

Table of Contents

Welcome to DROID	2
Whats New	3
License	6
Third-party components	8
Project team	10
FAQ	11
Profiling your files	17
Create a new profile	17
Choose files and folders	18
Running a profile	21
Open and save profiles	24
Command line control	26
Exploring your results	32
Information collected by DROID	32
Explore the results	39
Summary reporting	44
Exporting profiles	47
Detecting duplicate files	51
Configuring DROID	53
Change preferences	53
Update file format signatures	59
Startup configuration	61
Index	65

**DROID**
Digital Record Object Identification**Welcome to
DROID**[What's new](#)[Information
collected](#)[License](#)[Team](#)[Frequently Asked
Questions](#)

DROID is a tool to help you analyse what kinds of files you have. It collects [information](#) about your files into profiles, and lets you filter, export and report on your results. It recognises file formats by using 'signature' information provided by digital preservation researchers at The National Archives.

To begin using DROID:[Create a new profile](#)[Choose files and folders](#)[Run your profile](#)**Once you have some results:**[Filter your results](#)[Create a summary report](#)[Export your results](#)

You can also [open or save](#) existing profiles, update your file format [signatures](#), or change your profiling [preferences](#). DROID can be controlled by a graphical user interface or using a [command-line](#). For frequently asked questions and troubleshooting, please read the [Frequently Asked Questions](#).

If you would like to provide suggestions, bug reports or feedback on DROID, please e-mail droid-users@lists.sourceforge.net, or you can browse the mailing list archives[↗]. The latest version is available from the DROID web-site[↗].

It runs on Windows, Mac and Linux, and potentially on other platforms that support applications written in Java 6[↗], from Java 6 update 10 onwards. DROID has been designed and developed by the [DROID team](#) in the Digital Continuity Project[↗]. It has been tested on Java 6 update 17 to update 22.

[What's new](#)[Information
collected](#)[License](#)[Team](#)[Frequently Asked
Questions](#)

**DROID**
Digital Record Object Identification

What's new

[Welcome to DROID](#)[License](#)[Team](#)[Frequently Asked
Questions](#)

DROID 6 has many new features which improve the speed, accuracy and your exploration of its results. Behind the scenes, it introduces some new capabilities to allow signatures for new file formats to be developed more easily, and with greater power to match difficult file formats.

- [Speed](#)
- [Accuracy](#)
- [Reporting](#)
- [Exporting](#)
- [Filtering](#)
- [User interface](#)
- [Container signatures](#)
- [Binary signature syntax](#)

Speed

- A new binary signature recognition engine is approximately twice as fast as the old engine, and with an [enhanced syntax](#) offering new kinds of signatures.
- An option to [limit how much of a file is scanned](#) from its beginning and end.
- New profiles are quicker to start up, using copies of pre-built empty databases, rather than creating them anew each time.
- Profiling tools[☞] were used to locate and eliminate bottlenecks. In particular, binary signature recognition and archival file handling is now much faster.

Accuracy

- An option to choose [how file extensions are matched](#).
- A [new type of signature](#) for file formats which are not recognised well using the existing engine.
- Enhancements to the [binary signature syntax](#) allows for more powerful binary signatures to be developed in future.
- DROID can now process 64-bit zip files.

Reporting

- A new [reporting](#) engine offers 11 predefined [reports](#), giving statistical breakdowns of your files and folders in various different ways.
- Reports can be viewed on screen, and saved into several different file formats, including PDF, text, web pages and XML.

Exporting

- CSV files can now [export](#) results one row per file, or one row per format identification.

Filtering

- Filters can now be saved and loaded for quick access to commonly used filters.

User interface

- Icons are used to communicate essential information quickly.
- [Filtering](#) can be turned on or off from the main toolbar, via the Filter On checkbox.
- Folders and archival files which are filtered out in the main view appear greyed out, to distinguish them from items which were not filtered out.
- If a file has multiple identifications, its format information is now directly accessible from the main window.
- "Open Containing Folder" option to go straight to the files you have profiled from within DROID.
- A right-click popup menu offers common editing options quickly.
- Copy to clipboard for results in the main window.
- "Expand" and "Expand next three levels" options for selected results in the main window.
- General tidying up to maximise the use of screen space.

Container signatures

- Container signatures are a new kind of file format signature, to recognise file formats which contain other files inside themselves (for example, Microsoft Office 2007 files).

Binary signature syntax

The syntax of the signatures used by the binary signature engine has been enhanced, both in terms of readability and fundamental expressivity. In addition to existing syntax, it allows for text strings, case insensitive text strings and arbitrary sets of bytes. These features can be mixed together in ways which the previous engine cannot process. This makes it easier to create new signatures, and to

understand existing ones. In some cases, several old signatures can be combined into one signature, which increases the speed of DROID.

The new syntax will be documented in other technical documentation, and will not appear in signature files for some time until users of DROID have updated to the newer engine. Some of the new syntax can be seen in the container signature files, which do not have backward compatibility issues.

[Welcome to DROID](#)[License](#)[Team](#)[Frequently Asked
Questions](#)



DROID
Digital Record Object Identification

License

[Welcome to DROID](#)

[Third-party components](#)

DROID is made freely available by The National Archives to anyone under the open-source BSD License, which permits its use, modification, inclusion in other products and redistribution as long as the terms of the license are adhered to.

DROID license

Copyright (c) 2009-2010, The National Archives
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of The National Archives nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Third-party component licenses

DROID uses several [third-party components](#), which are made available under a few widely used open-source licenses. All licenses to code used in DROID permit the combined work as released by The National Archives to be freely used and re-distributed.

[Welcome to DROID](#)

[Third-party components](#)



DROID
Digital Record Object Identification











Third party components



[Welcome to DROID](#)

[License](#)

DROID uses a variety of open-source third-party components, providing indispensable functionality, such as this help system, a database, XML and CSV file processing, graphical components, web services, logging, file handling, archival file handling, command line control and many other utilities to enhance the java platform. The DROID project team would like to thank the authors of these components for making software like DROID possible.

Component	License
Apache Commons Logging	Apache License
Apache Commons Command Line Interface	Apache License
Apache Commons Input Output	Apache License
Apache Commons Language Extensions	Apache License
Apache Commons Configuration	Apache License
Apache Commons Compression	Apache License
Apache Commons Collections	Apache License
Apache Commons Database Pooling	Apache License
Apache Commons Configuration	Apache License
Apache Crossfire Services	Apache License
Netbeans Swing OpenIDE Platform	Common Development and Distribution License
Netbeans Swing Outline Component	Common Development and Distribution License
Beansbinding	Lesser General Public License
Spring Framework	Apache License
JodaTime	Apache License
OpenCSV	Apache License
Apache Log4J Logging Framework	Apache License
Xerces XML Parser	Apache License
Apache Derby Database	Apache License
Java Transaction API	Sun JTA License
Simple Logging Facade for Java	MIT License
Hibernate Entity Manager	Lesser General Public Licence
Code Generation Library	Apache License

Java Help 	General Public License with Classpath Exception 
ANTLR Parser Generator 	BSD License 
Flying Saucer XHTML & PDF Renderer 	Lesser General Public License 
TrueZip 	Apache License 
ByteSeek 	BSD License 

Only the compiled form of these components are included in DROID. No modifications to the source code of these components have been made by the DROID project. The source code to these components is available from their respective owners, via the links in the above table. The icons used in DROID are freely provided by Aha-Soft  and the Eclipse  project.

[Welcome to DROID](#)

[License](#)



DROID
Digital Record Object Identification

Project team

[Welcome to DROID](#)

Digital Continuity

Jessie Owen	Head of the Digital Continuity project and initiator of the DROID 6 project
Matt Palmer	Product owner for DROID, technical lead in the Digital Continuity project, development of new engine.
Alex Patterson	Scrum master for the DROID project and test manager
Michael Appleby	Original scrum master for the DROID project.
Richard Brennan	File format researcher
Richard Flitcroft	Senior java developer
Alok Dash	Senior java developer
Graham Haslehurst	Procurement lead for DROID and platform testing
Lorna Robinson	Project management and functional testing
Daniel Bowmar	Functional testing
Alec Mulinder	Functional testing
Ivan Molesworth	Scalability testing

Digital Preservation

Tim Gollins	Head of Digital Preservation
Ross Spencer	Digital Preservation analyst
Andrew Fetherstone	Digital Archivist
Tracey Powell	Digital Preservation business development manager
Lynne Montague	Digital Preservation PLANETS liaison

Special thanks...

