|  |
| --- |
|  |
| CrypgetSQL試験手順 |
|  |

|  |
| --- |
| 2016年4月 |

目次

[1. はじめに 3](#_Toc417393588)

[1.1. 試験概要 3](#_Toc417393589)

[1.1.1. 機能試験 3](#_Toc417393590)

[1.1.2. メモリリーク試験 3](#_Toc417393591)

[1.1.3. 性能試験 3](#_Toc417393592)

[1.1.4. マルチスレッド試験 3](#_Toc417393593)

[1.1.5. 結合試験 3](#_Toc417393594)

[1.1.6. 移行試験 4](#_Toc417393595)

[1.2. 試験アプリ概要 4](#_Toc417393596)

[1.2.1. 全体像 4](#_Toc417393597)

[1.2.2. 単体試験アプリ群 4](#_Toc417393598)

[1.2.3. 結合試験アプリ群 4](#_Toc417393599)

[1.3. 本書の構成 5](#_Toc417393600)

[1.4. 用語 5](#_Toc417393601)

[2. 試験環境の準備 6](#_Toc417393602)

[2.1. SDKとADTの準備 6](#_Toc417393603)

[2.2. SDK Platformの準備 8](#_Toc417393604)

[2.3. エミュレータの準備 9](#_Toc417393605)

[2.4. 実機の準備 13](#_Toc417393606)

[2.4.1. USBドライバのインストール 13](#_Toc417393607)

[2.4.2. 実機での設定 13](#_Toc417393608)

[2.4.3. 接続の確認 14](#_Toc417393609)

[2.5. 試験アプリの準備 15](#_Toc417393610)

[2.5.1. 試験アプリのインポート 15](#_Toc417393611)

[2.5.2. 試験アプリの設定 15](#_Toc417393612)

[2.5.3. モジュールの配置 16](#_Toc417393613)

[3. 試験手順 17](#_Toc417393614)

[3.1. 共通の手順 17](#_Toc417393615)

[3.1.1. Android JUnitの実行方法 17](#_Toc417393616)

[3.1.2. Android Applicationの実行方法 18](#_Toc417393617)

[3.2. 機能試験の手順 19](#_Toc417393618)

[3.2.1. SQL試験の手順 19](#_Toc417393619)

[3.2.2. Java API試験の手順 22](#_Toc417393620)

[3.3. メモリリーク試験 22](#_Toc417393621)

[3.3.1. メモリリーク試験 23](#_Toc417393622)

[3.3.2. Corrupt試験 25](#_Toc417393623)

[3.4. 性能試験の手順 25](#_Toc417393624)

[3.5. マルチスレッド試験の手順 25](#_Toc417393625)

[3.6. 結合試験の手順 26](#_Toc417393626)

[3.6.1. 試験番号1.1.X 26](#_Toc417393627)

[3.6.2. 試験番号1.2.X 28](#_Toc417393628)

[3.6.3. 試験番号1.3.X 29](#_Toc417393629)

[3.6.4. 試験番号1.4.X 30](#_Toc417393630)

[3.6.5. 試験番号1.5.X 31](#_Toc417393631)

[3.6.6. 試験番号1.6.X 31](#_Toc417393632)

[3.6.7. 試験番号1.7.X 31](#_Toc417393633)

[3.6.8. 試験番号2.X 33](#_Toc417393634)

[3.6.9. 試験番号3.X 33](#_Toc417393637)

[3.6.10. 試験番号4.X 35](#_Toc417393638)

[3.6.11. 試験番号5.X 36](#_Toc417393639)

[3.6.12. 試験番号6.X 37](#_Toc417393640)

[3.6.13. 試験番号7 .X 39](#_Toc417393641)

[3.6.14. 試験番号8.X 40](#_Toc417393642)

[3.6.15. 試験番号9.X 40](#_Toc417393643)

[3.7. 移行試験の手順 41](#_Toc417393644)

[4. 付録 42](#_Toc417393645)

[4.1. ADBについて 42](#_Toc417393646)

[4.1.1. ADBの環境変数設定 42](#_Toc417393647)

[4.1.2. ADBが認識している端末の確認 42](#_Toc417393648)

[4.1.3. ADBの再起動 42](#_Toc417393649)

[4.1.4. Android端末のログ取得 42](#_Toc417393650)

[4.1.5. APKのインストール、再インストール 43](#_Toc417393651)

[4.1.6. アプリケーションのアンインストール 43](#_Toc417393652)

[4.2. APKの作成 43](#_Toc417393653)

[4.3. 文字コードの設定 45](#_Toc417393654)

[参考文献 46](#_Toc417393655)

# Introduction

This document describes the procedure of the test CrypgetSQL.

CrypgetSQL is a library that runs on Android, provides encryption DB function.

As a feature of the CrypgetSQL, it has become operational with the Java API to work with SQLite, which is standard on Android roughly in the same Java API (depends in part). The details of the function of CrypgetSQL You are beyond the scope of this book. Please refer to the Programmer's Guide.

## Test Description

In this section, we describe the overview of the entire test. The specific test procedures are described later in Chapter 3.

Test, functional test, a memory leak test, performance test, multi-threaded test, binding studies, will be constructed from the migration test.

### Functional test

Functional test is divided into the Java API test to confirm and SQL test to make sure that there is no problem as a function of the database, that there is no as the API of the Android SQLite problem.

SQL test will be to a port of the SQLite test app to Android Java.

Java API test, CTS (Compatibility Test Suite) [footnoteRef: 1] of the, a modification of the application to test the android.database package, and an application to test the android.database.sqlite package to test the CrypgetSQL is. [1: https://source.android.com/compatibility/cts-intro.html][[1]](#footnote-1)

### Memory leak test

Memory usage be used for a long time CrypgetSQL Make sure that there is no increasing trend. It also serves as a heat run test.

### performance test

And measure the performance of CrypgetSQL.

### Multi-threaded test

Make sure that you can use the CrypgetSQL from multiple threads.

### Binding study

Test and to verify that available from the real application, the data stored in one application, the test to confirm that it can not read from another application, to ensure that no read data stored in the certain terminal on another device test, and so on. Test application are multiple, also procedure has become more complex.

### The migration test

Be changed CrypgetSQL from the old version to the new version, make sure that you can continue to use the database file that you created.

## Test app Overview

In this section, we describe a description of the archive that summarizes the test app.

### Overall picture

Overall picture of the archive is shown in Table 11.‑

Table 11 overall picture of the test app‑

|  |
| --- |
| CrypgetSQLTest  | CrypgetSQL試験手順.docx：本書  + V1.1  | + app  | | unit\_test ：単体試験アプリ群です。  | | integration\_test ：結合試験アプリ群を格納しています。  | + doc ：試験仕様書を格納しています。  | + modules : 試験対象のモジュールを格納しています。  + V2V3  | + app  | | unit\_test ：単体試験アプリ群です。  | | integration\_test ：結合試験アプリ群を格納しています。  | + doc ：試験仕様書を格納しています。  | + modules : 試験対象のモジュールを格納しています。 |

窶ｻ V1.1 might not bundled.

窶ｻ There is a case where modules is not also stored. In that case, you should get from a common foundation.

### Unit testing application group

It is composed of some of the Eclipse project. In these projects, the functional test, a memory leak test, performance test, and implement a multi-thread test.

### Binding test application group

It consists of ten Eclipse projects. In these projects, it will perform the migration test with the binding test.

## Configuration of this document

This manual configuration is as follows.

* Preparation of Chapter 2 test environment

It describes how to prepare the test environment.

* Chapter 3 test procedure

It described the procedure of the test.

* 4 Appendix

And tools (ADB), which seems to be used at the time of the test, described a little bit about how to create an application package (APK).

## the term

In describing the book, in such general terms and difficult to distinguish, the term it feels difficult to understand are listed in Table 12.‑

Table 12 terms‑

|  |  |  |
| --- | --- | --- |
| **No** | **用語** | **意味** |
| 1 | PULL | Android端末に保存されたファイルを、Android端末と接続したPC（Windows等）に移す処理のことです。  ADBでもCUIで実行できますが、本書ではFile ExplorerでGUIとして実行しています。（図 3‑3参考） |
| 2 | PUSH | Android端末と接続したPC（Windows等）から、Android端末にファイルを移す処理のことです。  ADBでもCUIで実行できますが、本書ではFile ExplorerでGUIとして実行しています。（図 3‑3参考） |

# Preparation of the test environment

This chapter describes the preparation of the test environment. Content that depends on CrypgetSQL only Section 2.5, "Preparing for test application". If you have knowledge of the test environment of Android, you do not have a problem by reading only the Section 2.5.

## Preparation of the SDK and ADT

In this section, we describe how to install the SDK and ADT (Android Developer Tools) plug-in to Eclipse.

|  |
| --- |
| 【注意】  本書記載時（2015年4月）現在、IntelliJ IDEA をベースとしたAndroid Studioが公開さいます。今後、このAndroid StudioがGoogle公式のAndroid用IDEになるようです。  しかし、以下の理由から、本書は旧IDEであるEclipseベースのIDEを用いた説明になっております。   * EclipseベースのIDEでCrypgetSQLを試験してきたため * CrypgetSQLの試験アプリがEclipseプロジェクト形式のため * 筆者がAndroid Studio及びIntelliJ IDEAのノウハウを持っていないため   また、Android Studioが公開される以前、GoogleのサイトからSDKやADTを同梱したEclipse（ADTバンドル版Eclipse）が公開されていました。本書の記述はそれを利用したものになっています。しかし、本書記載時、ADTバンドル版Eclipseを公開していたサイトでは、Android Studioに変わってしまったようです。  ADTバンドル版Eclipseと類似の環境を得られると思われるインストール方法について記載しますが、本書を記載するにあたって利用した環境と異なっていることをご了承下さい。 |

1. Installing the SDK

Download the SDK from the place from the following site of the SDK Tools Only, Please install.

<https://developer.android.com/sdk/index.html#Other>

1. Installation of ADT Plugin

The following sites to the reference lists how to install the ADT Plugin.

<http://developer.android.com/sdk/installing/installing-adt.html>

In addition, Eclipse is described on the assumption that you are pre-installed (Eclipse requires 3.7.2 or higher (2015 April)).

* Click the "Help" 竊� "Install New Software" in the Eclipse menu. Click the "Add" button, enter the "https://dl-ssl.google.com/android/eclipse/" and "ADT Plugin" (see Figure 21).‑

