Ionic SDK Release 1.8.0

Introduction

Welcome to the 1.8 release of the Ionic SDK. This is mainly a maintenance release with many small improvements.

Details are summarized below.

Getting Started

You can find an introduction to the Ionic platform including a variety of "getting started" tutorials at Ionic's developer portal: dev.ionic.com

SDK Reference

Ionic offers our SDK for a wide range of languages and platforms. In this package, we have included the SDKs for the following (click to view the documentation): Ionic 1.8 SDK Reference

If you require the Ionic SDK for another platform or language, or have other questions, please contact Ionic.

New Features/Improvements

Default Linux Profile Persistor implementation

The new default Linux profile persistor relies on the operating system file security and not Ionic encryption. The persistor writes to the user directory with only user access.

Default Linux Key Vault implementation

Similar to the profile persistor, the new Linux Key Vault relies solely on the operating system file security instead of Ionic encryption. The vault writes to the user directory with only user access.

Agent clone constructor

Previously, we recommended creating a pool of agents for multi-thread environments. Since the new clone constructor is significantly faster than constructing an agent from scratch, you can now just clone an agent as needed by new threads.

Profile Manager

The device profile management part of the ISAgent class has moved to the ISAgentProfileManager class. As part of this change you can now also create a profile manager from a raw JSON string containing a plain text profile definition, and you can initialize the agent with a profile manager.

KNS additions

KNS support has increased to allow getKeyspace calls from an agent with no profiles. We have also added the method, updateProfileFromKNS() which will update a profile with the latest URL available from the KNS provider. You may also pass in a different KNS provider URL instead of using the Ionic server.

Documentation Improvements

Improvements include new code samples, corrections, some new detail sections, removed different language samples, addressing various minor issues and improved parity between the Core C++ SDK and its wrappers.

Keep alive now used with HTTP connections

Keeping connections alive between calls should improve performance. (https://en.wikipedia.org/wiki/HTTP_persistent_connection)

New support for multiple key spaces

You may now query a list of profiles to determine which profile applies to a key. Decrypt operations will now automatically use the correct profile instead of only using the active profile.

Improved handling of entropy exhaustion on Linux

Previously the SDK could hang for long periods of time waiting on the operating system to generate entropy for the random number generator. (This was especially the case on VM's where there may be no external entropy source.) Now, the SDK should error out with a suggestion to install an entropy utility like 'haveged'.

New multi-process / multi-thread safe serialization for profile persistors.

Save profile is now safe for add / remove operations. Profile update operations are atypical, but simultaneous update operations on the same profile will still result in the change set saved last.

Generic v1.3 truncate

The generic v1.3 cipher now includes a file truncation method for encrypted files.

ISCryptoString improvements

As with ISCryptoBytes in the last release, ISCryptoString now receives the secure allocator which clears dead memory on every relevant size change instead of just on destruction of the object. This means you cannot pass ISCryptoString as a std::string anymore, so the new ISCryptoStringNoResize can be used in those cases. If you need to resize, we recommend you call outside functions with ISCryptoStringNoResize and then copy that value to ISCryptoString before any resizing happens. (In many cases, ISCryptoStringNoResize is all you need, since the common usage would be to call a function to collect a string and then pass it somewhere else.) This change removed the internal ISCryptoSecureContainer and ISCryptoScopedSecureCleaner classes.

Issues Addressed

Bug Fixes and optimizations for the Generic v1.3 cipher

Some read/write mode issues have been addressed along with optimizations for that mode.

Improvements in HTTP transactions

Improved handling of non-UTF8 characters.

Prevents double encryption

All file encrypt ciphers will now return ISFILECRYPTO_ALREADY_ENCRYPTED if the source file has already been encrypted by the same Ionic cipher. Previously, only the "Auto" cipher had this feature.

KeyVault function return codes fixed

Some key vault functions failed to correctly return the key vault count.

Manually edited plaintext profile fixes

Fixed an issue where if a user manually edited a plain text JSON device profile record and left a trailing slash on the URL.

More bug fixes

Including a fix for bad handling of a server 502 error.

ABI issues

SDK users may find ABI compiler errors on older Linux distributions. We found the following compiler switch resolves this issue: "-D_GLIBCXX_USE_CXX11_ABI=0".

Discontinued Support

SDK has discontinued the Java JNI language wrapper build. The native Java library replaces it.

Additional Notes

Boost library updated

Improved exception reporting in the Python and .Net language wrappers

Exceptions now include the SDK version and Git commit.

Improved testing coverage

More thorough unit testing, which in turn means a few minor bug fixes around error handling.

Supported Platforms

The Ionic SDK is tested against the following platform configurations:

PLATFORM	VERSION
LINUX	CentOS 7.5-1804
LINUX	Ubuntu 18.04
LINUX	Mint 18.2
WINDOWS	Windows 7 (32 and 64 bit)
WINDOWS	Windows 8.1 (32 and 64 bit)
WINDOWS	Windows 10 (32 and 64 bit)
MACOS	macOS 11 (El Capitan)
MACOS	macOS 12 (Sierra)
MACOS	macOS 13 (High Sierra)
PYTHON	2.7.x
PYTHON	3.5.x