Gemma Richardson	Human-computer interaction advisor
Matt Hilliard	XML schema design advisor

[Welcome to DROID](#)



DROID
Digital Record Object Identification

Frequently Asked Questions

[Welcome to DROID](#)

- [What version of Java do I need?](#)
- [Can I open profiles created in earlier versions of DROID?](#)
- [Why isn't DROID recognising certain formats?](#)
- [Why are some file times an hour different to those on my system?](#)
- [Can I configure DROID to give more accurate results?](#)
- [Can I continue to use information in the file system while I am profiling it?](#)
- [How long does profiling take?](#)
- [Can I set it up and just leave DROID running?](#)
- [Will DROID 6 impact on the performance of my systems?](#)
- [How much disk space do I need to run DROID?](#)
- [Are there any limitations to where DROID can profile?](#)
- [Does DROID 6 store copies of the files that it is profiling?](#)
- [Why has DROID stopped profiling before completion?](#)
- [Why can't I resume a paused profile?](#)
- [Why does filtering on a "Not Done" status give no results?](#)
- [Why does opening some filtered folders take a long time?](#)
- [What can I do if DROID fails to start up?](#)

What version of Java do I need?

You will need to have Java 6 installed on your machine in order to run DROID 6.

Please note that DROID will not run on early versions of Java 6 and requires versions that fully support JAXB 2.1. Testing was carried out on Java 6 update 17 and beyond. The latest JAVA updates can be obtained from oracle.com/technetwork/java/index.html.

Can I open profiles created in earlier versions of DROID?

No. Because of the expanded functionality incorporated in to DROID 6, you will not be able to load profiles from previous versions in DROID 6 – and vice-versa. If you need to re-analyse results in DROID 6 which were already obtained in earlier versions of DROID, you will have to re-profile your files using DROID 6.

Why isn't DROID recognising certain formats?

DROID can only recognise files for which there is a defined signature or known extension.

Signatures are regularly updated by the National Archives, and DROID can download them to improve the range of formats it can identify. In some cases, an existing signature may fail to recognise a file because the file is unusual in some way. If you know of such files, please email droid-users@lists.sourceforge.net so we can develop better signatures. Note that we usually require a sample of a failing file to find out why it is not being recognised. Please make sure that there is no sensitive or personal content in such files before sending them.

Why are some file times an hour different to those on my system?

DROID does not record or display dates and times using local daylight savings time, if any is configured on your system. The underlying dates and times are correct, but represented as universal time. This means that, if your operating system is configured to display times using daylight savings time settings, then some times can appear to be one hour different to those displayed in DROID.

Can I configure DROID to give more accurate results?

Yes. DROID 6 currently makes identifications of files based on scanning a limited amount of the file. The default configuration is set to 64kb (you can change this in the Tools>Preferences menu). This is sufficient to return accurate identifications for most file-types. There are some exceptions, for example in order to identify some uncommon formats (e.g. SIARD files), you will need to scan more or all of the file. If you choose to increase the amount of the file that you scan, this is likely to significantly affect the performance of DROID, particularly if you have numerous large files.

You can also configure whether DROID should match file extensions on all known formats for a given file extension, or only on file extensions for which there is no other signature. By default, DROID will only match a file extension if there is no other signature defined, on the grounds that if the other (usually more accurate) signature failed, it is unlikely to be the file format with that extension. Sometimes the more accurate signatures will fail to identify a format (e.g. if a newer version has recently become available). In these cases, you may want to turn on matching for all extensions. This will tend to produce more identifications for a larger number of files. More files will have identifications where they did not before, and more files will have multiple identifications, instead of a single identification. Whether this is more accurate is debatable and will depend on the files and signatures you currently have.

Can I continue to use the information in the file system whilst I am profiling it?


Yes. If you are running DROID over large file stores, it is conceivable that you may need to leave it running for a number of days or weeks. DROID will profile what is there at the time, and so over large active locations the results will be gradually built up even while change is happening. The

eventual profile will therefore not reflect the entire state of your system at any one time, but rather the state of each file or folder at the time DROID profiled it.

You don't need to worry about users then changing files in the location that DROID is profiling; if a file is identified in the file tree but then deleted or moved, it will be recorded as an error.

How long does profiling take?

DROID 6 is much faster than earlier versions of DROID. However, it depends on how much and what type of information you are profiling, what level of accuracy you choose and what information you need to get from the results. As a general rule, the larger the volume of data you profile the longer it will take for the profile to complete. However DROID will also take longer to complete if you chose to:

- Analyse the contents of archive files (e.g. zip, gzip, tar)
- Create an MD5 hash  for each file
- Increase the number of bytes that DROID scans in order to return an identification
- Set a delay between identifying files using the throttle (to reduce the load on your network or servers).

You will need to weigh up your requirements for accuracy, speed and what use you want to make of the profiling information, setting your preferences accordingly before you begin to profile your information.

Can I set it up and just leave DROID running?

Yes. Although you need to ensure that DROID has the correct access permissions to profile the locations that you have identified. Also, it is good practice to pause and save your profiling at regular intervals, to ensure that if you experience an unexpected problem (such as power shortage or the area that DROID is running in is affected) you can pick up your profiling again without losing all your results so far.

It is also possible to set up batch files or shell scripts to automate the process and produce reports when running DROID through the [command line](#). You should speak to technical specialist in your organisation for more information about this.

Will DROID 6 impact on the performance of my systems?

Yes, there will be some impact, but this depends on a number of factors. DROID does not typically exert a high load on the file systems and networks it is running over, but this will obviously depend on your particular environment and usages.

DROID 6 has dynamic throttling (configurable only in the GUI), so if you want to limit the impact on your systems you can increase the amount of time that DROID pauses between identifying each file. Because it is dynamic, you can for example slow down the rate at which DROID works during the day, to reduce the load on your network, then increase its speed in the evening. Remember that the more you ask DROID to pause, the longer it will take to complete profiling.

How much disk space do I need to run DROID?

DROID needs disk space in its working area to process [temporary files](#), and to record the database of results it builds up as it profiles. Profiles can require up to a gigabyte per two million files processed.

Are there any limitations to where DROID can profile?

You will need to ensure that DROID 6 has the appropriate access permissions to scan the repositories that it is you have requested. DROID can only profile files held in file systems or systems which provide a file-system view of its files to your operating system (e.g. a network share).

Does DROID 6 store copies of the files that it is profiling?

DROID 6 creates temporary files when profiling inside archival files, such as zip, tar or gzip files.

These are written out by default as ~tmp files in the **.droid6** tmp directory, unless the [temporary folder area](#) has been configured to be somewhere else. There are automatic cleanup processes to remove information from this directory when the temporary file is no longer needed, however on some occasions (such as if DROID stops unexpectedly), some files may not be deleted. It is good practice to check this location regularly and remove any ~tmp files that are no longer required by DROID. DROID will log a warning if it cannot delete a temporary file.

It is not possible to say how much space is required for temporary files while processing, as this will depend on the size of your archival files and their contents. DROID also profiles more than one file at a time, so more than one archival file may be being processed simultaneously. It is prudent to ensure that you have enough space in your droid working area to handle decompressing several archival files at any one point in time. If processing very large file systems, we would recommend that at least 10 gigabytes of disk space should be reserved for temporary file processing, but you may require more or less depending on your files.

Why has DROID stopped profiling before completion?

This may be for a number of reasons. DROID 6 has crash recovery built in, which means as long as DROID has sufficient warning that a file system has become unavailable (which it may not have in

some instances, such as a power shortage) it will pause the current profile ready to be saved and / or resumed at a later date.

It is good practice to pause and save your profiling at regular intervals, to ensure that if you experience an unexpected problem (such as power shortage or the area that DROID is running in is affected) you can pick up your profiling again without losing data.

Why can't I resume a paused profile?