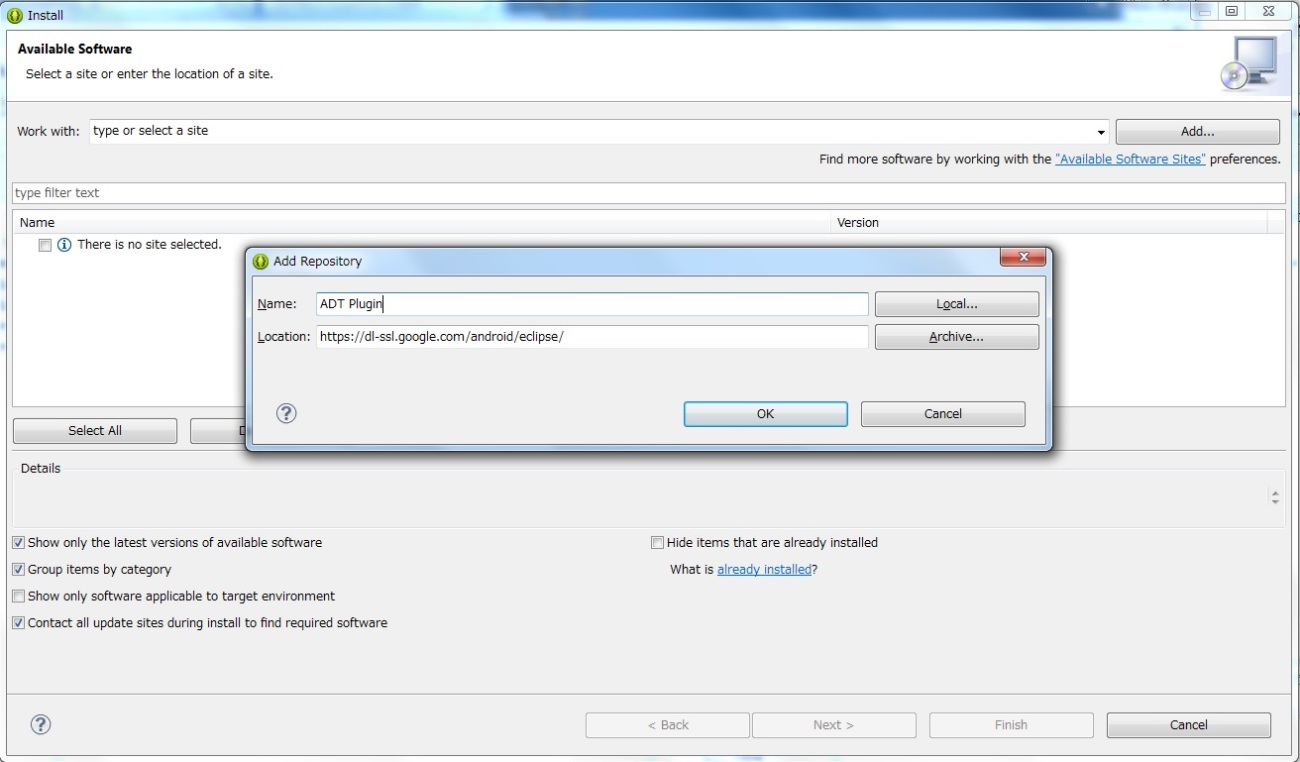


Figure 21 Installing the ADT Plugin‑

* Check the "Developer Tools", and press the Next button. Click Next in the next screen, in the next, do the consent of the license, and then click the Finish button. After installation, restart the Eclipse.
* After the restart, click on the "Window" 竊� "Preferences" in the menu of Eclipse, focus on "Android" in the left side, you must specify the SDK was installed in the above 竭� on the right side of the SDK Location (see FIG. 22) .‑

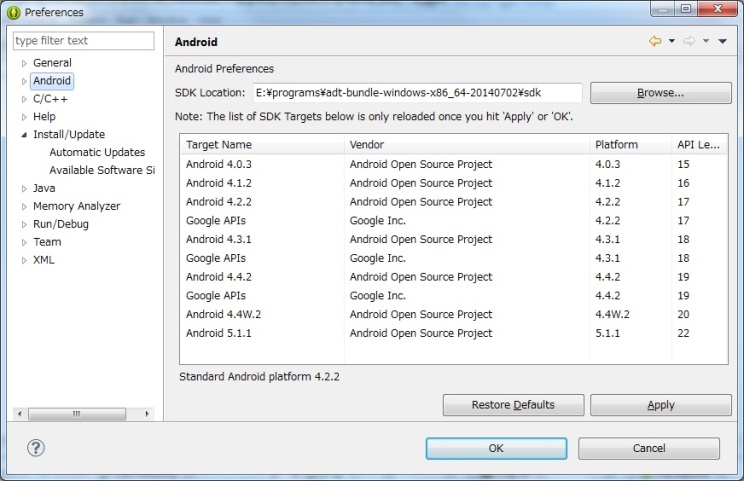


Figure 22 SDK specified Location‑

## Preparing the SDK Platform

In this section, you described for the Android version of the SDK Platform installation of the test.

1. Click on the red circle of the icon of FIG. 23.‑

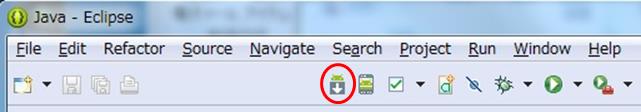


Figure 23 Eclipse menu (SDK Manager icon)‑

1. In the Android version required for the test environment as shown in Figure 24, to check on what you need, click on the "Install x Packages ..." button.‑

Here you are trying to install the environment of Android4.1.2. Installation of "ARM EABI v7a Image" at least "SDK Platform" and the emulator image is mandatory. It should be noted, CrypgetSQL is, of April 2015, does not correspond only to ARM, Intel and MIPS is operating out of warranty.

縲審eference information縲�

If the CPU is in real machine of Intel Atom (x86), binary translation function of the OS built-in and is enabled, also works CrypgetSQL in the automatic conversion, but is guaranteed to work. In the emulator it does not work because this feature is disabled.

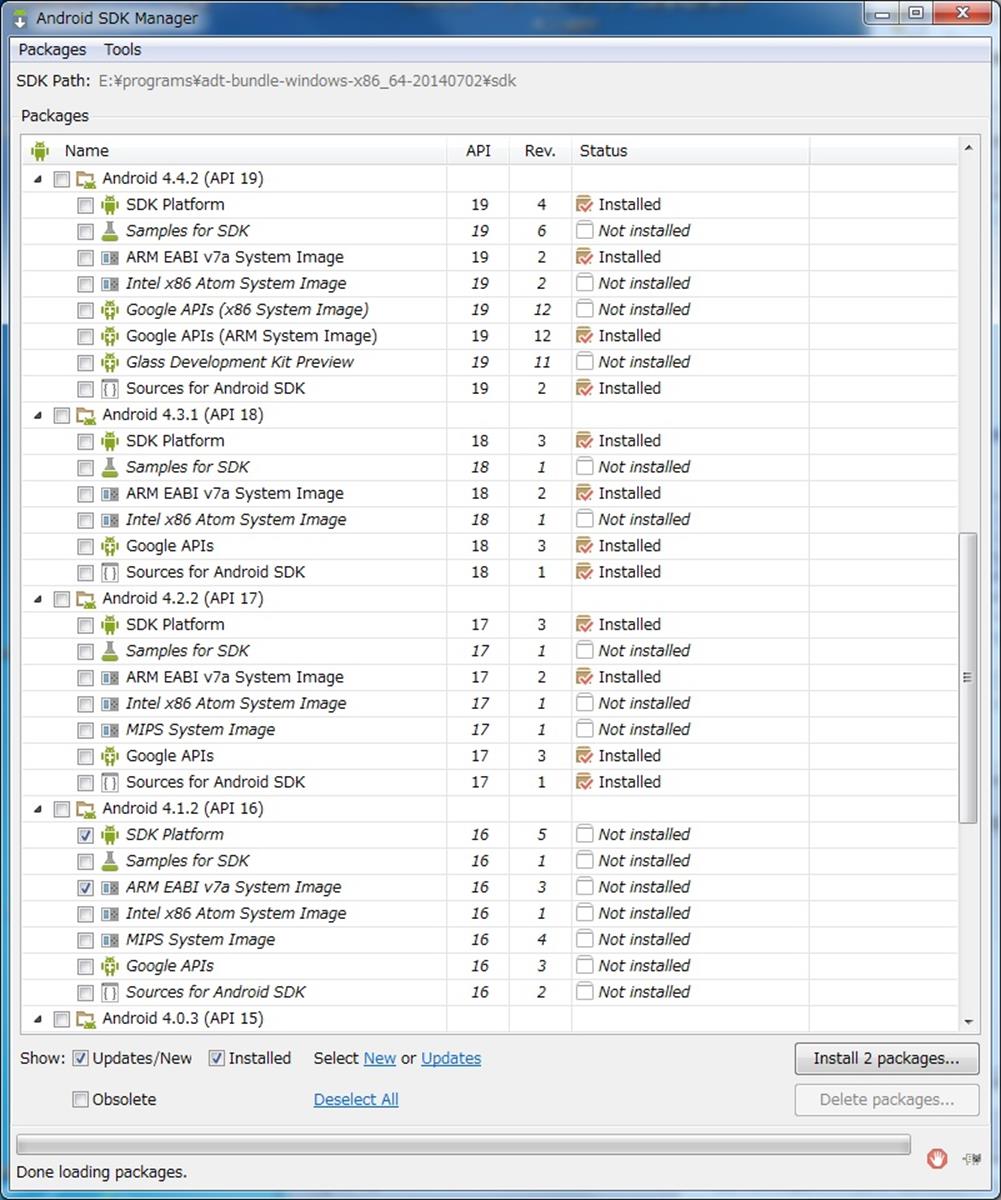


Figure 24 SDK Manager‑

Read the license terms, after selecting the radio button for Accept, to start the installation when the user presses the Install button. It should be noted that, when the installation is complete, you may need to perform a restart of Eclipse.

## Preparing emulator

In this section, it lists up to start the emulator. If you do not want to perform the test using the emulator, please skip the contents of this section. It should be noted, has been described on the assumption that you have finished the installation of the SDK Platform in Section 2.2. In addition, other settings of the emulator (Japanese localization, time zone, proxy) is, please refer to the Section 2.2 of reference [1] as a reference.

1. ﾂ�Click the Android Virtual Device (AVD) Manager of the icon.

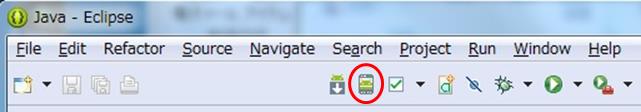


Figure 25 Eclipse menu (AVD Manager icon)‑

1. Click on the "Create ..." button. (See FIG. 26)‑

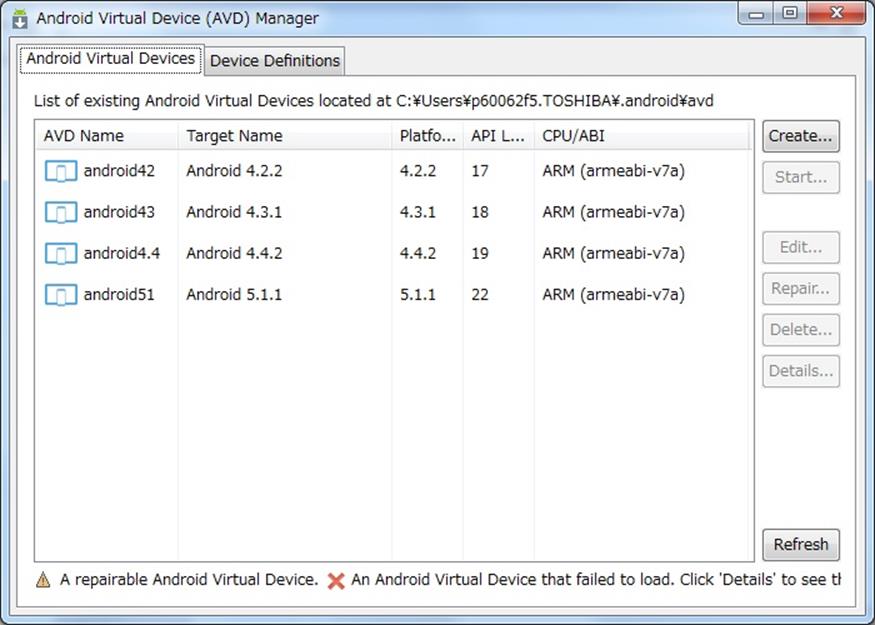


Figure 26 AVD Manager‑

1. Test you want the environment to the mating type, and then click the OK button (see Figure 27). It should be noted, AVD Name is optional. The choice of the Target of the pull-down box, if there is no Android version that you want to test, should be performed to download the SDK to Section 2.2 as a reference.‑

(For details of FIG. 27, "Creating an emulator" Section 2.2.1 of reference [1] will also be helpful.)‑

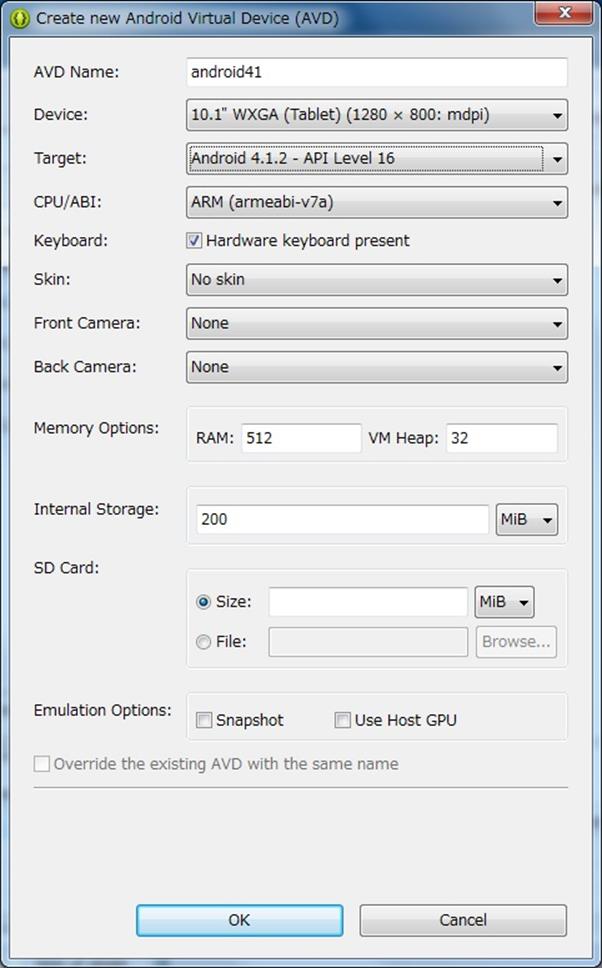


Figure 27 Create new Android Virtual Device‑

1. Since the AVD that was created on the left side of the table of FIG. 26 is displayed, it focuses, emulator When you click the "Launch ..." button on the screen that appears when you click the "Start ..." button will start. (The start-up will take some time.)‑
2. Make sure that the emulator is displayed in the Devices View on the Eclipse (see FIG. 28).‑

(If you Devices View on the Eclipse does not appear, click on the "Window" 竊� "Show View" 竊� "Other ..." in the Eclipse menu, focus on "Android" 竊� "Devices", click the "OK" button To do.)

