

IPHONE ANALYZER USER GUIDE

FORENSICALLY EXAMINING AN IPHONE OR IOS DEVICE



CONTENTS

Collecting Data.....	2
Learning About the Device	2
Opening the backup.....	3
Info.plist.....	4
Manifest.Plist.....	5
Navigating the Device.....	5
Finding Your Way Around The Application.....	6
Bookmarks	7
File System.....	7
Files.....	8
Exploring COnccepts.....	9
File Formats	9
PLists.....	9
SQLite.....	10
Special Views	11

COLLECTING DATA

The first challenge is to get a copy of the devices data which can be held in a way that assures it hasn't been tampered with. The method we recommend is to use iTunes or a 3rd party application to make a device backup.

If you use iTunes the default locations for the backup will be:

Windows (prior to Vista):

```
<user home>\Application Data\Apple Computer\MobileSync\Backup
```


Windows (Vista and Windows 7):

```
<user home>\AppData\Roaming\Apple Computer\MobileSync\Backup
```

MacOS:

```
/Library/Application Support/MobileSync/Backup/
```

Inside that directory you will find one or more backup directories. Each backup will have its own directory, which consists of 40 hex digits:









Name	Date modified	Type
 da963725d5c9bf05615b0369fb6000606d70969e	27/06/2011 21:33	File folder

At this point it is important to backup the selected folder and store it somewhere it cannot be tampered with. It may also be beneficial to take a hash of the directory so you can demonstrate it has not been changed.

LEARNING ABOUT THE DEVICE

If you look inside the backup directory you will see a long list of files (once again with filenames 40 hex digits long). Each of these represents a file from the IOS device, but packaged up in a non-readable format by iTunes.

The exceptions are a few files at the bottom of the list called Info.plist, Manifest (with various extensions) and Status.plist. These tell us something about the device, about the backup and about the files contained within it. These files are only partially human readable and also change between different IOS versions so we will use iPhone Analyzer to examine these.

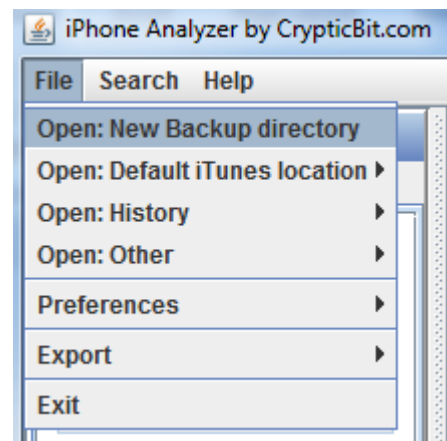
 107700ce01800243151d220c0e1e0041e24...	27/06/2011 21:33
 fb520955c98189505f20d2af90a46a1ced8c...	27/06/2011 21:33
 fdda2f81cc0b838dc00e3050b14da7ef2d83...	27/06/2011 21:33
 Info.plist	27/06/2011 21:33
 Manifest.mbdb	27/06/2011 21:33
 Manifest.mbdx	27/06/2011 21:33
 Manifest.plist	27/06/2011 21:33
 Status.plist	27/06/2011 21:33

OPENING THE BACKUP

Run iPhone Analyzer and select the File menu. If the file is where iTunes put it, you will find the file listed under "Open: Default iTunes location", however if you have moved it or are accessing it from a different device or user account then you will need to locate it manually using "Open: new backup directory".

Once you select this a file browser will appear. Simply navigate until you find the backup directory you want to open, and select "Open". Remember the backup directory will have be 40 hex digits long.

Once you have down this the backup will open, and more options will become available.



INFO.PLIST

On the right of the screen you will see the “Phone Information” appear. This is all extracted from Info.plist.

At the top you will see most important information extracted for you. This includes the date the backup was made, the phone number, the serial number and the device name.

Under this is the raw contents of the Info.Plist file. It is shown in a tree view which is described more fully in the PLists section on page 9.

Using this view it should be possible to see all the data from the Info.plist file and not just the content that the application decides is most relevant.

Other information that you may find useful includes:

- A list of library and synced applications
- IMEI number
- iTunes files such as IC-Info.sidb, IC-info.sidv and iTunesPrefs.