Because of the crash-recovery capability built in to DROID, it is important that any of the top level folders that are selected are not deleted, removed or renamed either during profiling (if the profile is paused) or before DROID has begun to profile them. This will cause the profile to pause and you will not be able to resume it. This is because DROID interprets a missing top-level folder as a missing file system, and it automatically puts the profile into a pause mode to enable you to save your work and to resume when the file system becomes available again.

It is best practice to select the highest folder possible when selecting the top-level folders that you intend to profile and ensure that these are not moved, deleted or renamed during or before profiling.

If you do experience your profile stopping for this reason, you can still interrogate the data that you do have. You can also create a new profile that contains the remaining top-level items which were not profiled. Multiple profiles can be reported on and exported together, so you will achieve the same results.

Why does filtering on a "Not Done" status give no results?

Files or folders which have not yet been processed have a Not Done status in DROID. These files and folders have not yet been recorded in the database, only in the specification of what to eventually profile. Filtering works by querying the underlying database in which DROID records its results, therefore filtering on the Not Done status returns no results, even if there are files and folders visible in the main view which have not yet been processed.

Why does opening some filtered folders take a long time?

When a filter is applied to a profile, the user interface has to do a lot of work determining the status of each file or folder in the opening folder. Folders with lots of direct files or other folders in them (i.e. many thousands) can take quite a long time to open. This does not affect the filtering speed for exporting or reporting, so if you are having problems opening a large folder with a filter, consider exporting the profile to a CSV file, or running a report over it, rather than trying to open it in the user interface.

What can I do if DROID fails to start up?

- If DROID fails to start up, you may be running the wrong [version of Java](#).
- You can try moving, deleting or renaming your **.droid6** folder, as described below. DROID will recreate a new working folder with all the default files it needs when it starts up again.
- Ensure that your installed version of Java is accessible from where DROID is installed. From a command-line in your droid installation folder, type "java -version". If you don't see the Java version being displayed, you may have to configure the path to your Java installation. Please consult the installation instructions for Java for your operating system for details on how to do this.

Deleting your **.droid6** folder


By default, your **.droid6** folder is set to be created in your home directory. On Windows, this is often found at `C:\Documents and Settings\USERNAME\.droid6`, or `C:\Users\USERNAME\.droid6`. On unix systems, this may be found under `/home/USERNAME/.droid6`


If you are running DROID on a Mac or other Linux based system, please note that the **.droid6** folder will be in the same location but will be 'hidden'. Note that Mac systems do not make hidden files easily accessible from your computer. You can delete the folder from your machine using the command line. Please refer to your operating system instructions for doing this.

(Please note: the location of the **.droid6** folder is [configurable using a batch file or shell script](#), and may have been set to a different location on your machine)


By default, the **.droid6** folder contains any downloaded binary and container signature files, log files, settings, report definitions, and temporary working area, although some of these can be [configured to be in different locations](#). Please make sure you no longer require any of this information before deleting. However, DROID will recreate all the default information it needs by default when it runs again, effectively resetting to a new installation.


[Welcome to DROID](#)

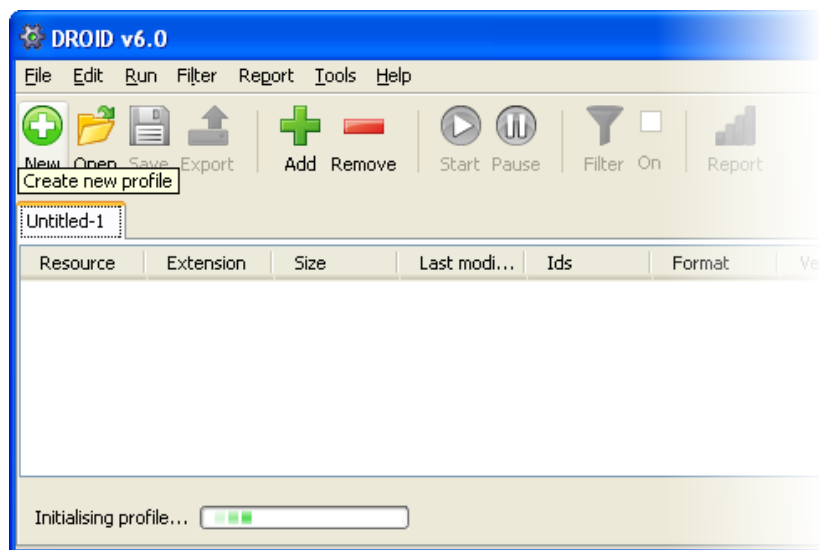
**DROID**
Digital Record Object Identification


 **Create a new profile**

[Welcome to DROID](#)

 [Choose files and folders](#)


A profile is the files and folders you want to find out about, and the results of profiling them. You can create profiles by clicking the  **New** button, or selecting the **File / New** menu item. DROID automatically creates a blank profile for you when it starts.




You can create as many profiles as you like. Each profile appears in its own tab, underneath the toolbar. You can [choose files and folders](#) as soon as the tab appears using the  **Add** command, but it takes a few seconds for a new profile to be ready to [run](#). By default, profiles are "Untitled" until they are [saved](#) with a filename.


Once a profile is created, its settings are fixed from the [Profile Defaults](#) specified in the preferences window.

[Welcome to DROID](#)

 [Choose files and folders](#)



DROID
Digital Record Object Identification



**Choose
files and
folders**

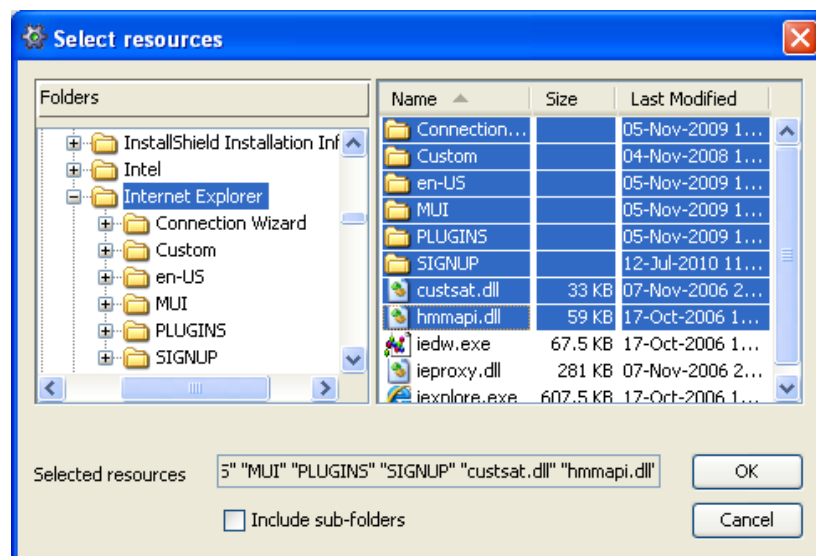
[Welcome to DROID](#)
[+ Create a profile](#)
[+ Adding](#)
[- Removing](#)
[▶ Run your profile](#)

+ Adding

To add files and folders to a profile, click the + **Add** button, or select the *Edit / Add files and folders* menu. You can also right-click with your mouse on the main profile window, and select the Add option from the popup menu.

A selection window will appear. On the left hand side is a navigator to explore your folders. The example below shows the "Internet Explorer" folder selected on the left hand side, and the contents of this folder are shown on the right.

Any files or folders selected on the right will appear in the "*Selected resources*" text box at the bottom. Select multiple files and folders at the same time by holding down the SHIFT key to select a list, or the CTRL key to individually select or deselect files or folders.