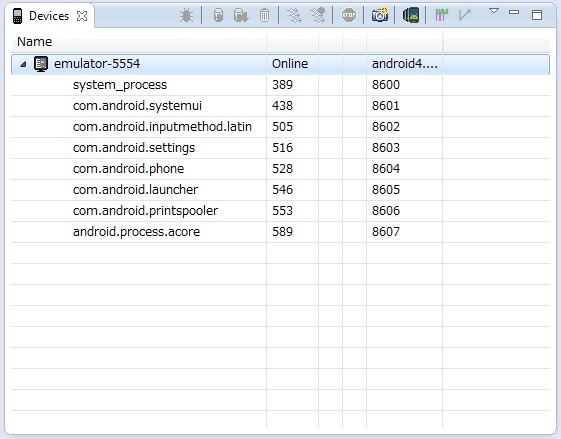


Figure 28 Devices View‑

## Preparation of the actual

And PC (Windows machine) running the Eclipse and the Android terminal and describes a method for USB connection. Content in this section is likely to depend on the actual equipment, please understand that there are many cases which can not be as described.

### Installing the USB driver

Unlike how to install USB drivers for each actual, unified installation method is unknown. How to install the USB driver in accordance with the equipment of the terminal to search the Internet, you must install it in accordance with.

For example, Fujitsu of part of the terminal, the text that describes the USB driver and installation instructions, can be downloaded from the following site.

Example) Fujitsu download site

http://spf.fmworld.net/fujitsu/c/develop/sp/android/

### Setting of the actual machine

In Android actual side, I described how to set so as to enable the USB debugging.

( "Settings" 竊� "Settings" in the case of Japanese-enabled Android devices, as "Developer Options" 竊� "option for developers", "USB Debugging" 竊� "USB debugging", "Build Number" 竊� "build number" will be displayed. as appropriate, it should be replaced.)

1. Start the Android real machine, and then tap the Settings icon (Figure 29 reference) from within the application.‑

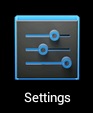


Figure 29 Settings icon‑

1. Tap the "Developer Options", and then check the check box for "USB Debugging" (Figure 210 reference). Sometimes we see a real machine you do not see the "Developer Options" by default 窶ｻ. In that case, tap "About Phone", will be displayed when you tap the "Build Number" 7 times in a row.‑

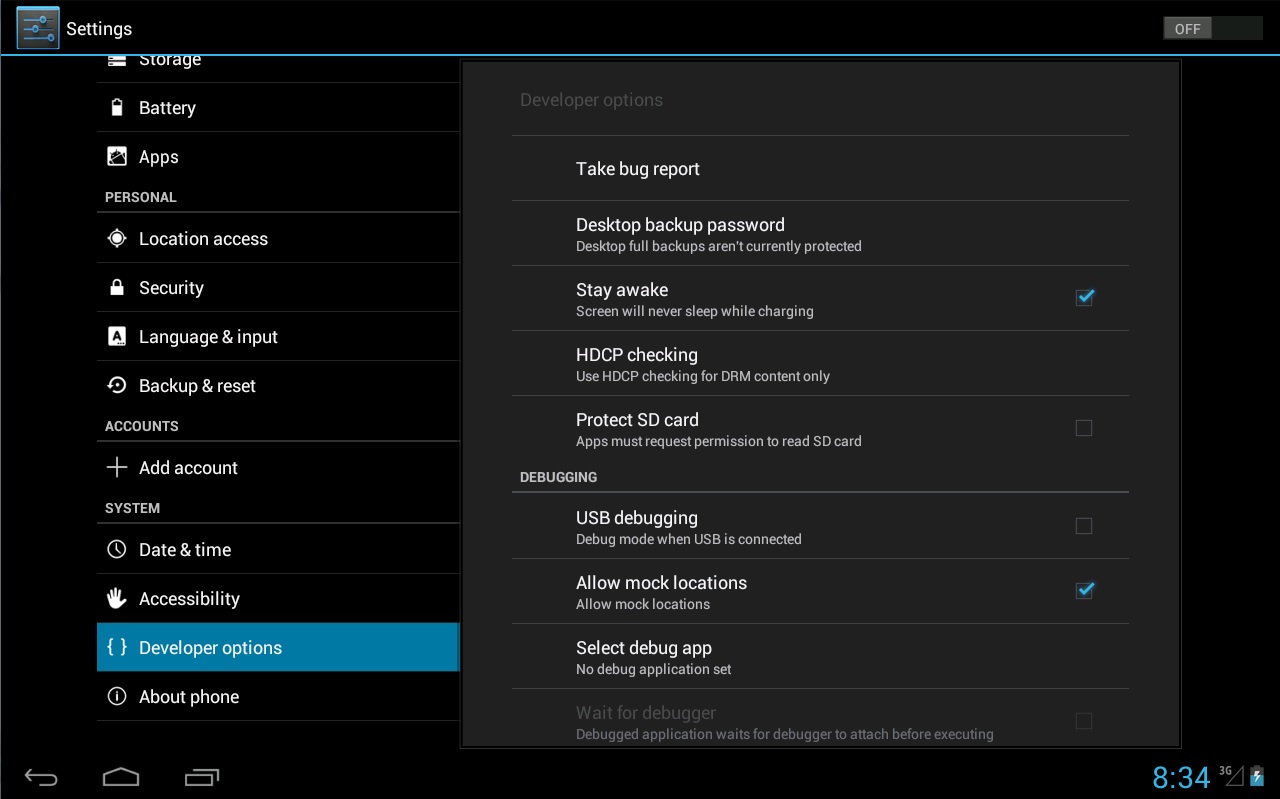


Figure 210 Developer Options‑

1. The Android actual and USB connection. Because at the time of the first connection confirmation dialog of the "Allow USB debugging?" Appears on the screen of the Android Tap the "OK".

### Confirmation of the connection

Windows PC running the Eclipse is, if you recognize the Android actual equipment connected via USB, as in the case of the emulator of FIG. 28, it will appear in the Devices View.‑

## Prepare for the exam app

In this section, we describe how to prepare for the exam application.

### Import of test app

Import the test application in Eclipse.

1. Click on the "File" 竊� "Import" in the Eclipse menu.
2. In the screen that appears, focus on "Android" 竊� "Existing Android Code Into Workspace", and then click the "Next>" button.
3. Click the "Browse ..." button in the Root Directory, select the folder of the test application.

### Set of test app

Set the Android version of the test.

1. Right-click on the project of the Eclipse of the test application, and then click the "Properties" (see FIG. 211).‑

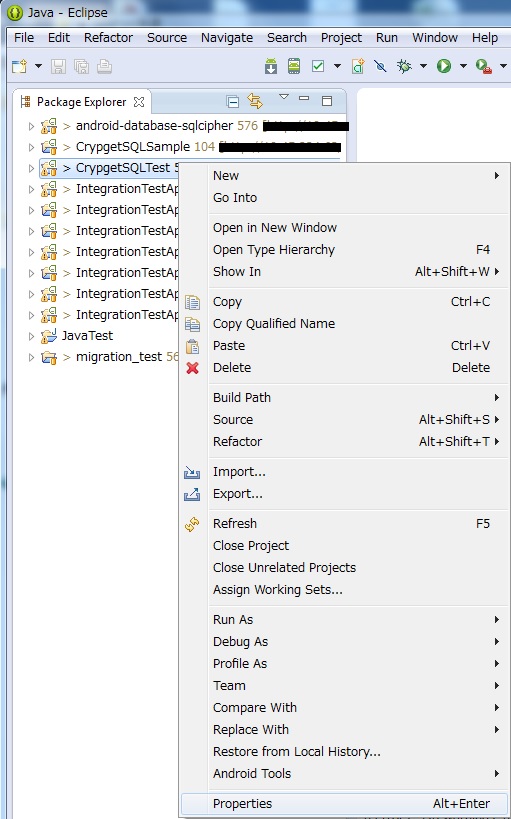


Figure 211 right-click of the Eclipse project‑

1. Click on the "Android" in the left-hand side, and select the "Project Build Target" on the right side (see Figure 212). It should be noted that, if the Android version that you want to test in the "Project Build Target" does not appear, in reference to Section 2.2, you should check that can be installed that version of the SDK Platform.‑

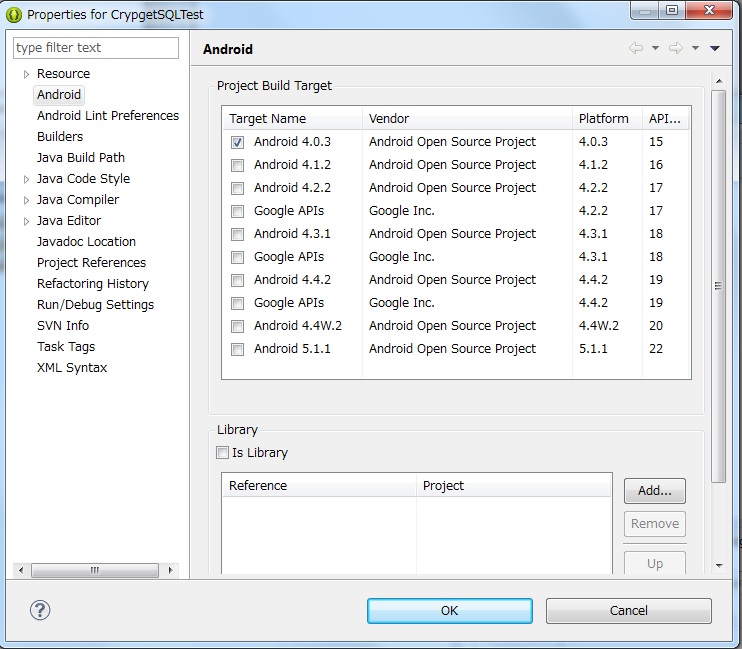


Figure 212 Eclipse Project Properties‑

1. If a compile error, inconsistency of character code settings will be considered. Please make changes and re-compile of character code in accordance with the setting of 4.3 character code.

### Placement of the module

The module to be tested, should be placed in the libs of directly under the root folder of the project of the test application.

# Test procedure

In this chapter, we describe the specific steps of the test described in Section 1.1. In addition, it will assume that you have finished the preparation of the test environment as described in Chapter 2.

## Common procedure

In this section, we describe the two types of how to run the test application. One is how to run the Android JUnit Test, the other one will be how to run the Android Application.