NOTE: You can open files such as IC_Info.sidb even though they are embedded within another file simply by right clicking on them and choosing to Open them as a new file

The screenshot shows the 'Examine Files' application interface. The top pane, titled 'Phone Information', displays the following data:

- Last Backup Date: 03-Jun-2011
- GUID: CCC38C52624D8A1CE1E61F07E884B8BD
- Unique Identifier: A0BB1C63FF5AA09CB485B2ABAE0DCB84F213DF05
- Phone Number: 07856 383781
- Product Version: 4.2.1
- Product Type: iPhone1,2
- Target Type: Device
- Serial Number: [REDACTED]
- iTunes Version: 10.2.2
- Target Identifier: a0bb1c63ff5aa09cb485b2abae0dcb84f213df05
- Display Name: Cryptic Bit iPhone
- Build Version: 8C148
- Device Name: Cryptic Bit iPhone

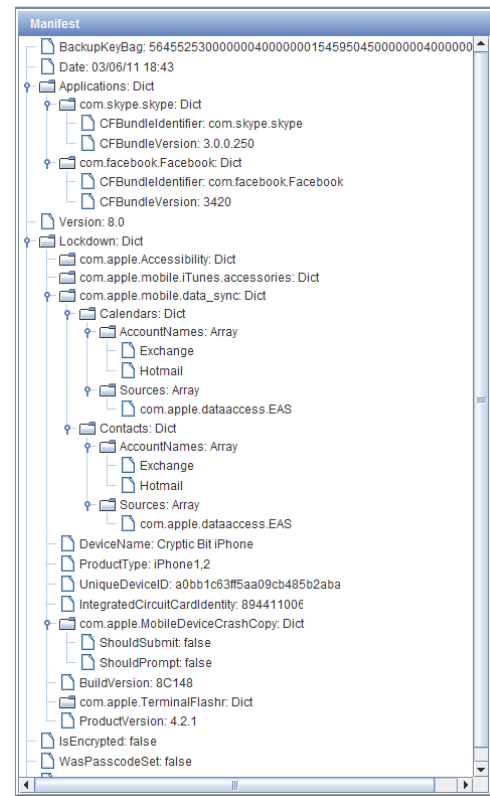
The bottom pane, titled 'Detail', shows a tree view of the file's contents:

- kAMSDDataClassEnabled: true
- Calendar Day Limit: 30
- GUID: CCC38C52624D8A1CE1E61F07E884B8BD
- iTunes Settings: Dict
 - SyncedApplications: Array
 - LibraryApplications: Array
 - com.facebook.Facebook
- Unique Identifier: A0BB1C63FF5AA09CB485B2ABAE0DCB84F213DF05
- Phone Number:
- iTunes Files: Dict
 - IC-Info.sidb: 060002862d4b985e683d747fd4bf1ceb420c394c
 - iTunesPrefs: 667270640100120001010001ec5321e05a2602
 - IC-Info.sidv: 0100019b64d2aa04771b406a6d93ce08a1ef7931
 - iTunesPrefs.plist: 3c3f786d6c2076657273696f6e3d22312e30
- ICCID: 8944110064992331763
- Product Version: 4.2.1
- Product Type: iPhone1,2
- Target Type: Device
- Serial Number:

MANIFEST.PLIST

On the far right hand side of the screen is the Manifest from the backup.

This is only one of the Manifest files, with others existing on some versions of the IOS operating system. However the others don't provide much human readable content as they simply let us associate the encoded files with their original filename, which is something iPhone Analyzer will do for you.



NAVIGATING THE DEVICE

Once you have opened a backup directory and examined any of the device information which is presented on the first page it is time to get your hands dirty. This section takes you through the key ways you can explore the contents of an IOS backup.

FINDING YOUR WAY AROUND THE APPLICATION

You can explore the device using bookmarks or the file system (see the two sections below)

A set of tabs across the top allow you to flip between (and close) open files



Using both bookmarks and the file system views allow you to open files. These files appear as new tabs on the right hand side. As well as opening and exploring files you can also use the menu to set preferences or export the entire file system to a directory of your choosing.

If you wish you can close unneeded files by clicking the small "x" on the right hand side of the tab.



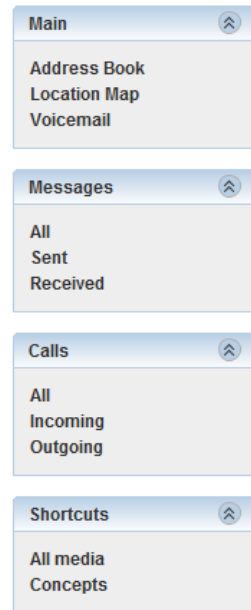
BOOKMARKS

Bookmarks are the easiest way to explore the file system. What we've done is choose the places you are most likely to want to explore on the backup and make them readily accessible.