Include sub-folders

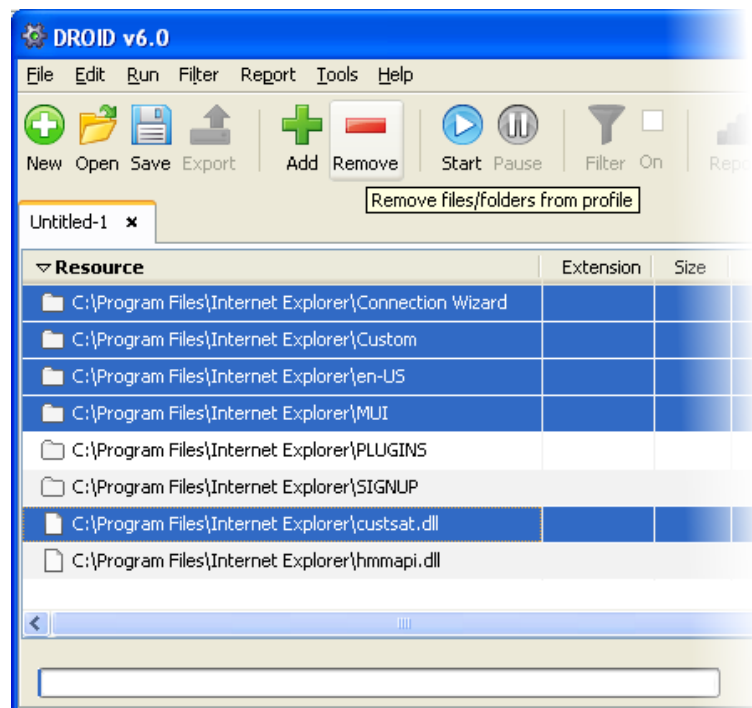
When a folder is added to a profile, it is often useful to also profile all of its sub-folders. Underneath the selected resources text box, ☒ Include sub-folders is checked by default. Uncheck this box if you only want to profile the files directly inside a folder and to ignore any sub-folders. If you are only selecting files, this setting makes no difference. Note that if you are [profiling inside archival files](#), they will still be profiled regardless of this setting, which only applies to folders in the file system you are profiling.

Adding your selection

Press the **OK** button to add the selected files and folders to your profile, or the **Cancel** button to leave the selection window with no changes made. **+ Add** can be used repeatedly to add files and folders from different locations in your file system to your profile.

Removing

To remove files or folders from the profile, select them in the main window, and either press the  **Remove** button, or use the *Edit / Remove files or folders* menu item. You can also right-click with your mouse on the main profile window, and select the Remove option from the popup menu.



Only the top-level files and folders you have added to a profile can be removed. When you [run](#) a profile with a folder, the profiling process will automatically add the files and folders it finds

underneath it. After you run a profile, you cannot add or remove files or folders from the profile anymore - its specification becomes fixed.

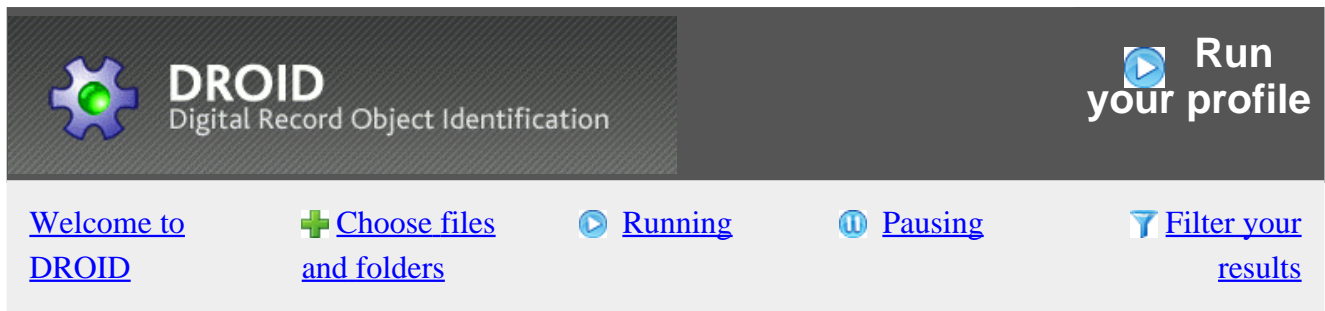
[Welcome to
DROID](#)

[!\[\]\(dfbd6b3763a6d1d9afaa974f64e2e4b5_img.jpg\) Create a
profile](#)

[!\[\]\(e78f798d4ea5c530c9db49e7d26e6b95_img.jpg\) Adding](#)

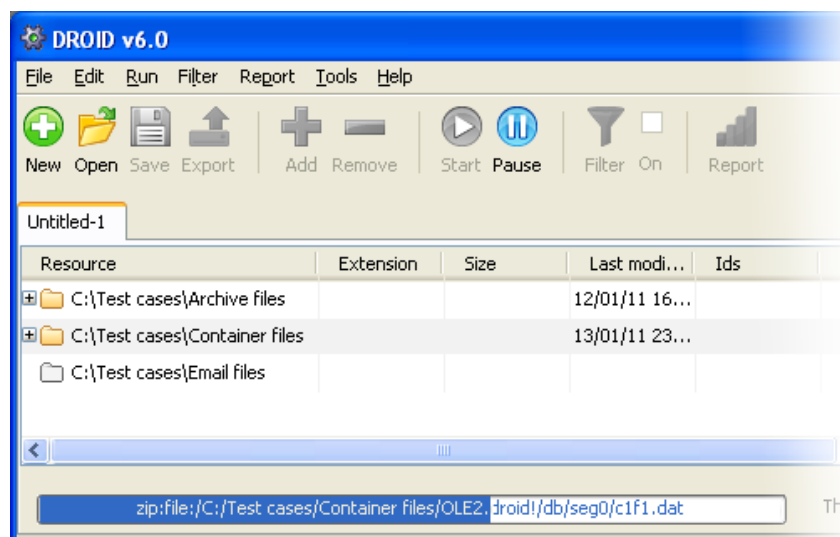
[!\[\]\(23d9fc146e83b5c3013cfa32c784f8d5_img.jpg\) Removing](#)

[!\[\]\(c694a3ff3b077d76910920a6a1593ab4_img.jpg\) Run
your profile](#)



▶ Running

You can start profiling files or folders using the ▶ **Start** button, or by selecting the ***Run / Start Identification*** menu item.



As files and folders are identified, they are added to the profile, and you can see all the results obtained so far. If you expand a folder, it will show all the files and folders found so far. If you happen to open the folder that is being profiled at that time, you will see its child files and sub-folders appearing under it.

Once you have started a profile running, you cannot [choose](#) any further files or folders in it. The specification of which files and folders to process in your profile becomes fixed at the point it first begins running. If you want to subsequently profile other files or folders, then you can do this in a [new profile](#). Results held in multiple profiles can be [exported](#) and [reported](#) on together.

Progress

When your profile first starts running, it counts all the files and folders in your profile, including those in sub-folders. Once it has counted them, a progress bar will show how much work has been completed, and how much more there is to do.

The progress bar only gives an *estimate* of progress. Files which exist inside other files (e.g. zip files) are not accounted for, and in any case, files can be added or removed from your file system while your profile is still running. In most cases, the progress bar gives a fairly good indication of the amount of work remaining to be done. The current file being analysed is also displayed, so even if the progress bar doesn't seem to be moving, you can see that it is still profiling.

Throttling back

You can control how quickly or slowly DROID processes the files in your profile, and this can be done at any time, whether your profile is running or not. This can take the load off your computer, network or disks if running it would impact you, or other users. By default, DROID works as quickly as it can, but you can tell DROID to delay for a short amount of time between each file it processes.

To slow down or speed up DROID, use the slider control at the bottom right of the main window. When the slider is at the far left of the control, DROID will work as fast as it can (a delay of zero). When the slider is at the far right, DROID will wait for one second between processing each file. Even very small delays can reduce the load on networks and file servers, so the normal useful range for throttling is usually between zero and a hundred milliseconds.