Unit testing (function test, a memory leak test, performance test, multi-threaded test) way of carrying out the will to how to run the exception of some Android Junit Test. Please refer to Section 3.1.1.

How to perform the migration test and binding studies, it will be how to run the Android Application. Please refer to Section 3.1.2.

### Android JUnit Test method of execution

Eclipse of (and or Navigator, Project Explorer) Package Explorer view, right-click the Android JUnit class or test suite file. "Run as" 竊� click the "Android JUnit Test" (see FIG. 31).‑

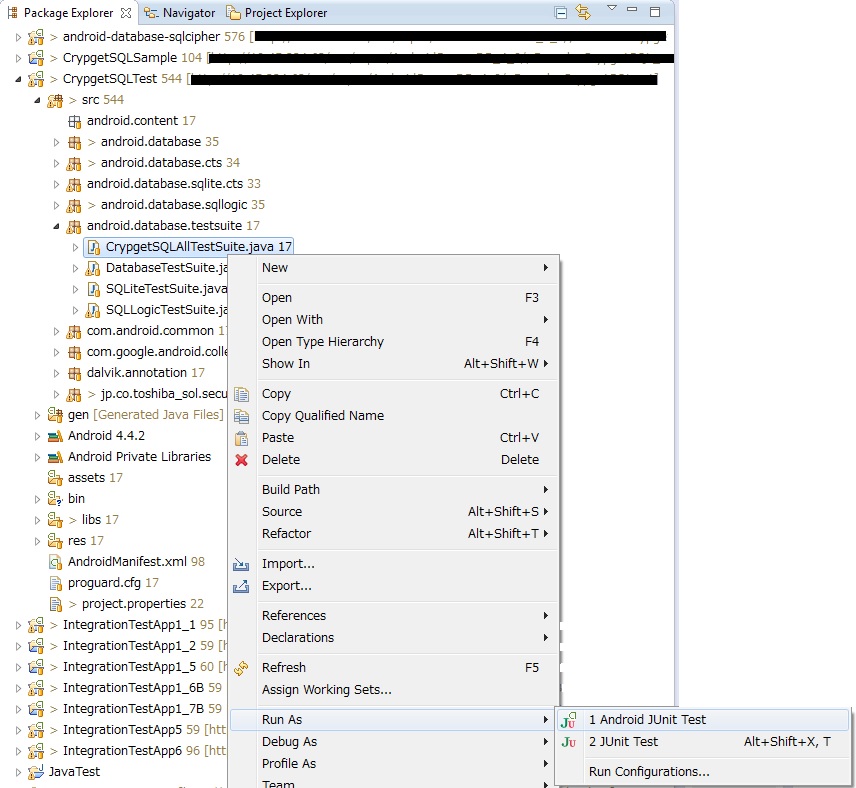


Figure 31 execution of Android JUnit‑

### How to Run Android Application

(And or Navigator, Project Explorer) Eclipse of the Package Explorer view, right-click the Eclipse project. Click the "Run as" 竊� "Android Application" (see FIG. 32).‑

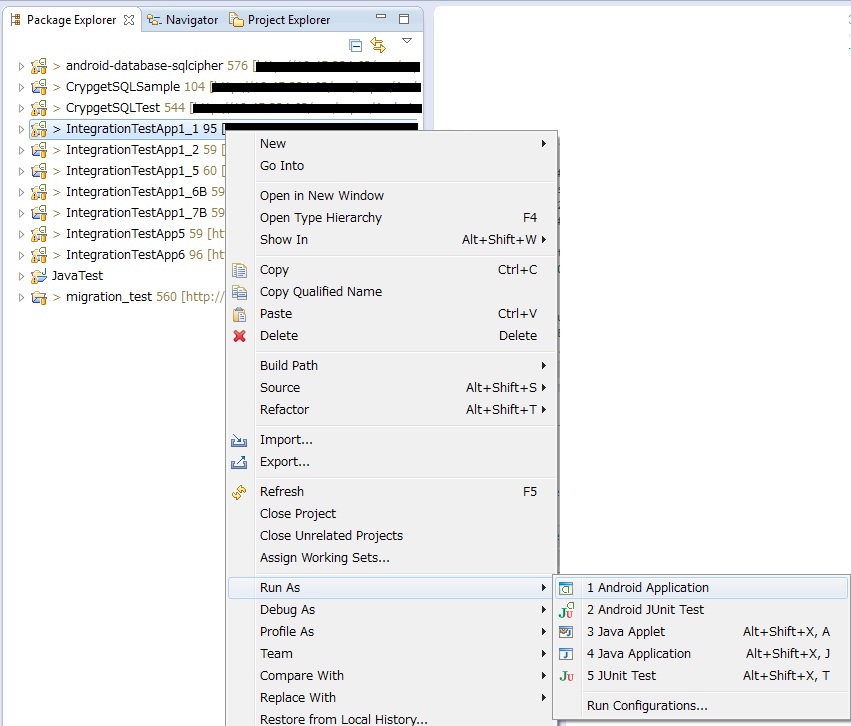


Figure 32 Running the Android Application‑

## Procedure of the functional test

Functional test is divided into SQL test and Java API test.

### Procedure of SQL test

SQL test, except for the crash test described below, have been automated.

1. And the tests that have been automated. Run the file of Table 31.‑

File to run in Table 31 SQL test‑

|  |  |
| --- | --- |
| **V1.1** | android.database.testsuite.SQLLogicTestSuite.java |
| **V2.0/V3.0** | test.jp.co.toshiba\_sol.adnroid.sec.database.TestSuiteSQL.java |

1. We will conduct a crash test.

Crash test generates an error in the transaction, it is a test to confirm whether possible recovery in the journal file. It will be conducted on the emulator.

The file of the table 32 and run twice. The procedures in this study, will be a little more complicated.‑

File to run in Table 32 crash test‑

|  |  |
| --- | --- |
| **V1.1** | android.database.sqllogic.ZZCrashTest.java |
| **V2/V3** | test.jp.co.toshiba\_sol.android.sec.database.sql.ZZCrashTest.java |

* The first time and the second time, you must modify the source code prior to performing each.

The first time, Test Name: crush-2.1 from Test Name: will be carried up to crash-2.4 (see Table 33). The second time, Test Name: will conduct the only crash-2.4 (see Table 34). The red bold comment portion of the table 33 and the table 34, should be modified every time to implement.‑‑‑‑

Table 33 crash test source code of the (first time)‑

|  |
| --- |
| **///\***  // Test Name: crush-2.1  （略）  **// \*/**  //################################################  // 2.4　はクラッシュの際に作成されたデータを使用して実行すること  //################################################  // Test Name: crash-2.4  assertTrue(this.verifySignature(SIGNATURE2)); |

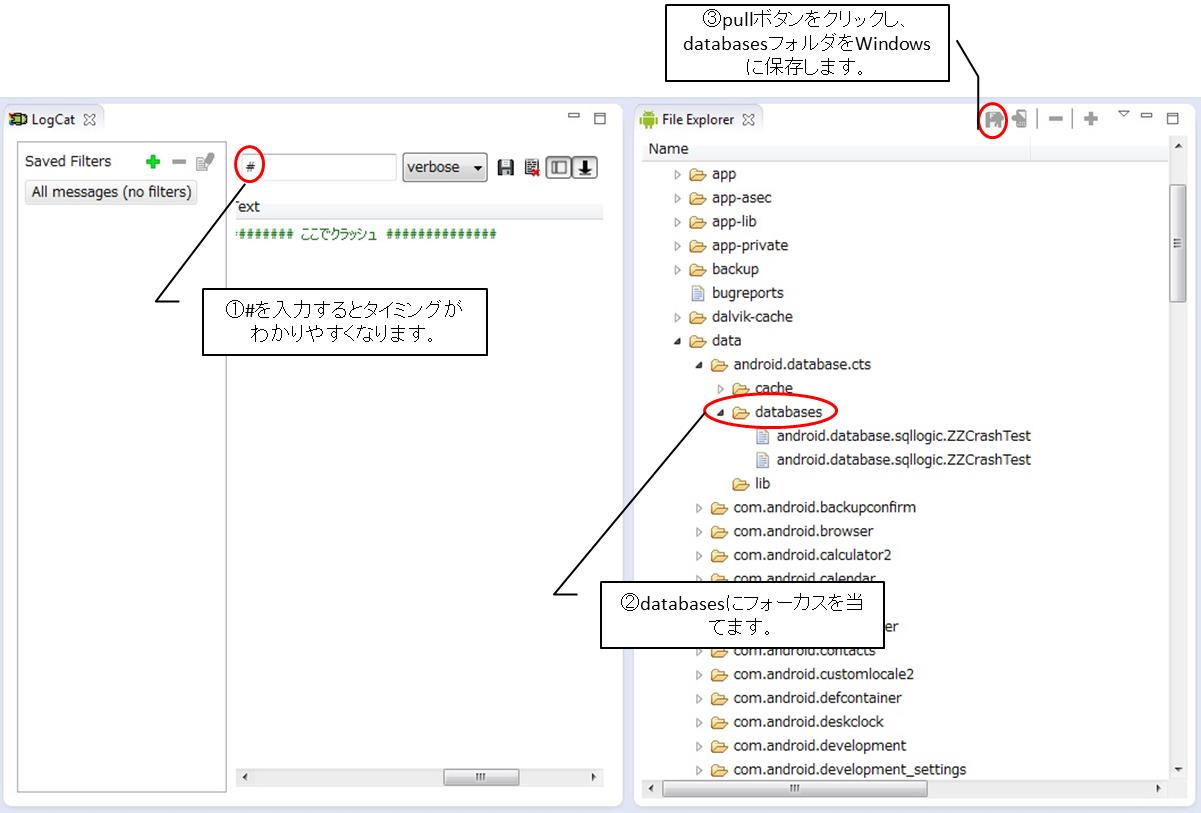
Table 34 crash test source code of the (second time)‑

|  |
| --- |
| **/\***  // Test Name: crush-2.1  （略）  **\*/**  //################################################  // 2.4　はクラッシュの際に作成されたデータを使用して実行すること  //################################################  // Test Name: crash-2.4  assertTrue(this.verifySignature(SIGNATURE2)); |

* During the first run, to get the database and journal files, please use the second time running.

And database files during the first run, to get the journal file. Clearly if you enter "#" to the filter of "######## here crash ##############" is displayed and the timing (Logcat in Logcat in will you), and / data / data / PULL the [package name] / databases (processing to copy the files from the Android device to Windows terminal).

In FIG. 33, to display the Logcat and File Explorer In Eclipse, describes a method for PULL. It should be noted that, if you do not see the Logcat and File Explorer in Eclipse, click on the "Window" 竊� "Show View" 竊� "Other ..." in the Eclipse menu, "Android" 竊� "Logcat" and "Android" 竊� "File focus the Explorer ", click on the" OK "button.‑



Screen when you PULL in Figure 33 crash test‑

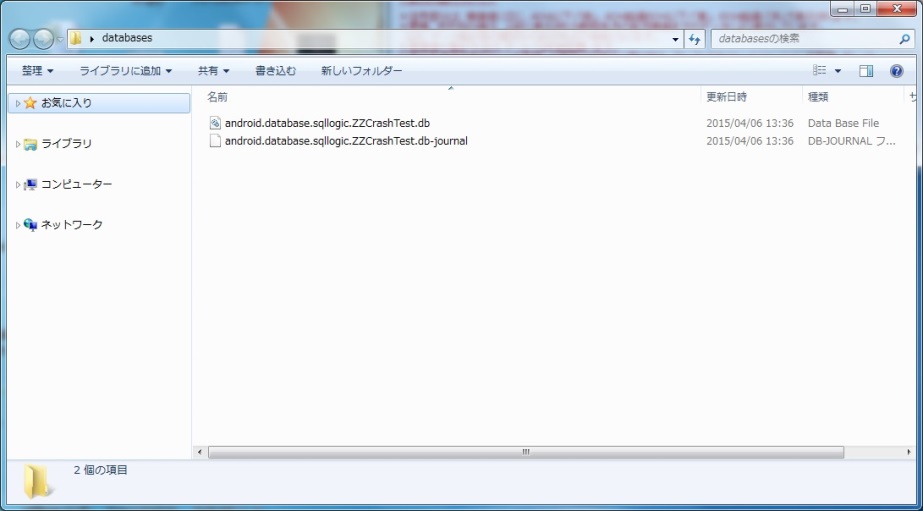


Figure 34 database and journal files that have been stored in the Windows‑

No street is the first test (turns red in JUnit). The second time, the file that you PULL in the first round (save the file on Windows to Android terminal) PUSH in the same location to you. Right next to the pull button shown in FIG. 33, it will be to push the button. Test if you run the part shown in Table 34 passes (green will be displayed in the JUnit). If the green does not appear, there is a possibility was bad timing PULL in the first round. Again, please re-implemented from the first time.‑‑

### Procedure of Java API test

Java API test, has been fully automated. Run the JUnit, make sure that a successful completion.

