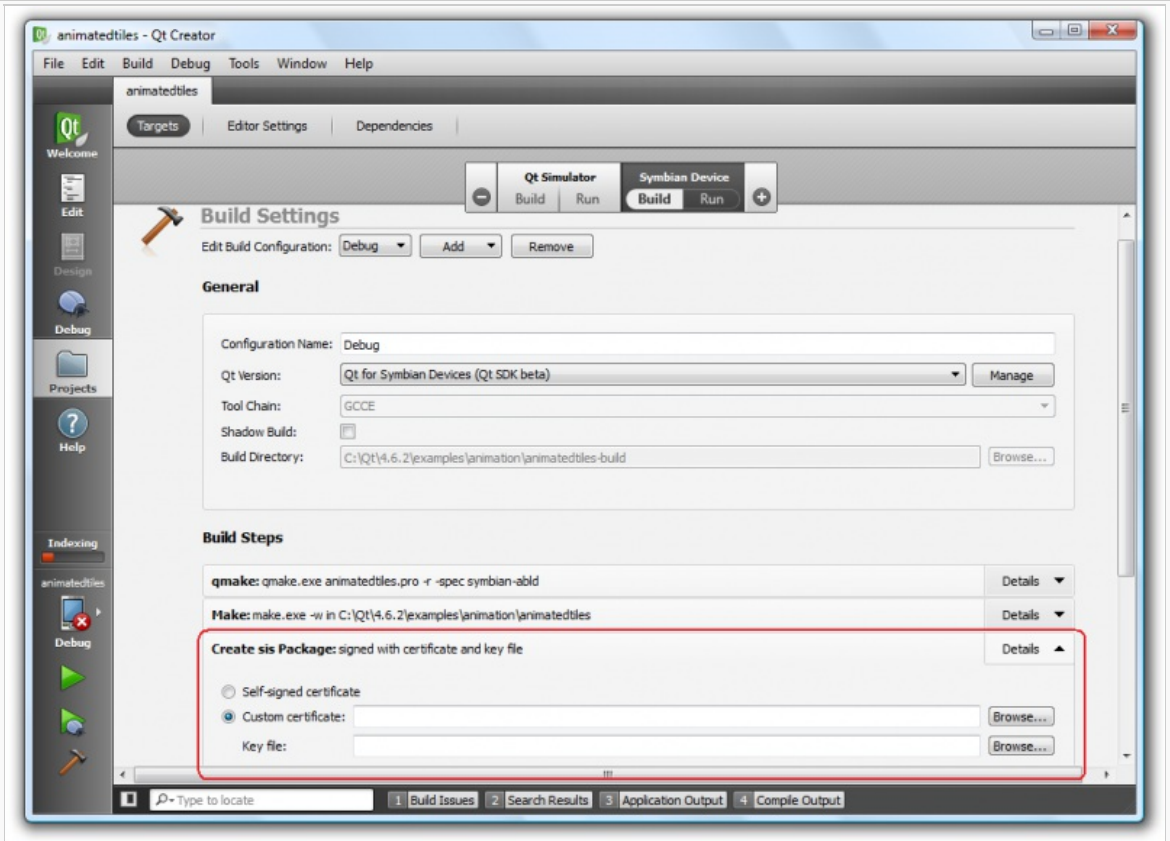
诺基亚Smart Installer for Symbian是一个工具，它可以与你的应用SIS 文件一起被包括到一个封装SIS 文件中，当那个封装包被安装到终端上时，应用的Qt dependencies也被先行下载并安装（如果终端中并不存在的话），然后才会安装应用。这是确保你的应用能在安装后立即运行的最有效方法，因为智能安装程序比起让应用自己绑定Qt二进制程序的做法来要智能得多。一些应用商店（如Ovi）要求Qt应用必须包括Smart Installer。'

使用智能安装程序的应用必须经过两次签名！首先要将应用绑定到一个SIS文件并加以签名，然后被签名应用和智能安装程序一起被嵌入到另一个SIS文件，后者也必须被签名。使用本地证书（自签名或开发者签名）时，Qmake命令行工具链为你完成这些工作。如果你准备对自己的应用作Symbian Signed签名，你就还需要多做几步，因为应用SIS文件需要被提交到 https://www.symbiansigned.com/signedui/welcome 网站进行两次签名。

而且还会产生附加成本，因为每次签名都需要金钱。已经在研究一些方案以减轻这项负担，例如，本文撰写时，Ovi正在试行一项beta版服务，以一次性的小额会员费吸收签名成本。

如何改变Qt Creator为签名所使用的证书？

在"Project"模式（见左功能条中的按钮）中针对"device"构建目标指定证书。请注意**Build Steps | Create sis Package**。就选择"Custom certificate"单选按钮，指定替换证书、私钥和密码。



Qt creator – 改变证书

Tip: 撰写本文时你还不能改变Qt Creator要构建哪个包文件。这也是其不能用于创建Smart Installer SIS文件的原因。

(C) Copyright Nokia 2013. All rights reserved.

如何起步？

读完本文，你已经知道你要通过申请，以Capabilities (Symbian Signed)签名功能为基准应该选用那一种签名。。Qt开发伙伴通常适用自签名，或Express Signed签名。

接下来可以阅读应用：配置Qt应用，它讲解了如何在对你的项目签名之前你要做的其它方面的改变。

在哪里可以咨询到更多信息？

应用：Symbian Signed 全解、分录：Symbian Signed及下面有关选项的一些文章提供了更多的信息：

- User guide: Symbian Signed
- Express Signed
- Certified Signed
- Developer Certificates
- 自签名

可以在下面这些论坛就Symbian Signed提问：

- Symbian Signed Support, Application Packaging and Distribution and Security

如果你对本文有一些见解，请发布一条维基评论（登录的用户可在该页页脚看到"Leave a comment"链接）。

综述

本文向Qt应用开发伙伴简要介绍应用程序的签名。文章讲解了各种签名选项，如何改变Qt Creator所使用的证书以实现应用程序的签名，及如何通过Qt工具链对应用程序进行签名。

© 2010 Symbian Foundation Limited. This document is licensed under the Creative Commons Attribution-Share Alike 2.0 license. See http://creativecommons.org/licenses/by-sa/2.0/legalcode for the full terms of the license.

Note that this content was originally hosted on the Symbian Foundation developer wiki.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.