Restrictions while profiling

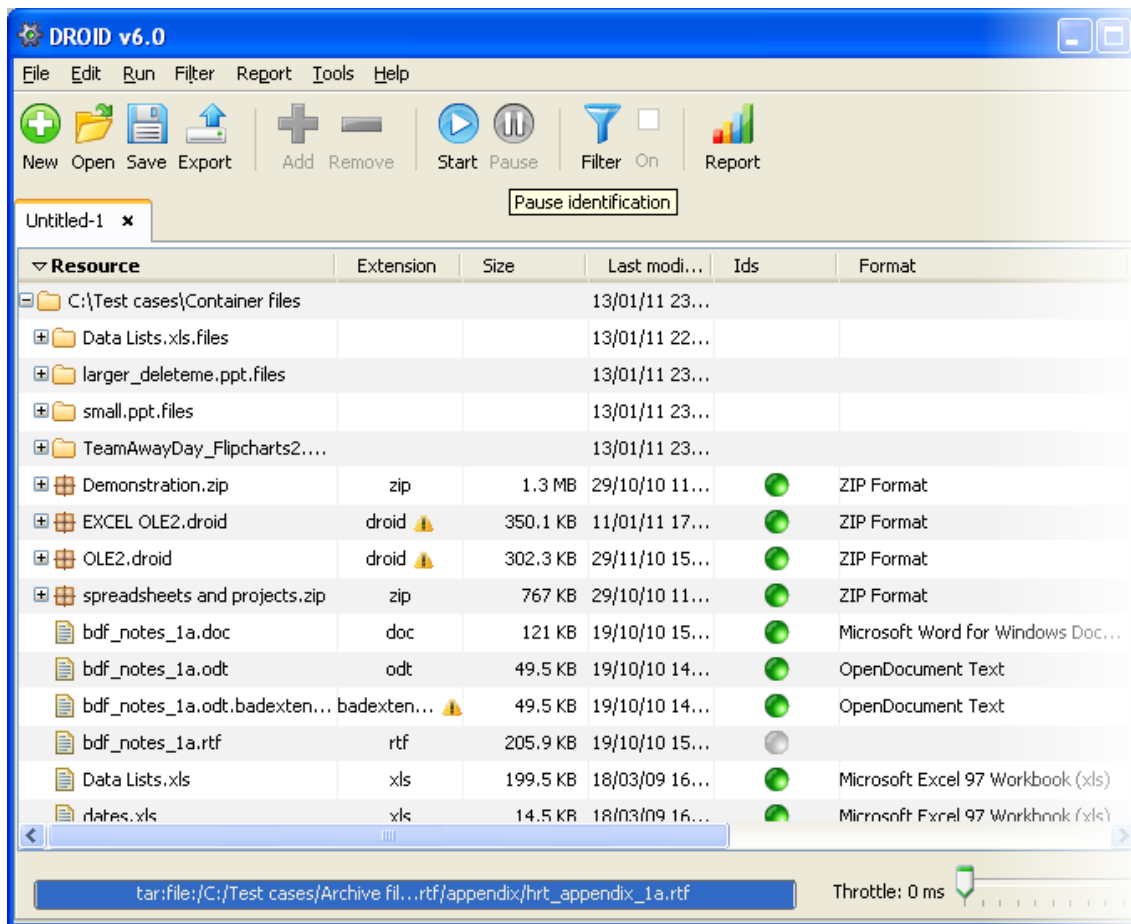
While your profile is running, its filter cannot be enabled, and you cannot save it. If you had enabled a filter previously, it will be automatically disabled when your profile is started. Disabling a filter doesn't get rid of it - it just turns it off temporarily. You can turn it on again when the profile is finished, or paused.

In addition, you cannot run any [reports](#) or [export](#) a profile while it is running. You can report and export other profiles which are not running.

If you want to do any of the above tasks while a profile is running, you can temporarily [pause](#) it, carry out the task, and then resume profiling later.


Pausing

You can pause your profile at any time by pressing the  **Pause** button, or by selecting the ***Run / Pause identification*** menu item.




The [progress](#) bar will freeze at whatever point it reached, and you will see no further messages about files being analysed in the status bar. Your profile may not pause straight away, as there may be a few outstanding items to be processed in its work queue, particularly if it is in the middle of uncompressing a large archival file. Once paused, there are no [restrictions](#) on what you can do with your profile (except that you cannot [choose](#) further files or folders once you have started profiling).

Resuming

You can resume profiling by simply [running](#) it again, using the  **Start** command. Profiling will pick up running from the point it left off, leaving all results so far intact. You can also [save](#) a paused profile, and then [open](#) and resume it at a later date. If the files where your profile was last paused are now different, DROID will attempt to resume by locating the nearest place it can successfully re-start from.

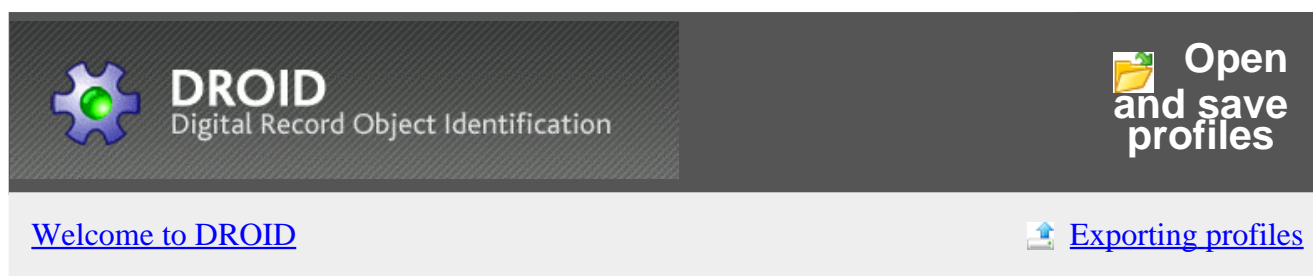
[Welcome to
DROID](#)

 [Choose files
and folders](#)

 [Running](#)


 [Pausing](#)

 [Filter your
results](#)




DROID can open and save profiles to a file with a **.droid** filename extension. Note that DROID 6 cannot open profiles created by earlier versions of DROID.

Opening a profile



To open a saved profile, press the  **Open** button, or select the **File / Open** menu item. A standard file open selection dialog will appear. Navigate to the droid profile you want to open, and press the **OK** button. The profile will open in a new tab. If it is a large profile (containing hundreds of thousands of files and folders), it may take a few minutes to open. A progress bar at the bottom of the screen shows how much of the profile has been opened so far.

Saving a profile

To save a profile, press the  **Save** button, or select the **File / Save** menu item. If your profile has never been saved before, a standard file save dialog will appear, and you can choose where to save the file and what name it has. If your profile has been saved before, it will be saved to the place it was opened from, and no file save dialog will appear. If it is a large profile (containing hundreds of thousands of files and folders), it may take a few minutes to save. A progress bar at the bottom of the screen shows how much of the profile has been saved so far.

To save an existing profile to another file, select the **File / Save As...** menu item. This will always bring up a file save dialog, allowing you to choose a different file to save to.

DROID Profile files

Profile files are actually zip files, which contain some XML files describing the profile, and a database containing any results of profiling so far. DROID currently uses the Apache Derby  database, version 10.7, which can be opened using various third-party tools, such as DB Visualizer .

The username to connect to a droid database is **droid_user**, and the password is the same as the username.

It is possible to manually edit the profile settings contained within the *profile.xml* file. However, it is **not recommended** that you do this, as changing settings within a profile may mean that

inconsistent results are returned (if the profile is paused and there are remaining results to process), or may even cause DROID to crash if the settings conflict with the profile state. In particular you must not change which signatures are used by a profile. We cannot guarantee that other settings are safe to change. Changing this file is entirely at your own risk.

[Welcome to DROID](#)



[Exporting profiles](#)



DROID
Digital Record Object Identification

**Command
line control**

[Welcome to DROID](#)

DROID can be controlled via the the command line. It is a little more limited than the full control offered by the graphical user interface (GUI). Only a single filter can be specified via the command line to apply to profiles for export or reporting, whereas in the GUI, each profile can have a separate filter.

Command line control allows DROID to be scripted into batch processes or automated work-flows. The following following command-line options are available:

- [General options](#)
- [Profiles](#)
- [Filtering](#)
- [Exporting](#)
- [Reporting](#)
- [Signatures](#)

Command line usage does not currently allow all operations to be combined in a single command-line. Some options simply print information to the screen, such as the version of DROID. For the others, you can create new profiles and save the results in one step. To filter, report or export a profile requires a second step, opening a previously saved profile to process.

General

-h, --help

Display this help. More help is available using the help menu in the graphical user interface.

-v, --version

Displays the version of the DROID software.

-q, --quiet

DROID will limit its console output to the requested information and errors only. Errors will be sent to the standard error output of the console.

Profiles

-a,--profile-resources <resources...>

Adds resources to a new profile and runs it. Resources are the file path of any file or folder you want to profile. The file paths should be given surrounded in double quotes, and separated by spaces from each other. The profile results will be saved to a single file specified using the `-p` option. For example:

```
droid -a "C:\Files\A Folder" "C:\Files\file.xxx" -p
"C:\Results\result1.droid"
```

Note: You cannot use reporting, filtering and exporting when using the `-a` option.