Table 35 file to run in the Java API test‑

|  |  |
| --- | --- |
| **V1.1** | android.database.testsuite.DatabaseTestSuite.java  android.database.testsuite.SQLiteTestSuite.java |
| **V2/V3** | test.jp.co.toshiba\_sol.android.sec.database.TestSuiteCTS.java  test.jp.co.toshiba\_sol.adnroid.sec.database.TestSuiteCTS2.java |

## Memory leak test

ﾂ�Memory leak test, and an object of the present invention the memory leak test (memory usage be used for a long time CrypgetSQL confirmation to that there is no increasing trend) is a test of the street (Section 3.3.1), the number of pages in the database file There are divided into Corrupt test to confirm whether withstand to change (Section 3.3.2).

### Memory leak test

In the memory leak test, a typical SQL (CREATE TABLE statement, SELECT statement, INSERT statement, UPDATE statement, DELETE statements) has been run repeatedly (from the fact that also serves as an endurance test, possibility of releasing the resources there DROP TABLE are excluded).

Prior to the test carried out, should be adjusted as necessary number of loops in the source code (this study, by the actual performance to be used in the test, the test will be excess and deficiency). It can be adjusted by changing the value of the variable OUT\_LOOP\_NUM of testMemoryleak method. Before and after the loop processing, and explicitly GC treatment, but there is a Sleep for 30 seconds, in the process for the test's convenience (loop processing easy to check the memory usage of the front and rear), the essence of the test is It does not matter. It does not matter if I have commented out.

If the increase is Miataru to memory usage, but then analyzed using the MAT (Memory Analyzer), please refer to the reference [1] is also about the MAT.

Table 36 file to run in the memory leak test‑

|  |  |
| --- | --- |
| **V1.1** | jp.co.toshiba\_sol.securedb.test.SQLMemoryLeakTest.java |
| **V2/V3** | CrypgetSQL V2/V3ではメモリリーク試験アプリは、Android JUnitではなくAndroid Applicationになります。MemoryleakTestを利用します。 |

Memory leak test of CrypgetSQL V2 / V3, you can use the MemoryleakTest is an Android Application. The application screen button on the not only one of the "start". Tap the button, and you're done is displayed as "finish." On the screen. After completion, please get the "CrypgetSQL\_Memoryleak\_ [date] \_ [time] .txt" files from external storage (see Figure 35). You use the File Explorer, but this method of use, please refer to the peripheral figure 33. In addition, the output point of this text file is terminal dependent.‑‑

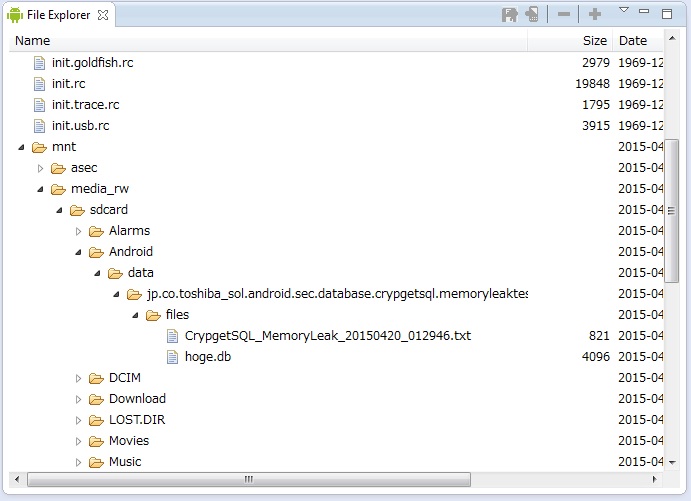


Figure 35 Memoryleak result text‑

The contents of this file will be the CSV, such as shown in Table 37. Contents of the first line has been described. In Excel, etc., by, for example, to graph, the increase or decrease of memory and visualization, make sure the presence or absence of a memory leak.‑

Table 37 Memoryleak result text Details‑

|  |
| --- |
| 2015-04-20-01:29:46.125,app\_dalvik\_total,app\_dalvik\_alloc,app\_native\_total,app\_native\_alloc  2015-04-20 01:29:46.154,3153920,3057008,1970176,1963424,beforeOpen  2015-04-20 01:29:50.281,3383296,3171088,2162688,1991744,end of 0  2015-04-20 01:29:53.120,3592192,3047776,2162688,1993792,end of 1  2015-04-20 01:29:55.532,3592192,3447656,2166784,1996392,end of 2  2015-04-20 01:29:57.742,3592192,3331240,2166784,1993864,end of 3  2015-04-20 01:29:58.979,3600384,3201920,2166784,1993888,end of 4  2015-04-20 01:30:00.528,3600384,3078440,2166784,1993888,end of 5  2015-04-20 01:30:02.746,3600384,3478312,2166784,1996488,end of 6  2015-04-20 01:30:04.962,3600384,3361368,2166784,1993888,end of 7  2015-04-20 01:30:07.361,3600384,3233816,2166784,1993888,end of 8  2015-04-20 01:30:09.316,3600384,3110008,2166784,1993888,end of 9 |

### Corrupt test

There is a header on the page size of the database file, page size after encryption is 0x0200 (512), if they represent, such as 0x0800 (2048), CrypgetSQL is, regarded as a database file is not encrypted (really is encrypted), there was a bug that accurately can not be file operation earlier. Although the bug has been fixed, in the present study, it is modified correctly, make sure that the same bug is not reproducible. Run the JUnit, make sure that a successful completion.

File to run in Table 38 Corrupt test‑

|  |  |
| --- | --- |
| **V1.1** | jp.co.toshiba\_sol.securedb.test.CorruptTest.java |
| **V2/V3** | test.jp.co.toshiba\_sol.android.sec.database.performance.CorruptTest.java |

## Procedure of the performance test

Performance test, a test to measure the SQL statement 16 kinds of performance, and a test to measure the performance of such library load, there is a test to measure the data by size of the performance.

SQLCrypgetPerformanceTest.java of Table 39 is a test to measure the SQL statement 16 kinds of performance. SQLCrypgetPerformanceTest2.java is a test to measure the performance of such library load. CrypgetSQLPerformanceTrendTest.java is a test to measure the data by size of the performance.‑

File to run in Table 39 performance test‑

|  |  |
| --- | --- |
| **V1.1** | jp.co.toshiba\_sol.securedb.test.SQLCrypgetPerformanceTest.java  jp.co.toshiba\_sol.securedb.test.SQLCrypgetPerformanceTest2.java  jp.co.toshiba\_sol.securedb.test.CrypgetSQLPerformanceTrendTest.java |
| **V2/V3** | test.jp.co.toshiba\_sol.android.sec.database.performance.SQLitePerformanceTest.java  test.jp.co.toshiba\_sol.android.sec.database.performance.SQLitePerformanceTest2.java  test.jp.co.toshiba\_sol.android.sec.database.performance.SQLitePerformanceTrendTest.java |

## Procedure of multi-threaded test

Test of availability of CrypgetSQL in a multi-thread, only to run the file of the table 310. In particular, it does not describe the description.‑

Table 310 file to run in a multi-threaded test‑

|  |  |
| --- | --- |
| **V1.1** | jp.co.toshiba\_sol.securedb.test.CrypgetSQLMultiThreadTest.java |
| **V2/V3** | test.jp.co.toshiba\_sol.android.sec.database.performance.CrypgetSQLMultiThreadTest.java |

## Procedure of binding test

Binding studies, it does not have can be automated. In the order of the test specification, and describes the intent of the procedure and testing of test for each test item. It should be noted that the binding test application, we become Android Application. If the execution method is unknown, please refer to Section 3.1.2.

### Test number 1.1.X

* The contents of the test

Make sure that you can use the CrypgetSQL from Android applications that do not perform a screen transition.

* Test app

You use the IntegrationTestApp1\_1 (see FIG. 36). Description of the buttons are described in Table 311.‑‑

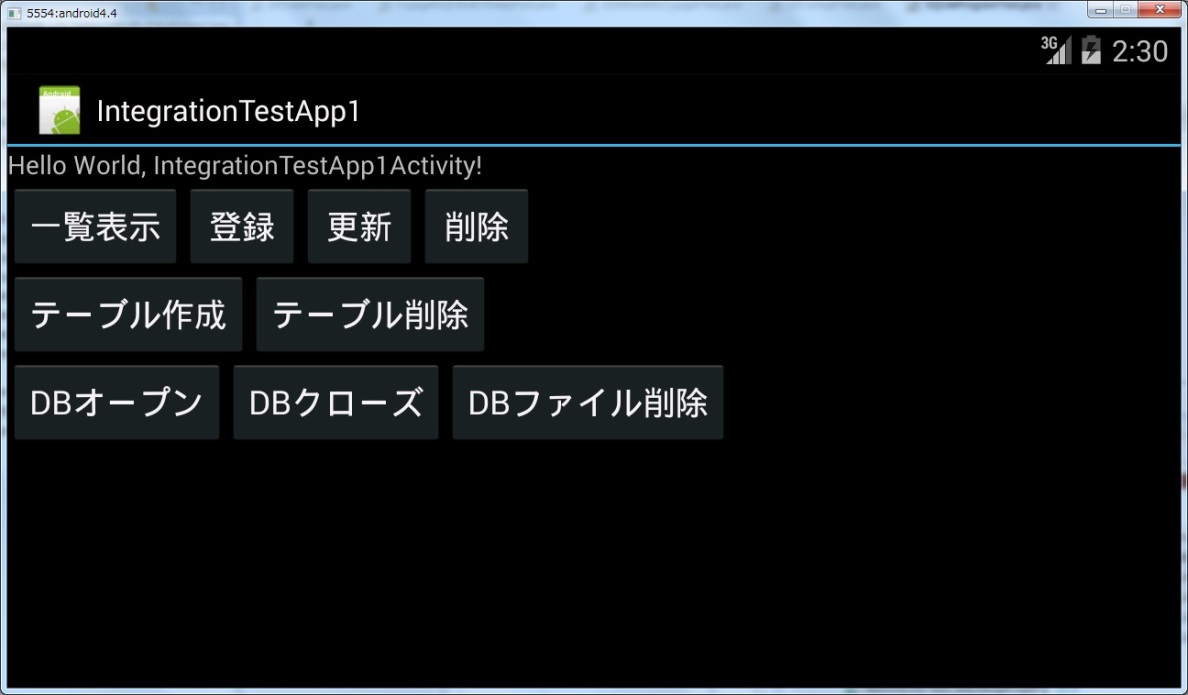


Figure 36 IntegrationTestApp1\_1 screen of‑

Table 311 description of the buttons of IntegrationTestApp1\_1‑

|  |  |  |
| --- | --- | --- |
| **No** | **ボタン** | **処理の内容** |
| 1 | 一覧表示 | 登録されているデータを検索し、画面上に表示します。データは「DBオープン」ボタンの下に表示されます。データが登録されていない場合は、表示されません。 |
| 2 | 登録 | 4件のデータを登録します。「東芝太郎、45」「東芝花子、40」「浜松太郎、35」「浜松花子、30」が登録されます。複数回タップすれば、タップした分だけ4件ずつ追加登録されます。 |
| 3 | 更新 | 「東芝太郎、45」を「東芝次郎、20」に更新します。 |
| 4 | 削除 | 登録されているデータを一括削除します。 |
| 5 | テーブル作成 | テーブルを作成します。 |
| 6 | テーブル削除 | テーブルを削除します。 |
| 7 | DBオープン | DBをオープンします。 |
| 8 | DBクローズ | DBをクローズします。 |
| 9 | DBファイル削除 | データベースファイルを削除します。 |
| 10 | 画面遷移 | 別の画面を遷移します。  IntegrationTestApp1\_1には当ボタンはありません。後述のIntegrationTestApp1\_2、IntegrationTestApp1\_5にあります。 |
| 11 | 前画面に戻る | 前の画面に遷移します。  IntegrationTestApp1\_1には当ボタンはありません。後述のIntegrationTestApp1\_2、IntegrationTestApp1\_5にあります。 |

Below, lists the normal operation example of IntegrationTestApp1\_1 (see Figure 37).‑

The first diagram from the left of FIG. 37, to start the application "DB open" 竊� "Create Table" 竊� thing you tap the "list". In addition, "registration" 竊� what you tap the "list" is the second diagram from the left. This figure further "update" 竊� what you tap the "list" is the third from the left. This figure further "Delete" 竊� what you tap the "list" is fourth from the left.‑



Use the example of FIG. 37 application (Part 1)‑

This describes another normal operation example. If the data is tap the button in the order of "registration" 竊� "Update" 竊� "Register" 竊� "List" from the state that are not registered, you look like the one shown in Figure 38.‑



Use the example of FIG. 38 application (Part 2)‑

It should be noted, and the state have not created a table, in a state that does not have DB open, when the "list" and "registration", "update", "Delete", an error occurs, the application will stop.