These consist of:

1. The address book – listing all the contacts
2. The Location Map – the list of recently spotted GSM masts and wifi access points which provide a basic geo-location history
3. Voicemail messages – the metadata and sound clips from the most recent voicemails
4. Text Messages – selectable by incoming or outgoing
5. Calls – a list of phone numbers and whether they were incoming, outgoing and/or successful
6. All media – a quick view of all the images found on the device

Concepts – A combined data visualization view which we will explore in the section



7. Exploring Concepts on page 9.

Each of these views are explored later in this guide, but simply clicking on the bookmarks should be enough to get you going.

FILE SYSTEM

The file system view faithfully reconstructs the structure of the file system on the target device. To do this it uses a tree structure containing:

- 📁 Directories – which you can expand by double clicking, or selecting the expansion icon next to it.
- 📄 Files – which you can open by single clicking. They will appear on the right hand side in a new tab.

In order to understand the files you see it may be useful to read the section File Formats on page 9.

The directory structure is always the same with the top level headings:

1. Documents
2. Library
3. Media
4. System Configuration

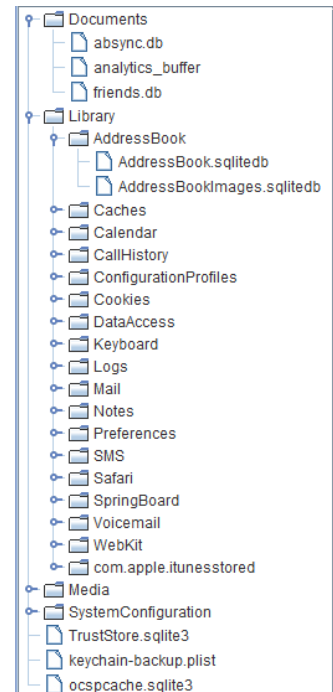
Some of the most interesting files are listed under IOS devices also rely heavily on sqlite files, which Wikipedia describes as:

SQLite is an ACID-compliant embedded relational database management system contained in a relatively small (~275 kB)[4] C programming library. The source code for SQLite is in the public domain[1][5] and implements most of the SQL standard. In contrast to other database management systems, SQLite is not a separate process that is accessed from the client application, but an integral part of it.

We currently use an embedded SQLJet browser (to provide inline) browsing capability for these.

Initially when browsing to a sqlite file you will be presented with the table structure. You can use the internal tab to “Browse Data” and then select any table to view the contents of the database.

Next to the “Sql” tab at the bottom of the page is another tab called “deleted fragments”. This tab does its best to recover deleted items from the database, and display them as a list. The complex algorithm for this is not always able to distinguish between deleted records, and bits of unneeded database content, so you are likely to see it punctuated with “gibberish” entries. We leave this in place as humans are much better at figuring out what’s important than computers.



Special Views on page 10, whereas others will simply require exploration.

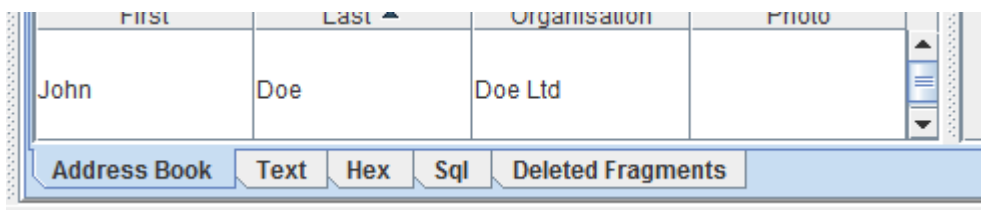
FILES

Once you select a file (from either the bookmarks or selecting it from the file system) it will open in a new tab on the right of the screen. Any existing files will remain open and can be found again by navigating the tabbed browser at the top of the screen.

When a file opens iPhone Analyser it will be visible in a variety of formats. The common ones include:

- A “special view” (see page 10) which has been written specially for the file being opened (this is only done for the most interesting files)
- A text view – where the valuable information is extracted and made available in a text only view, suitable for quick viewing and copy and pasting.
- An image view – for any sort of photo or image, along with any meta-data associated with it.
- PList view - a tree viewer capable of showing any XML or binary plist
- Sqlite view – shows both table structure and content
- Sqlite undelete – shows text extracts that appear to have been deleted from an sqlite database
- Hex view – shows any file format in raw hex format

All views that apply to the selected file will be visible, and you will be able to navigate them using the tabs at the bottom of the main right hand pane.