-R,--recurse

Recurse into all subfolders of any folder specified using the `-a` option. This means that the contents of all sub-folders of the original folder will be processed, and any sub-folders in them, and so on.

If this option is not specified, only the files directly under a folder will be processed. If `-R` is specified, files in all sub-folders (and their sub-folders, and so on) will be processed as well. Note that in the Graphical User Interface, processing sub-folders is the default, whereas in the command-line.

For example:

```
droid -R -a "C:\Files\Another Folder" -p
"C:\Results\result3.droid"
```

-p,--profile <filename ...>

When used in conjunction with reporting, filtering or exporting, `-p` specifies a list of profiles to open. The file paths of the profiles should be given surrounded in double quotes, and separated by spaces from each other. When used in conjunction with the `-a` option, the results of the new profile will be saved to that file, and you can only specify a single file. For example:

```
droid -p "C:\Results\result1.droid" "C:\Results\result2.droid" -e
"C:\Exports\combinedResults.csv"

droid -a "C:\Files\A Folder" "C:\Files\file.xxx" -p
"C:\Results\result1.droid"
```

Filtering

-f, --filter-all <filter ...>

Filter all the profiles specified using the `-p` option. Only results which match ALL filter criteria specified will appear. Filter criteria are specified using the following method:

```
"<field> <operator> <values>"
```

where `<field>` is the name of a filterable field, `<operator>` is the type of comparison to use, and `<values>` are the value or values against which the field value should be compared. The `-k` option provides information on the available fields and operators. You can specify more than one filter criteria, surrounded in double quotes and separated by spaces from each other. For example:

```
droid -p "C:\Results\result3.droid" -f "PUID any_of fmt/111  
fmt/112" -e "C:\Exports\filteredResults.csv"  
  
droid -p "C:\Results\result1.droid" "C:\Results\result2.droid" -f  
"file_size > 0" -e "C:\Exports\filteredCombinedResults.csv"
```

-F, --filter-any <filter ...>

Filter profiles as the `-f` option does, except results which match ANY of the specified filter criteria will appear.

-k, --filter-fields

Lists the available fields to use in filters and the operators which can be used with them.

Exporting

-e, --export <filename>

Export the specified profiles as a single CSV file, with one row for each file profiled. This means that if a file has multiple identifications, only a single row for that file will be written out, with the identifications added as additional columns at the right of the row. If any filters are specified, then they will apply to the exported file. For example:

```
droid -p "C:\Results\result1.droid" "C:\Results\result2.droid" -e
"C:\Exports\combinedResults.csv"

droid -p "C:\Results\result3.droid" -f "PUID any_of fmt/111
fmt/112" -e "C:\Exports\filteredResults.csv"
```

-E, --export <filename>

Export the specified profiles as a single CSV file, with one row for each format for each file profiled. This means that if a file has multiple identifications, then a separate row will be written out for each file and separate identification made. If any filters are specified, then they will apply to the exported file. For example:

```
droid -p "C:\Results\result1.droid" "C:\Results\result2.droid" -E
"C:\Exports\combinedResults.csv"

droid -p "C:\Results\result3.droid" -f "PUID any_of fmt/111
fmt/112" -E "C:\Exports\filteredResults.csv"
```

Reporting

-n, --report-name <report name>

Runs the report with the [specified name](#) on any profiles opened using the [-p option](#). If any filters are specified, then they will apply to all the profiles specified to report against. For example, this command would load the profile from the "C:\Results\result1.droid" file, run the "File count and sizes" report, and save the results to a PDF file: "C:\Reports\result1Report.pdf":

```
droid -p "C:\Results\result1.droid" -n "File count and sizes" -r
"C:\Reports\result1Report.pdf"
```

-t, --report-output-type <output type>

Sets the [output file format](#) of the generated report. If not specified, reports default to writing out as a PDF file. For example, this command would load the profile from the

C:\Results\result1.droid" file, run the "Comprehensive Breakdown" report, and save the results in Planets XML format to the file: "C:\Reports\result1Report.xml":

```
droid -p "C:\Results\result1.droid" -n "Comprehensive breakdown"
-t "Planets XML" -r "C:\Reports\result1Report.xml"
```

-r, --report <filename>

Saves the report generated to the file specified. For example, this command would load the profile from the "C:\Results\result1.droid" file, run the "Total unreadable files" report, and save the results in DROID XML format to the file: "C:\Reports\result1Report.xml":

```
droid -p "C:\Results\result1.droid" -n "Total unreadable files"
-t "DROID Report XML" -r "C:\Reports\result1Report.xml"
```

-l, --list-reports <filename>

Lists the available reports, and the output formats each report can be saved in. For example:

```
droid -l
```

Signatures

-c, --check-signature-update

Check for signature updates, but does not download them.

-d, --download-signature-update

Download the latest signature updates, if a newer version is available.

-X, --list-signature-files

Lists all locally available signature files.

-x, --display-signature-file

Displays the current default signature file.

-s, --configure-signature-file <version>

Sets the current default signature file version. For example:

```
droid -s 42
```

[Welcome to DROID](#)



DROID
Digital Record Object Identification

Information
collected by
DROID

[Welcome to DROID](#)

 [Filter your results](#)

 [Report on your results](#)

DROID collects a variety of information about your files and folders, including:

Type	File name	File name extension
Extension mismatch warning	Location	File size
Last modified date	Number of format identifications	File formats
Identification method	Content hash	Status

Type

[top](#)

DROID categorises the files and folders it profiles as being one of three types:

-  file
-  folder
-  archival file

Files have format identifications, but do not have other files or folders inside them. **Folders** do not have any format identifications or sizes, but can contain other folders, files and archival files inside them. **Archival files** are like folders, in that they can contain other folders, files and archival files inside them, but they are also files, so they have format identifications and a file size. In this version, DROID can look inside zip, tar and gzip archival files. Archival files may have other archival files nested inside them. DROID will also profile inside these, and in any further nested archival files.

File name

[top](#)

The name of a file, folder or archival file is its name, independent of its location on a disk or inside an archival file. The file name extension (if any) is part of its name. DROID treats all filenames as [case-sensitive](#). For example, “MYDOCUMENT.DOC” and “mydocument.doc” are regarded as different file names.

File name extension

[top](#)

File extensions are a convention to indicate the broad type of a file (or archival file) by appending a short string to a file name, separated by a full stop. On Microsoft Windows, the filename extension is used to indicate to the operating system what application to run when double-clicking on the file.

Other operating systems do not use the filename extension to determine which application to use. However, filename extensions have become a de-facto standard for indicating the broad type of a file format, and are usually appended to filenames, even when a file is created on other platforms.

DROID extracts the file extension (if any) from a file name or archival file name and stores it separately, to facilitate reporting, sorting and filtering on the extension alone.

File names which begin with a full stop and have no other full stops in them are not regarded as having an extension. For example, a file called “.myfile” has a filename of “.myfile” and a blank extension, whereas “.myfile.doc” has a file name of “.myfile.doc” and an extension of “.doc”. This is because file names starting with a full stop are hidden files in unix file systems, and also because it is not likely that a file name would be entirely composed of a file extension, with no name before it.


DROID treats file extensions as [case sensitive](#). However, it converts all file extensions to lower-case to facilitate filtering and reporting.



File name extension mismatch warning


[top](#)

Sometimes file extensions are incorrect for the type of the file, or are missing where there should be one. If DROID detects that the file extension for a file name does not match the formats it has identified, it will issue a file extension mismatch warning. For example, if a file called “myfile.doc” is identified as a spreadsheet, then a file extension mismatch warning will be issued.

In the graphical user interface, extension mismatch warnings appear as a warning symbol  against the file extension itself. When exported to a CSV file, it will appear as a True or False value in its own column.

Location

[top](#)

DROID records the location of every file and folder it profiles. It records location in two ways, using a file Uniform Resource Indicator (URI) , and a file path where one exists. Like file names and extensions, DROID treats file paths and URIs as case sensitive.

There are two ways of recording location because not all files and folders have a file path, although this is the usual method of identifying location in a file system. Any file, folder or archival file which is inside another archival file does not have a defined file path, as it is inside the archival file, not directly in the file system.