* The test procedure

Prior to performing the test, make sure that the IntegrationTestApp1\_1 is not installed on the terminal.

IntegrationTestApp1\_1 Install and launch new in terminal, it should be performed to order the operation of each test item in the test specification. It eliminates the explanation for it is not difficult.

### Test number 1.2.X

* The contents of the test

Make sure you can use the CrypgetSQL from the Android application that there is a screen transition.

* Test app

You use the IntegrationTestApp1\_2. IntegrationTestApp1\_1 (Figure 36, Table 311) and is roughly the same.‑‑

* The test procedure

Prior to performing the test, make sure that the IntegrationTestApp1\_2 is not installed on the terminal.

IntegrationTestApp1\_2 Install and launch new in terminal, it should be performed to order the operation of each test item in the test specification. It eliminates the explanation for it is not difficult.

### Test number 1.3.X

* The contents of the test

Make sure you can use the CrypgetSQL from Android applications that do not perform a screen transition.

* Test app

You use the IntegrationTestApp1\_1.

* The test procedure

As of the test specification, and press the button. It eliminates the explanation for it is not difficult.

* Remarks

The difference between the test number 1.1.X is, from where already IntegrationTestApp1\_1 is installed on the Android device, and start the application.

Since being installed at the time of test of test number 1.1.X, to manipulate the screen of the Android device, there is no problem if the start-up once kill again.

Forced termination, tap the Settings (see FIG. 29), and then tap the "Apps" from the left side (located on eight of the Developer Options in Figure 210). Since the application is listed, the application information screen that appears when you tap the name of the application from there (see Figure 310, however, the screen is of IntegrationTestApp1\_5) Tap the "Force Stop" button.‑‑‑

The start-up, when you tap the button, such as shown in FIG. 39, because of the installed application icon is a list, you should start by tapping the IntegrationTestApp1\_1 icon.‑



Figure 39 button that lists the application's icon‑



Figure 310 application information screen‑

### Test number 1.4.X

* The contents of the test

Make sure you can use the CrypgetSQL from the Android application that there is a screen transition.

* Test app

You use the IntegrationTestApp1\_2.

* The test procedure

As of the test specification, and press the button.

* Remarks

The difference between the test number 1.2.X is, from where already IntegrationTestApp1\_2 is installed on the Android device, and start the application.

Since being installed at the time of test of test number 1.2.X, to manipulate the screen of the Android terminal, once killed, there is no problem if start-up again. For Force how to start and how to restart the application, please refer to the remarks of the previous section (Section 3.6.3).

### Test number 1.5.X

* The contents of the test

Make sure you can use the CrypgetSQL from the Android application that there is a screen transition.

* Test app

You use the IntegrationTestApp1\_5.

* The test procedure

Please operate following the procedures of the test specification. A place that has been described as "(confirmed by both A screen screen B)" is to make sure the contents field, tap the "screen transition" and "go back to the previous screen" button, make sure from two screens. So that there is in the remarks column of the later, in this test application, it has been operating one of the database from the two screens. Operating in A screen, after the transition to the B screen, tap the "List" button, make sure that the operation of the A screen is also reflected in the B screen.

* Remarks

This is similar to test number 1.2.X, but the test number to operate on the same database from multiple screens in 1.5.X, except you are working with multiple screens separate database in test number 1.2.X.

### Test number 1.6.X

* The contents of the test

Make sure that you can use function that can operate the database of another application from one application (ContentProvider) is.

* Test app

We use the IntegrationTestApp1\_1 and IntegrationTestApp1\_6B.

* Test procedure

IntegrationTestApp1\_1 will operate the IntegrationTestApp1\_6B with terminals operating in the background. For details on how to operate IntegrationTestApp1\_6B, it omits the explanation for it is not difficult. To operate in the background IntegrationTestApp1\_1, was carried out to start the usual (Section 3.1.2 Reference) IntegrationTestApp1\_1, then, if the start the IntegrationTestApp1\_6B, IntegrationTestApp1\_1 will be in the background.

### Test number 1.7.X

* The contents of the test

Applications that have the same UID as the application that created the database file, make sure that the database file can be manipulated.

* Knowledge of the background

UID and is the user ID in Linux terms. In Android, UID is assigned to each application. Originally, the CrypgetSQL, database file that was created in one application can not be operation can not be decrypted (decoded) in another application. However, the same case of the UID, (you need to UID not only the same terminal, such as the conditions are met) an application it is possible to manipulate the database file that was created from another application.

It should be noted that, in the same the UID should be the same as the value of the sharedUserId in the Android manifest file for the application (AndroidManifest.xml directly under Eclipse project). The sharedUserId, you need to include one or more periods.

It is to be noted that the application of the UID, can be found in the ADB (4.1 Section Reference) (red bold UID of the table 312). Usually, su command can be used in the emulator, but can not be used in the actual equipment. In this case it can not be confirmed.‑

UID how to check the application of the table 312 adb‑

|  |
| --- |
| > adb shell  > ps | grep [アプリケーションのパッケージ名]  u0\_aXX …  > su u0\_aXX  > id  uid=**xxxxx**(u0\_aXX) gid=xxxxx(u0\_aXX) |

* Test app

And tested using the IntegrationTestApp1\_1 and Integration.app1\_7B. It is the same as the value of the sharedUserId "jp.co.toshiba\_sol.android.test".

* Test procedure

1. Register the some data in IntegrationTestApp1\_1.
2. And PULL the sqlcrypget.db from IntegrationTestApp1\_1. PULL can be in the File Explorer. Please refer to Figure 33.‑
3. And PUSH the sqlcrypget.db acquired in 竭｡ to IntegrationTestApp1\_7B. PUSH is available in File Explorer. Please refer to Figure 33.‑
4. After pressing the "DB Open" button in IntegrationTestApp1\_7B, perform the operation described in the test specification.
5. And PULL the sqlcrypget.db from IntegrationTestApp1\_7B, and PUSH it to IntegrationTestApp1\_1.
6. After pressing the "DB Open" button in IntegrationTestApp1\_1, to verify whether the operation of the above 竭｣ in the "List" button is reflected.

### Test number 2.X

CrypgetSQL Make sure you are able to encrypt the database. Open the database file in a binary editor, etc., make sure that there is no readability. Database files, you can also get in the File Explorer using at the time of the crash test (see Figure 33). Check the contents of the database file that you PULL on the PC side in the binary editor. There is no designated especially for binary editor, but here is using the BZ. Make sure that there is no readability on the right side of the figure 311. And described in the test number 3.X on the next page for the Seed and IV.‑‑

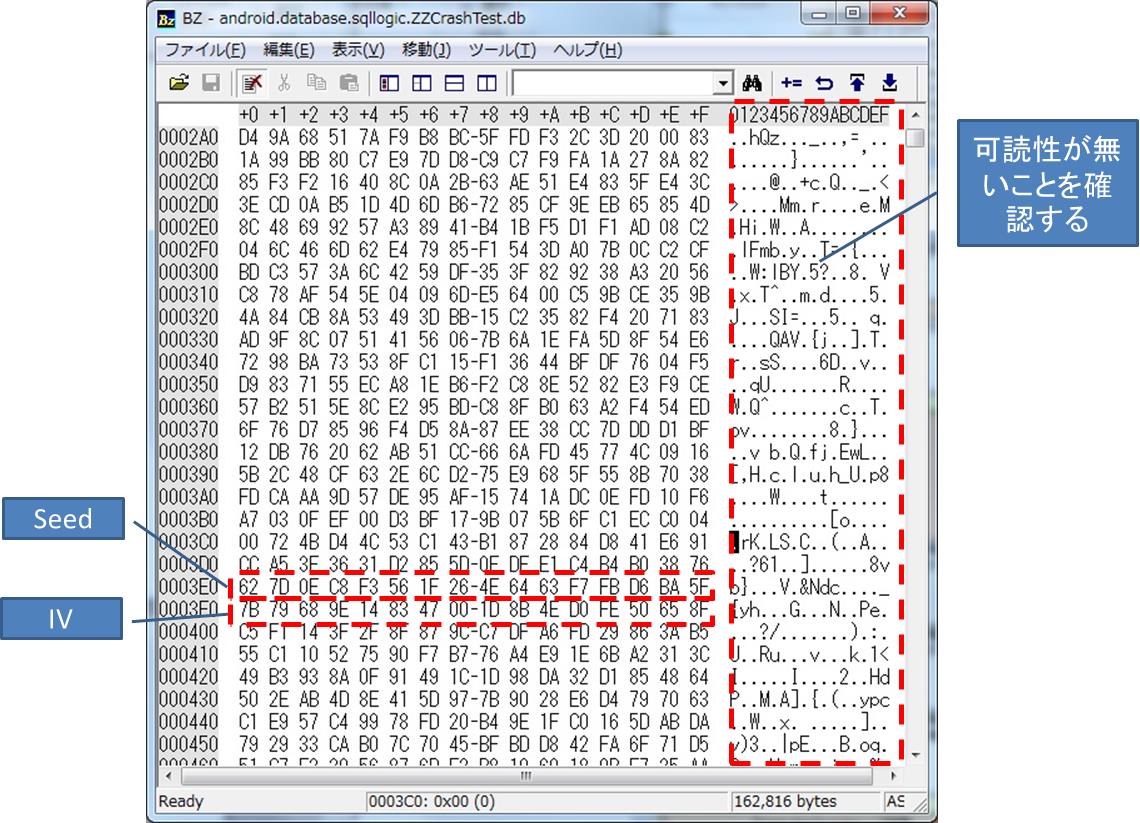


Figure 311 database files that were opened in the binary editor‑

### Test number 3.X

* The contents of the test

Key management of CrypgetSQL is to make sure as intended. Test number 2.X Like, read the database file in a binary editor.