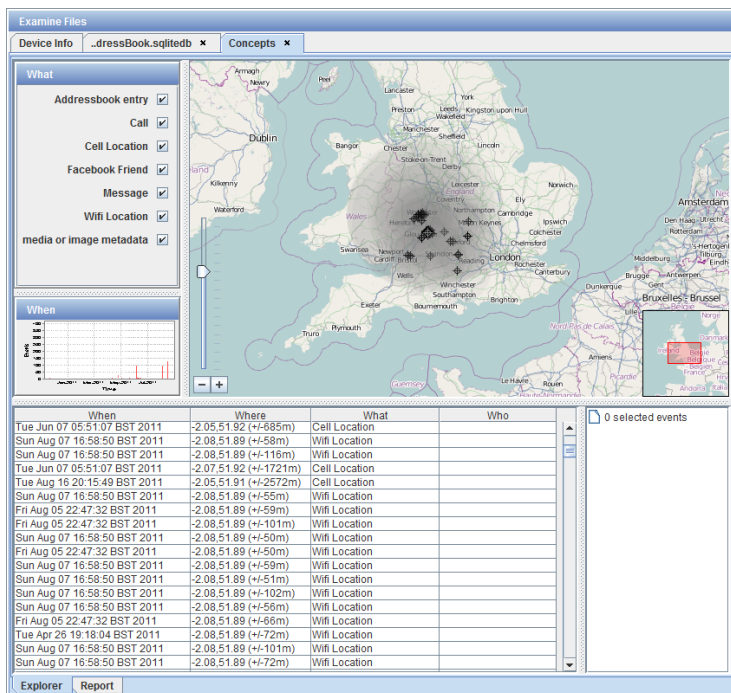


EXPLORING CONCEPTS

As you will have realized there is an awful lot of information on an iPhone. To aid analysis iPhone Analyzer contains a powerful analytic tool which allows you to explore the data from multiple files concurrently in a single view. We call this ConceptExplorer.

ConceptExplorer scans files on the device looking for:

- Any times or dates (when)
- Any names or personal identifiers, e.g. email addresses (who)
- Any location information (where)
- The activity associated with any of the above (what)



By pulling together these concepts we can create a powerful interactive display, allow you to filter on location, time or activity – and explore the details of the activity that caused it.

You can quickly narrow in on interesting information by filtering in any part of the view. Changes in one part of the screen will affect all other parts, so as you zoom in on the map records will disappear from all other views. You can filter by either:

- Zooming or panning the map (only records on screen, or that contain no geo information will be shown elsewhere)
- Unselecting activity types from the “what” panel.
- Selecting or zooming the “when” panel. You can zoom in by highlighting part of the screen or using the scroll wheel on your mouse. You can zoom out again using the scroll wheel.

FILE FORMATS

PLISTS

A common format for preference and system files on iOS devices is the PList (or property list) file format. From Wikipedia:

In the Mac OS X, iOS, NeXTSTEP, and GNUstep programming frameworks, property list files are files that store serialized objects. Property list files use the filename extension .plist, and thus are often referred to as plist files.

Property list files are often used to store a user's settings. They are also used to store information about bundles and

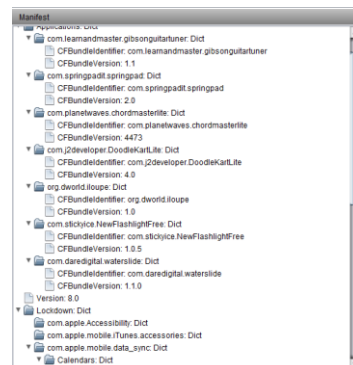


Figure 1: The Manifest is a PList

applications, a task served by the resource fork in the old Mac OS.

For iOS devices these come in two formats:

- Binary
- XML

Regardless of type iPhone Analyzer will render these in the same way using a tree structure. Nodes can be expanded and shrunk.

When viewing a preferences file you will also have the opportunity to view the plist file as text. This will display the original XML for XML plists, and will convert binary plists to XML.