For example, if we have:

1. a folder called “Folder” on the “C:\” drive of a Windows computer
2. a file called “Document.doc” inside “Folder”,
3. an archival file called “Archive.zip” inside “Folder”

4. a spreadsheet called "Spreadsheet.xls" inside "Archive.zip"
5. a folder called "Another folder" inside "Archive.zip"
6. a picture called "Large picture.jpg" inside "Another folder"

Then we have the following file paths and URIs:

File path	Uniform Resource Indicator (URI)
1 C:\Folder	file:/C:/Folder/
2 C:\Folder\Document.doc	file:/C:/Folder/Document.doc
3 C:\Folder\Archive.zip	file:/C:/Folder/Archive.zip
4	zip:/file:/C:/Folder/Archive.zip!Spreadsheet.xls
5	zip:/file:/C:/Folder/Archive.zip!Another%20folder/
6	zip:/file:/C:/Folder/Archive.zip!Another%20folder/Large%20picture.jpg

Only files, archival files or folders which are directly accessible in the file system have a file path. Those files and folders which are inside the zip file do not have a file path, but do have a URI, which tells you that they are inside the zip file, where they can be found in it, and where the zip file they are inside is to be found.

The prefixes of a URI tell you what sort of resource is being described by the URI, and the exclamation marks indicate where one type of resource is contained by another. For example, for "Spreadsheet.xls", we can see that there is a file, C:/Folder/Archive.zip, with the prefix <file:/>. The exclamation mark (!) tells us that the spreadsheet is contained by the Archive.zip file, and the first prefix <zip:/> tells us the type of the containment is a zip file. Note that spaces in URIs are encoded by "%20", and folder separators are always forward slashes. If zip files are contained inside zip files, inside zip files, more prefixes and exclamation marks are added as needed.

URIs mean that all resources profiled by DROID have a unique reference which tells you where the resource is, even if it is inside an archival file, inside another archival file, and so on. This is something that file paths cannot do. However, both are provided, as working with file paths is easier, where they exist for a resource.

File size

[top](#)

The size of a file or archival file is recorded as the number of bytes used by the file. Files can have a size of zero (no content, just a record in the file system). Folders do not have a size.

The size of an archival file is the size of the archival file itself, not the sum of the sizes of its contents. For example, zip files compress their contents, so the sum of the sizes of the files inside a zip file will be bigger than the size of the archival file itself.

Last modified date and time

[top](#)

Most files, folders and archival files record the date and time on which they were last modified. This is not the same as the date a file was originally created, or the date on which a file was last read. Unfortunately, due to limitations in Java 6, DROID can only acquire the last modified date, even though the other dates may be present on the file system.

It is possible that not every file, folder or archival file will have a last modified date. For example, in some cases, resources inside archival files may not record this date.



It is important to note that last-modified dates can be changed when files are copied from one server to another, so this date may not reflect the last date a user actively modified the content of a file. Also, the content of a file (the data within it) may actually be older than the file itself – if a file was copied, or simply typed up manually from an older piece of content.

Some files may have noticeably inaccurate dates, e.g. 1 Jan 1970. In this case, the files will be newer than indicated. This error will likely be caused by the battery failing on the internal clock of the computer from which the document was uploaded, or some other error which caused the date to be set incorrectly.

Number of format identifications

[top](#)

DROID attempts to identify the format of files, including archival files, but not folders. The number of identifications DROID records for a file can vary. It can have

-  zero, if DROID can't identify a format at all.
-  one, if it is unambiguously identified as a single format.
- **(2)** more than one, if DROID can't unambiguously decide what format it is in.

In the user interface, the number in brackets indicates the number of possible format identifications made. Clicking on the link will bring up a window showing all the identifications in a table. Multiple possible identifications can happen for three reasons.

1. A format is identified purely on the basis of its file extension, so multiple versions of a file format may match the same extension.
2. A format has several versions which are very similar and hard for DROID to distinguish between, so DROID will simply report all the possible versions.
3. A file may contain patterns, purely by chance, which appear in more than one file format.

File formats

[top](#)

When DROID identifies a file format, it records four pieces of information:

- format name
- format version
- PRONOM Unique Identifier (PUID)
- mime-type.

The format name is simply a human-readable name given to a file format or family of file formats, for example, “Microsoft Word”. The format version is the version of the format, for example “97-2003”. The PUID is a globally unique, persistent identifier for a file format and version, assigned by the National Archives through its PRONOM file format registry[🔗](#). For example, the PUID for the “Microsoft Word 97-2003” file format is “fmt/40”.

PUIDs are guaranteed never to change, although new PUIDs may be defined. Clicking on a PUID in DROID will take you to the relevant page for that file format on the National Archives PRONOM website. The website will also help you with some file format names that you may be unfamiliar with. In particular, you may see files identified as 'OLE2 Compound Document Format' (PUID fmt/111) which you can interpret as 'Microsoft Office generic'. In these cases, the file is a Microsoft Office file which DROID could not identify any more closely, but the file extension may indicate more precisely.

Finally, the mime-type is another scheme for identifying broad types of files in use on the internet. They are assigned by a body called the Internet Assigned Numbers Authority[🔗](#). Mime-types are quite broad classifications, so many different file formats will have the same mime-type. For example, the mime-type for “fmt/40” is “application/msword” – which is shared by all other binary Microsoft word formats.

Identification method

[top](#)

DROID has three different methods of identifying file formats:

- extension
- signature
- container.

An “**extension**” identification means that a format was identified purely on the basis of its file extension. Such an identification may not be reliable, as files can be named in any way, and extensions do not identify formats down to the version level, so such identifications can be quite broad, and may result in multiple identifications.

A “**signature**” identification means that a format was identified by finding signature patterns inside the file which are known to occur in particular file formats and versions. This method is quite reliable, as it is fairly unlikely that by chance a file will happen to have a pattern belonging to a different file format than its own.

A “**container**” identification means that a format was identified by finding embedded files (possibly with signatures of their own) inside the main file. For example, Microsoft Office 2007 word processing files are actually zip files containing xml files, images or other resources used in the document. A container identification would identify the main file as a Microsoft Office 2007 file, not a zip file. This method is very reliable, as not only does the broad type of container have to be identified (e.g. zip), but the zip file must then be opened, and files inside scanned for further identifications to be made. The original zip identification is removed, and replaced by the Office 2007 identification, on the basis of the files discovered within it.

Note that this is not the same as profiling files inside Archival files, even though container-format files may be based on an archival format like zip. A container-format is a single file format, whose specification relies on specific files being inside it to define the overarching format. An archival file format is a format whose only purpose is to contain other files, and the particular files inside it has no effect on its identification as an archival format.

Content hash

[top](#)

DROID can optionally generate a content hash of the contents of each file and archival file, using the industry standard “MD5” algorithm. A content hash is a short signature that can be used to identify the content of the file. It is extremely unlikely that two different files will have the same content hash (although this is a remote possibility).


Content hashes can be used to detect files with duplicate content, or can be linked to forensic hash databases to find or exclude files which are widely used (and therefore not unique to your organisation) or which contain illegal content. See "[Detecting duplicate files](#)" for more information.



Content hashing is turned off by default, as producing a hash requires reading the entire file, which will slow down DROID significantly.

Status

[top](#)

As DROID profiles your files and folders, it records whether the profiling was successful or not. There are four different statuses which a file or folder can have:

Done	The file or folder was read successfully and any results found recorded.
 Not found	The file or folder was moved or deleted before it could be profiled.


 Access denied	The operating system refused read access to DROID. You will have to grant read permission to those files or folders if you want DROID to profile them.
 Error	An error occurred while trying to read the file. You may be able to determine the cause of the error by examining DROID's log files.


In the user interface, these status icons are overlaid on the files, folders and archival files as needed.

Case Sensitivity


[top](#)

All text collected by DROID is treated case sensitively, so upper case and lower case text is regarded as different. This is due to limitations of the underlying database, which must either be entirely case sensitive, or entirely case insensitive. DROID requires some fields in its database to be case sensitive in order to operate properly, which means we cannot make only some information case insensitive, even where it might be more useful to do so.