* Knowledge of the background

CrypgetSQL has embedded the Seed (key seed) and IV (initial vector) to the page of the database file (see Figure 312). Page size of CrypgetSQL is 1024byte fixed. The number of pages (the size of the 竕� database file), the more the more data you have saved. Last 16byte the IV of each page, the previous 16byte will be the Seed. In addition, if you use a binary editor BZ, Please note to be converted 1024 (decimal) 竊� 400 (16 decimal) (see Fig. 311).‑‑

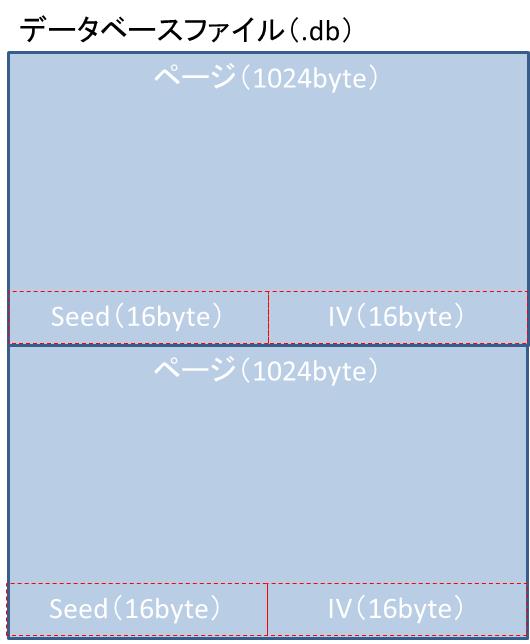


Figure 312 of the database file structure‑

* The test procedure

Follow the test specification, please get the database file. Opened in binary editor, please compare the database file obtained. Acquisition of database files can be in the File Explorer (right-hand reference in FIG. 33).‑

It should be noted that, since the test number 3.1.2 and 3.1.3 of the operation is the same, there is no problem using the three database files obtained by the test number 3.1.2 test number 3.1.3.

For example, the test number 3.1.2, to get the three database files, make sure that the IV is different from each other. It is hard to read, but you can see that the figure 313 IV each is different. Database file name, there is no problem to change when you save to the PC of Windows such as in the File Explorer. 992 bytes part before the IV in test number 3.1.2 Make sure that you are different.‑

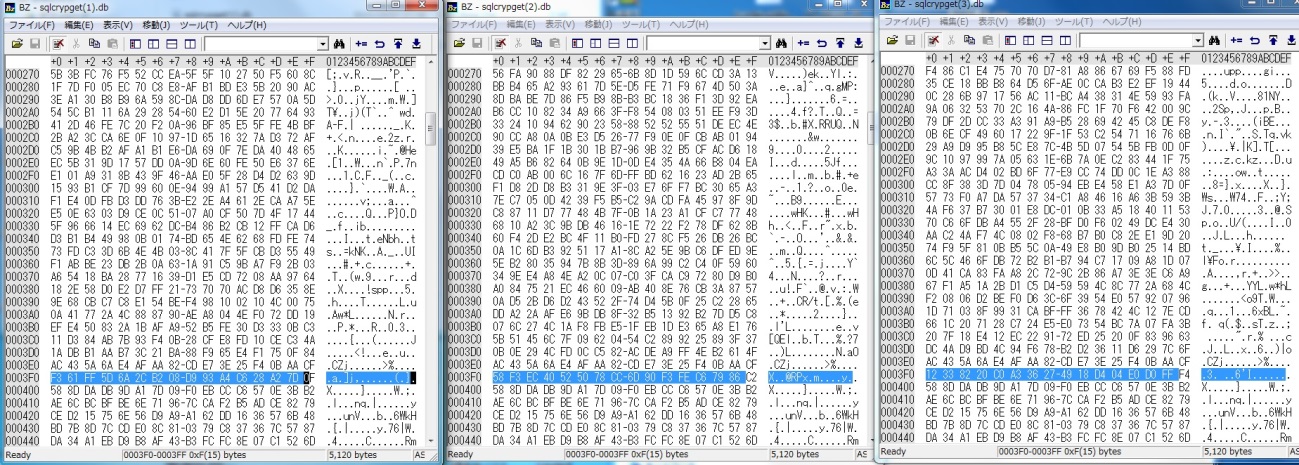


Figure 313 Comparison of database files‑

### Test number 4.X

* The contents of the test

Database file that was created on a terminal is to confirm that you can not operate in a different terminal.

* Background of test

CrypgetSQL has a terminal authentication function, you can not open the database file that was created on a certain terminal in a different terminal. In this test, to see if this feature is working properly.

* Test app

You use the IntegrationTestApp1\_1.

* Notes

In the case of emulator, terminal ID identifying the terminal will be null. When tested with two emulator, are recognized as the same terminal, it will be opened.

In the real machine, it can not be accessed from the File Explorer / data / data / later file. And tested using the SD card (/ mnt / sdcard /).

**窶ｻ path of SD card may vary by Android actual equipment. Modify as appropriate.**

* The test procedure

1. Run the IntegrationTestApp1\_1 in the emulator.

/ Data / data / [package name] /databases/temp.db Please PULL the. Package name will be the following.

jp.co.toshiba\_sol.android.sec.database.crypgetsql.test.integration.app1

1. Modify the source code.

In the following files of the src of IntegrationTestApp1\_1, it should be modified in accordance with the table 313 the processing of onCreate method. At the red bold is the correction point. (You have to change the comment)‑

jp.co.toshiba\_sol.android.sec.database.crypgetsql.test.integration.app1. IntegrationTestApp1Activity.java

Table 313 modification of the source code (test number 4.X)‑

|  |  |
| --- | --- |
| **修正前** | // - ▼ - 試験No.4用コード  File dbFile**//**= new File("/mnt/sdcard/", "temp.db"); //実機用  = new File(  "/data/data/jp.co.toshiba\_sol.android.sec.database.crypgetsql.test.integration.app1/databases/", "temp.db"); //エミュレータ用 |
| **修正後** | // - ▼ - 試験No.4用コード  File dbFile = new File("/mnt/sdcard/", "temp.db"); //実機用  **//**= new File(  "/data/data/jp.co.toshiba\_sol.android.sec.database.crypgetsql.test.integration.app1/databases/", "temp.db"); //エミュレータ用 |

1. Of a real machine / mnt / sdcard /, please PUSH the temp.db was PULL above 竭�.
2. Start the IntegrationTestApp1\_1 that was fixed in the above 竭｡ the actual machine.

Look at the Locat, make sure that the following exception has occurred.

Exception class: jp.co.toshiba\_sol.android.sec.database.crypgetsql.SQLiteException

Exception message: file is encrypted or is not a database

### Test number 5.X

* The contents of the test

Database file that was created in one application, make sure that can not be operated from another application. However, test number 5.1.3, in the same application, when you update, make sure that you can operate.

* Background of test

CrypgetSQL has a application authentication function, you can not open the database file that was created on one application in another application. In this test, to see if this feature is working properly. However, test number 5.1.3, in the same application, when you update, make sure that you can operate.

* Test application

IntegrationTestApp1\_1 and then tested in the IntegrationTestApp5.

* The test procedure

On which to understand the notes described below, it should be carried out as the test specification. You can in the File Explorer. Please refer to Figure 33.‑

Test No. 5.1.3, on Eclipse, to change some of the source code, and then try again.

System.out.println in the third row of the onCreate method of jp.co.toshiba\_sol.android.sec.database.crypgetsql.test.integration.app1.IntegrationTestApp1Activity ( "test number 5.1.3"); because there, it Please comment out (or comment in).

* Notes

In the test number 5.1.2, Android4.3 and earlier, after you uninstall the application, becomes also the installation again UID (see 3.6.7 Section) are the same, it will open the database file. After uninstall, install some applications, you must install the application (IntegrationTestApp1\_1 here) required for the test.

### Test number 6.X

* The contents of the test

Make sure the behavior in password difference of CrypgetSQL.

* Test application

IntegrationTestApp6 and tested using. It lists the description of IntegrationTestApp6.

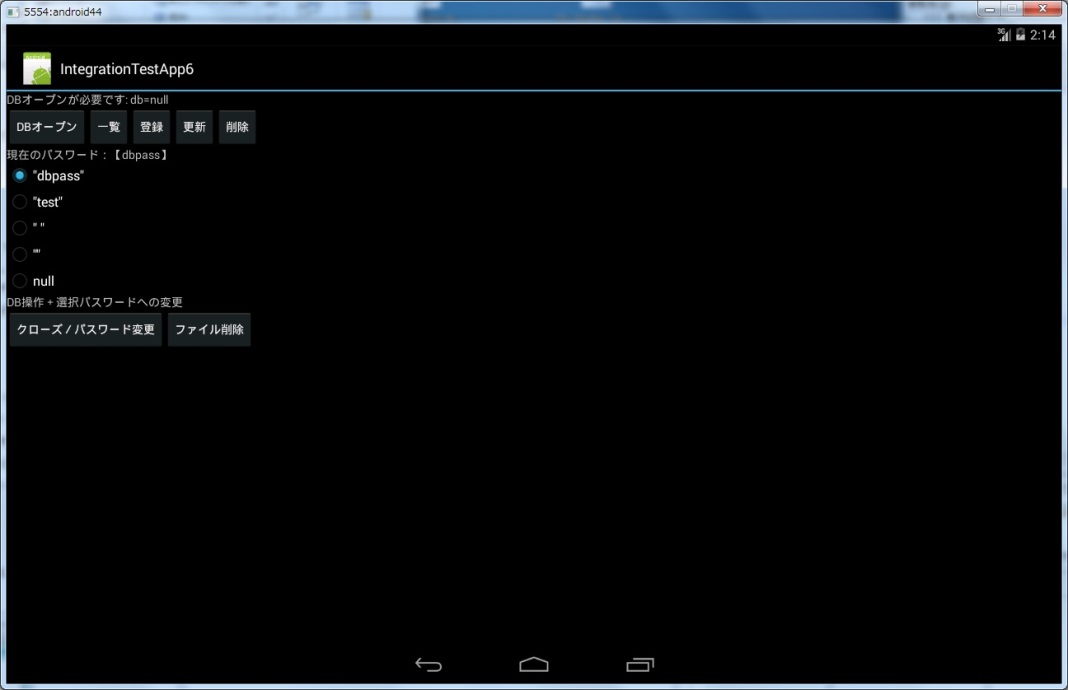


Figure 314 IntegrationTestApp6‑

Table 314 description of the buttons of IntegrationTestApp6‑

|  |  |  |
| --- | --- | --- |
| **No** | **ボタン** | **処理の内容** |
| 1 | DBオープン | ラジオボタンで指定したパスワードにてDBをオープンします。DBをオープンする前に、No.7のボタンにより、クローズする必要があります。 |
| 2 | 一覧 | IntegrationTestApp1\_1と同じです。表 3‑11参照。 |
| 3 | 登録 | IntegrationTestApp1\_1と同じです。表 3‑11参照。 |
| 4 | 更新 | IntegrationTestApp1\_1と同じです。表 3‑11参照。 |
| 5 | 削除 | IntegrationTestApp1\_1と同じです。表 3‑11参照。 |
| 6 | 現在のパスワード | ラジオボタンで、利用するパスワードを指定できます。パスワードの変更の反映には、No.7のボタンでクローズし、No.1のボタンでオープンする必要があります。 |
| 7 | クローズ/パスワード変更 | DBをクローズします。パスワード変更するときなどに利用します。パスワード変更の手順は、No.6でラジオボタンの指定を変更した後、No.7のボタンをタップします。 |
| 8 | ファイル削除 | データベースファイルを削除します。暗号化したパスワードを別のパスワードで開こうとすると、画面上に「file is encrypted or is not a database」というエラーメッセージが表示されます。その場合、ファイル削除した後、No.1のボタンでオープンすれば、再度アプリの利用を継続できます。 |

* The test procedure

It should be performed as of the test specification. Since it is not difficult to omit the description. However, because it displays the current password on the screen, such as "current password [dbpass]" in FIG. 314, should be carried out while sure that you are able to operate with a password that is intended.‑

* Remarks

In CrypgetSQL, password "" and null is (empty string) will be treated the same. Therefore, test number 6.1.7 is not an error occurs.

It should be noted, and password when you create a database, if the password that you set when the load is different, "file is encrypted or is not a database" will be displayed on the screen.

### Test No. 7 .X

* The contents of the test

Make sure that the permissions of the database file that CrypgetSQL has created is as intended.

* Test app

Which is good, even for applications, but make sure the permissions of sqlcrypget.db file. (Other file name, for example, temp.db in IntegrationTestApp1\_1, so how to create a database file in order to perform the test number 4.X is different, permission is different.)

