<http://delta.affinix.com/qca/>

**Qt Cryptographic Architecture (QCA)**

Language: English | [Serbo-Croatian](http://science.webhostinggeeks.com/qca) | [Bulgarian](http://weknownyourdreamz.com/qca.html)



**What is it?**

Taking a hint from the similarly-named [Java Cryptography Architecture](http://java.sun.com/j2se/1.3/docs/guide/security/CryptoSpec.html), QCA aims to provide a straightforward and cross-platform crypto API, using Qt datatypes and conventions. QCA separates the API from the implementation, using plugins known as Providers. The advantage of this model is to allow applications to avoid linking to or explicitly depending on any particular cryptographic library. This allows one to easily change or upgrade crypto implementations without even needing to recompile the application! QCA should work everywhere Qt does, including Windows/Unix/MacOSX.

**What do I need to be able to use it?**

QCA depends on [Qt 4.2](http://www.trolltech.com/).

**What features are supported?**

* SSL/TLS
* X.509
* OpenPGP
* Cryptographic Message Syntax (for S/MIME)
* SASL
* PKCS#12
* Smart cards and other key storage
* Secure memory
* Secure random
* RSA/DSA/Diffie-Hellman
* Hash/Cipher/MAC algorithms
* Threading considerations
* Access operating system root certificates

**How does it work?**

The application includes and links to libqca, which provides the ‘wrapper API’ and plugin loader. Crypto functionality is determined during runtime, and plugins are loaded from the ‘crypto’ subfolder of the Qt Library Paths.

**What is the development plan?**

The base QCA library is considered mostly complete. Future development may include OCSP, shared certificate management, and improving the plugin API. However, there is no development plan for the base library at this time. Work will continue on the various plugins though, in order to get them into a release-quality state.

**Compatibility**

QCA follows a typical API/ABI compatibility scheme. Versions use the “major.minor.patch” format. Major versions of QCA may be incompatible with each other, minor versions are backwards compatible within the same major version, and patch versions are forwards and backwards compatible within the same minor version. This is the same scheme that Qt itself uses.

**Documentation**

API documentation in HTML format is included in the source distribution, and is also available [on the web](http://delta.affinix.com/docs/qca/), enjoy!

**Where can I get it?**

**Source:** [**qca-2.1.0.tar.gz**](http://delta.affinix.com/download/qca/2.0/qca-2.1.0.tar.gz)

QCA has built-in support for the SHA1 and MD5 hash algorithms, and a weak random number source. To perform any other crypto operations (or to utilize improved versions of the built-in ones), an appropriate provider plugin is needed. There are several available so far, and you could always write your own, too.

**NOTE: As of QCA 2.1, the latest plugins are included in the main tarball and don’t need to be downloaded separately.**

|  |  |  |
| --- | --- | --- |
| **Provider** | **Capabilities** | **Depends on** |
| [qca-ossl-2.0.0-beta3.tar.bz2](http://delta.affinix.com/download/qca/2.0/plugins/qca-ossl-2.0.0-beta3.tar.bz2) | TLS, CMS, X.509, RSA, DSA, Diffie-Hellman, PKCS#7, PKCS#12, SHA0, SHA1, SHA224, SHA256, SHA384, SHA512, MD2, MD4, MD5, RIPEMD160, Blowfish, DES, 3DES, AES128, AES192, AES256, CAST5, HMAC(SHA1, MD5, RIPEMD160), PBKDF1(MD2, SHA1), PBKDF2(SHA1) | OpenSSL 0.9.6+ |
| [qca-cyrus-sasl-2.0.0-beta3.tar.bz2](http://delta.affinix.com/download/qca/2.0/plugins/qca-cyrus-sasl-2.0.0-beta3.tar.bz2) | SASL | Cyrus SASL2 |
| [qca-gnupg-2.0.0-beta3.tar.bz2](http://delta.affinix.com/download/qca/2.0/plugins/qca-gnupg-2.0.0-beta3.tar.bz2) | OpenPGP | GnuPG 1.x or 2.x (runtime dependency only) |
| [qca-pkcs11-2.0.0-beta2.tar.bz2](http://delta.affinix.com/download/qca/2.0/plugins/qca-pkcs11-2.0.0-beta2.tar.bz2) | PKCS#11 (for smart cards) | [pkcs11-helper](http://www.opensc-project.org/) 1.02+ |
| [qca-wingss-2.0.0-beta1.tar.bz2](http://delta.affinix.com/download/qca/2.0/plugins/qca-wingss-2.0.0-beta1.tar.bz2) | SASL GSSAPI (for Kerberos) | Windows only (uses SSPI) |
| [qca-logger-2.0.0-beta2.tar.bz2](http://delta.affinix.com/download/qca/2.0/plugins/qca-logger-2.0.0-beta2.tar.bz2) | Internal module logging (for debug purposes) | None |

The [latest source](http://delta.affinix.com/source/) can be found in the ‘qca’ module of the KDE Git. The code can also be browsed [on the web](http://quickgit.kde.org/?p=qca.git).

Older versions may be downloaded [here](http://delta.affinix.com/download/qca/).  There are also old binaries: [MinGW](http://delta.affinix.com/download/qca/2.0/qca-2.0.1-mingw.zip), [Visual Studio 2003](http://delta.affinix.com/download/qca/2.0/qca-2.0.1-vs2003.zip), [Visual Studio 2005](http://delta.affinix.com/download/qca/2.0/qca-2.0.1-vs2005.zip), [Mac OS X](http://delta.affinix.com/download/qca/2.0/qca-2.0.1.dmg).

Last updated November 7th, 2014.  
*‘keys’ icon from KDE*