Occasionally plists include binary data, which can be the contents of a new file (sometimes another plist). iPhone Analyzer can unwrap these embedded files if you write click on them, allowing you to

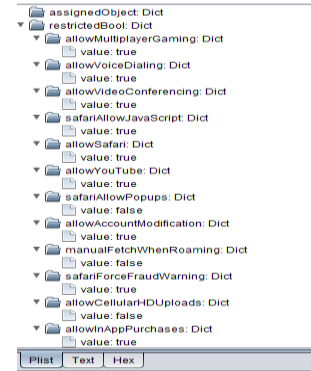


Figure 2: A plist preferences file

SQLITE

iOS devices also rely heavily on sqlite files, which Wikipedia describes as:

SQLite is an ACID-compliant embedded relational database management system contained in a relatively small (~275 kB)[4] C programming library. The source code for SQLite is in the public domain[1][5] and implements most of the SQL standard. In contrast to other database management systems, SQLite is not a separate process that is accessed from the client application, but an integral part of it.

We currently use an embedded SQLJet browser (to provide inline) browsing capability for these.

Initially when browsing to a sqlite file you will be presented with the table structure. You can use the internal tab to “Browse Data” and then select any table to view the contents of the database.

Table Name	Type	Schema
Cell	table	CREATE TABLE Cell (Timestamp FLOAT, MCC INTEGER, MNC INTEGER, LAC INTEGER, CI IN...
Cell	table	CREATE TABLE Cell_location (MCC INTEGER, MNC INTEGER, LAC INTEGER, CI INTEGER, Tim...
Cell	table	CREATE TABLE Cell_location_Base_node (nodeno INTEGER PRIMARY KEY, data BLOB)
Cell	table	CREATE TABLE Cell_location_Base_parent (nodeno INTEGER PRIMARY KEY, parentnode INTE...
Cell	table	CREATE TABLE Cell_location_Base_rowid (rowid INTEGER PRIMARY KEY, nodeno INTEGER)
Cell	table	CREATE TABLE Cell_locationCounts (Count INTEGER)
Cell	table	CREATE TABLE Cell_locationHarvest (MCC INTEGER, MNC INTEGER, LAC INTEGER, CI INTE...
Cell	table	CREATE TABLE Cell_locationHarvestCounts (Count INTEGER)
Cell	table	CREATE TABLE Cell_locationLocal (MCC INTEGER, MNC INTEGER, LAC INTEGER, CI INTE...
Cell	table	CREATE TABLE Cell_locationLocal_Base_node (nodeno INTEGER PRIMARY KEY, data BLOB)
Cell	table	CREATE TABLE Cell_locationLocal_Base_parent (nodeno INTEGER PRIMARY KEY, parentnode...

Next to the “Sql” tab at the bottom of the page is another tab called “deleted fragments”. This tab does its best to recover deleted items from the database, and display them as a list. The complex algorithm for this is not always able to distinguish between deleted records, and bits of unneeded database content, so you are likely to see it punctuated with “gibberish” entries. We leave this in place as humans are much better at figuring out what’s important than computers.

Table: CellLocation	MCC	MNC	LAC	CI	TimeSta...	Latitude	Longitu...	Horizont...	Altitude	Vertical...	Speed	Course	Confide...
1	Oxea	10	42515	0	3.13345...	51.5124...	-0.1332...	1174	0	-1	-1	-1	70
2	Oxea	10	12424	7949	3.13345...	51.5123...	-0.1330...	500	0	-1	-1	-1	70
3	Oxea	10	42115	4024287	3.13345...	51.5125...	-0.1337...	500	0	-1	-1	-1	70
4	Oxea	10	42115	4019287	3.13345...	51.5125...	-0.1338...	500	0	-1	-1	-1	70
5	Oxea	10	34583	4014942	3.13345...	51.5127...	-0.1326...	855	0	-1	-1	-1	68
6	Oxea	10	42515	3401088	3.13345...	51.5124...	-0.1340...	500	0	-1	-1	-1	70
7	Oxea	10	42515	3348088	3.13345...	51.5119...	-0.1333...	500	0	-1	-1	-1	70

Powered by SQLJet v1.0.4.bLocal
© 2009-2010, TMat Software, <http://sqljet.com/>

Location Map | Sql | Deleted Fragments | Text | Hex

SPECIAL VIEWS

In addition to displaying plists, sqlite, images, and other file formats we try to render the most useful files in a more intuitive way. This list is every-growing but to-date we have special support for:

- Address book (including images)
- SMS
- Call records
- Location mapping (recent wifi and cell sites)
- Voicemail (including audio clips)