* The test procedure

Make sure that the permissions of the database file that CrypgetSQL has been created is a 660 (-rw-rw ----). Permission, in File Explorer that utilizes also in FIG. 33, you can check as shown in Figure 315.‑‑

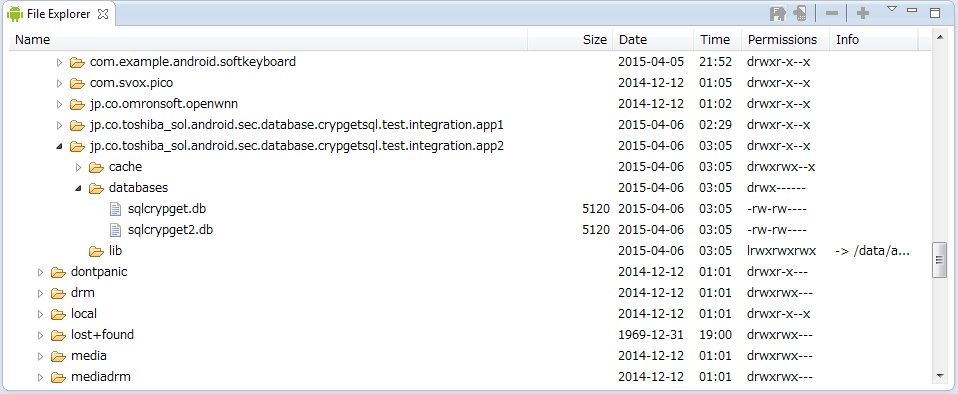


Figure 315 File permission in Explorer‑

### Test number 8.X

* The contents of the test

In CrypgetSQL V1.1, it has been divided into a plurality of so files in accordance with the Android environment. In the library of the Java layer, to get the Android version of the library is operating, there is a process of switching the so file that you want to use. Make sure that this process is running. In CrypgetSQL V2 / V3, this writing April 2015 now, because so file is integrated, This test will be excluded (not implemented).

* The test procedure

so file, (it is what was placed in Section 2.5.3), which located directly under libs / armeabi folder of the Eclipse project. The so file that environment that the test is available to grasp from the table 315, it should be deleted from the libs / armeabi.‑

Table 315 CrypgetSQL of so files and Android relationship (CrypgetSQL V1.1sp3)‑

|  |  |
| --- | --- |
| **Androidのバージョン** | **利用しているsoファイル** |
| Android4.0.3 (API Level 15) | libcrypgetsql\_android15.so  libdatabase\_crypgetsql15.so |
| Android4.1 (API Level 16)  Android4.2 (API Level 17) | libcrypgetsql\_android16.so  libdatabase\_crypgetsql16.so |
| Android4.4 (API Level 19) | libcrypgetsql\_android19.so  libdatabase\_crypgetsql19.so |

Link errors such as the following when you start in IntegrationTestApp1\_1 occurs.

Please check the Logcat (For Logcat, please reference Figure 33).‑

|  |
| --- |
| java.lang.UnsatisfiedLinkError: Couldn't load crypgetsql\_android19 from loader dalvik.system.PathClassLoader[DexPathList[[zip file "/data/app/jp.co.toshiba\_sol.android.sec.database.crypgetsql.test.integration.app1-2.apk"],nativeLibraryDirectories=[/data/app-lib/jp.co.toshiba\_sol.android.sec.database.crypgetsql.test.integration.app1-2, /system/lib]]]: findLibrary returned null |

### Test number 9.X

* The contents of the test

CrypgetSQL Make sure that you can start on platforms that support. Start the sample application, make sure that you can operate. The test of the platform, which has become a test subject is not required (that is, in the case of CrypgetSQL V1.1SP3, if is a Android4.4 tested platform, will Android4.0.3 the Android4.1 and Android4.2).

* The test procedure

The sample application to start, as of test specification, please operation. Because it is not difficult, it omits the description.

## The migration test procedure

* The contents of the test

Be changed from a previous version of CrypgetSQL to a new version, make sure that you can continue to use the database file that you created.

* Test app

There is no specified, but you can test in IntegrationTestApp1\_1.

* The test procedure

1. Start the Android devices to be tested.
2. In Eclipse, the old version of CrypgetSQL placed on libs, you clean the project.
3. In Eclipse, to create the APK file. How to create lists in Section 4.2.
4. At Eclipse, new version of CrypgetSQL be placed libs, Clean project.
5. In Eclipse, to create the APK file. How to create lists in Section 4.2.

So as not to overwrite the APK that you created in the above 竭｢, please save it in another folder. In addition, it should be signed with the same key as the APK that was created in the above 竭｢.

1. At the command prompt, to install the APK that was created in an earlier version of CrypgetSQL (created above 竭｢). (See Section 4.1.5. Specified at this time option -r is not required)
2. On the screen of the terminal, to register some data in the installed application (installed in the above 竭･).
3. At the command prompt, to install the APK that you created in the new version of CrypgetSQL (created above 竭､). (Please specify the -r to see Section 4.1.5. Option at this time)
4. On the screen of the terminal, make the list of the data in the installed application (installed in the above 竭ｧ). Make sure that the data that has been registered in the above 竭ｦ is displayed.

* Remarks

Migrating from V1 system to the V2 system can be you, but they can not migrate from V2 system to V3 system. From V3-based V3 systems (example: V3.0 from V3.1) assumes that you can do [footnoteRef: 2]. [2: for the test subject, I'd like a test about what you are supposed to be able to. If the test subject is V3.1, you will only to verify the migration from V3.0 to V3.1. If the test subject is a V2.0, the version that is available in V1 system to check on the raw skills dono, please check the transition from the version to V2.0. ][[2]](#footnote-2)

# appendix

## About ADB

Carrying out the test in the Android, called ADB (Android Debug Bridge), it is a useful tool for manipulating the Android terminal (actual emulator) from a Windows machine. In this section, we describe some of the ADB. Details on how to use, please check the following site.

<http://developer.android.com/tools/help/adb.html>

### Environment variable settings of ADB

If the adb from the command prompt command is not recognized, please through the path to adb.exe of the following SDK in the environment variable. It should be noted that, if you have not installed the SDK, install the 竭� in Section 2.1 to the reference.

[SDK installation folder] \ platform-tools \ adb.exe

### Confirmation of the terminal that ADB is aware

Devices looks even View (see FIG. 28), but you can see even adb.‑

|  |
| --- |
| > adb devices |

### Restart of ADB

Sometimes there is that the operation of the ADB is funny. For example, if the terminal, which has been recognized is no longer suddenly recognized, it may heal in the restart of the ADB. Restart of the ADB will be in the following command.

|  |
| --- |
| > adb kill-server  > adb start-server |

It should be noted that, if an error occurs in this start-server from the Windows Task Manager, to stop the process of adb.exe, again, when the adb start-server, you may move.

### Log acquisition of Android terminal

|  |
| --- |
| > adb logcat |

Please check the following site for details of options.

<http://developer.android.com/tools/help/logcat.html>

### APK install, re-install

APK installation of command in there is a APK folder directory is as follows.

|  |
| --- |
| > adb install hogehoge.apk |

Command of the re-installation of the APK in there is a APK folder directory is as follows.

|  |
| --- |
| > adb install –r hogehoge.apk |

### Uninstallation of applications

|  |
| --- |
| > adb uninstall [パッケージ名] |

It should be noted that, if not in the adb command, it is also possible to uninstall from Android devices (tap the "Uninstall" button shown in FIG. 310).‑

## Creation of APK

In this section, we describe how to create the APK file. The APK, stands for android application package, the necessary files to the Android application (main body of the program, such as resources, configuration information, such as the image that the program will use) is a file format that was turned into ZIP a. In such as the method shown in 4.1.5 section, it can be installed on Android devices.

1. Click on the "File" 竊� "Export ..." in the Eclipse menu.
2. In the next Select screen and select the "Export Android Application" in the "Android", and then click the "Next>" button (see Figure 41).‑

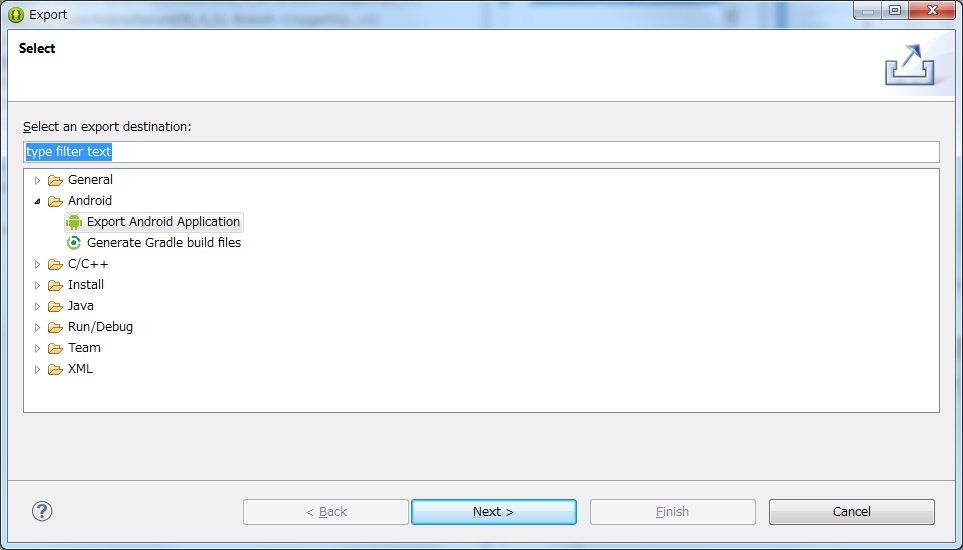


Figure 41 Export Select‑

1. In the next Project Checks screen and click the "Browse ..." button, select the project you want to create the APK file. Then click the "Next>" button.
2. In the next Keystore selection screen, you specify the Keystore that stores the key used to sign the APK.

If you want to use an existing one, check the "Use existing keystore", to specify the Keystore of the file in the "Location". Keystore is because it is protected by a password, enter the password to open the Keystore in the "Password" field. Then click the "Next>" button.

If you want to create a Keystore new, check the "Create new keystore", to specify the location of the Keystore file in the "Location" and enter the password for Keystore protection in the "Password" field and the "Confirm" column . Then click the "Next>" button.

In whether you select the "Use existing keystore" or "Create new keystore", it will change the procedure described below.

1. It is about the Key alias selection screen. If you check "Use existing keystore" in the above 竭｣, to use an existing key from Keystore "Use existing key" or, to create a new key "Create new key" or, to check.

In the case of "Use existing key", and Alias 窶銀��(values 窶銀�逆hat are used to specify the key (alias)), enter the password to use the key, and click the "Next>" button, go to later 竭ｦ.

In the case of "Create new key", then click the "Next>" button, go to later 竭･.

1. It is about Key Creation screen. If you check "Create new keystore" in the above 竭｣, or will this screen if you check the "Create new key" in the above 竭､. Click on the "Next>" button and enter the required information.
2. Finally, in Destination and key / certificate checks screen, specify the output folder of the APK, click the "Finish" button.

## Setting the character code

If the output of the application is garbled, check the setting of the character code. Character code of the source code is UTF-8 or MS932.

Right-click on the project of the Eclipse of the test application, and then click the "Properties" (Figure 211 reference). Focus on "Resource" in the left-hand side, check the "Other" in the right side of the "Text file encoding", and then select from the pull-down or enter the character code name. After the character code change, you must re-compiled with the "Clean" in the "Project" in the menu of Eclipse.‑

# Bibliography

[1] Staveware Mobile for AndroidTM / Developers Toolkit AndroidTM application development and testing guide

1. https://source.android.com/compatibility/cts-intro.html [↑](#footnote-ref-1)
2. 試験対象について、できると想定しているものについて試験をお願いします。

   試験対象がV3.1であれば、V3.0からV3.1への移行の確認のみになります。

   試験対象がV2.0であれば、V1系で利用されているバージョンを生技殿に確認し、そのバージョンからV2.0への移行の確認をお願いします。 [↑](#footnote-ref-2)