[Welcome to DROID](#) [Filter your results](#) [Report on your results](#)



DROID
Digital Record Object Identification



**Filter
your results**

[Welcome to DROID](#)
[Run your profile](#)
[Information collected
by DROID](#)
[Open and save
profiles](#)


Results

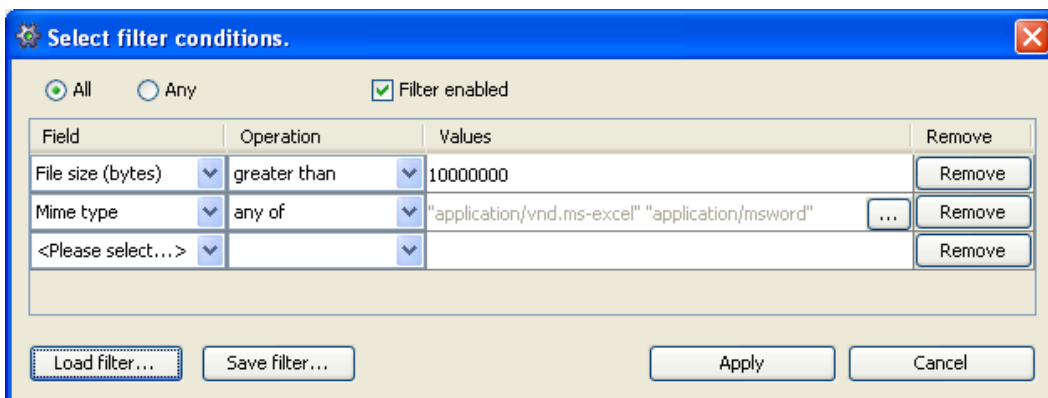
Results appear in your profile tab as soon as they are profiled. If you have profiled any folders, then you can see the files and sub-folders inside them by clicking on the open folder icon to their left. To the right of each resource are columns containing [information](#) about each resource.

Sorting

You can sort the columns by clicking on the column header. Clicking once will sort in descending order, another click will sort in ascending order, and a third click will remove the sort. The sort will also group resources that can have resources inside them (e.g. folders or zip files) together, followed by the files, to keep similar resources together.

Filtering

When you have a lot of results, it is useful to be able to filter them, in order to narrow down on the files or folders of particular interest. You can define a filter for your results by clicking the  **Filter** button, or by selecting the **Filter / Edit Filter...** menu item. In the filter definition dialog, you can add one or more conditions that a file must meet in order to be visible in your profile, change whether your files must meet all the conditions you specify, or any of them, and set whether the filter is enabled or not.



Select filter conditions.

☒ All
 ☐ Any
 ☒ Filter enabled

Field	Operation	Values	Remove
File size (bytes)	greater than	10000000	<input type="button" value="Remove"/>
Mime type	any of	"application/vnd.ms-excel" "application/msword" ...	<input type="button" value="Remove"/>
<Please select...>			<input type="button" value="Remove"/>

Any or All

- All - files must meet all filter conditions specified in order to be visible in your profile.
- Any - files can meet any filter condition specified in order to be visible in your profile.

Filter enabled

This checkbox allows you to set whether your filter is enabled or not. When enabled, results in your profile will be filtered, when disabled your filter conditions will simply not apply (but their definition will remain). You can also enable or disable filters via the **Filter / Filter on** menu item.

Adding Filter Conditions

To specify a filter condition, you must fill out three items in a row in the filter table.

1. Field - the type of information you want to filter on,
2. Operation - the type of comparison to make,
3. Values - what to compare the value of the field to.

Fields

Fields are the kind of information you want to filter on. Click on the first column of the filter condition table, on the drop down box that says "<Please select...>". You can select from the following fields:

- **File name** - the (case sensitive) name of a file or folder
- **File size** - the size of a file in bytes
- **File extension** - the (case sensitive) file name extension
- **Last modified date** - the data a file or folder was last modified
- **Resource type** - whether the resource is a folder, file or archival file (a file that contains other files - e.g. a zip file).
- **Mime type** - the mime-type of a file
- **PUID** - the PRONOM Unique Identifier of a file.
- **Format name** - the name of a format identified by a PUID.
- **Identification method** - the method by which a file or archival file was identified.
- **Job status** - whether a file has been processed, or had an error when profiled.

For more information on these fields and the information contained in them, please see "[Information collected by DROID](#)". You can specify more than one filter condition on the same field. Notice that if you are filtering using the [All](#) method, this makes it possible to create filters which no file can meet (e.g. size < 100 and size > 100)!

Operation

Once you have selected the field you want to filter on, you can specify what kind of comparison operation should be performed on it. The available operations vary by the type of field you have selected:

- **Numbers and Dates:** less than (<), less or equal (<=), equals (=), greater than (>), greater or equal (>=), not equals (<>)
- **Text** (*case sensitive*): equals (=), not equals (<>), starts with, ends with, contains, does not start with, does not end with, does not contain.
- **Sets:** some types of field can have one or more values. selected out of a defined set of values (PUID, Mime type, Identification method, Job status). To compare sets, you can specify that there are "any of" the values in the specified set, or "none of" the values in the specified set of values.

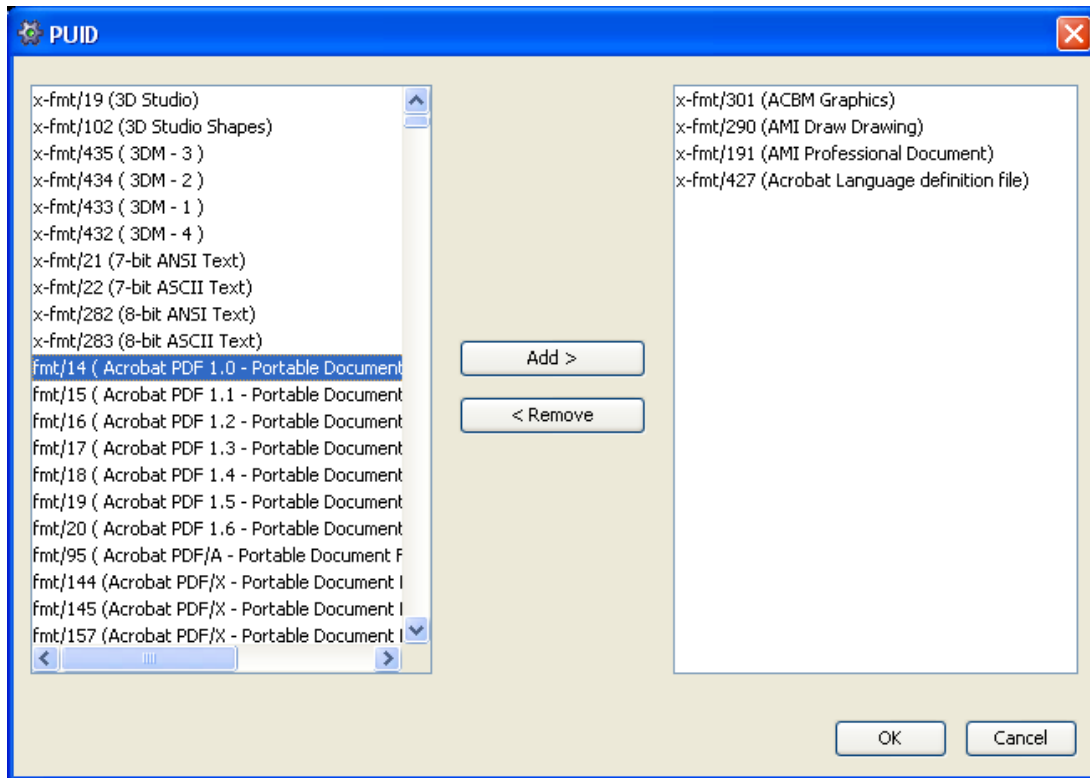
Values

Like the operations, the values you provide depend on the type of [fields](#) you are filtering on:

- **Numbers:** enter a positive whole number. Only digits can be entered - no decimal points, plus or minus signs are allowed.
- **Dates:** provide the day, month and year in the boxes provided.
- **Text:** enter your text into the box provided.
- **Sets:** You cannot enter set values directly into the value area - you must select them using the [set selection dialog](#). To bring up the dialog, click the **...** button to the right of the value area. Once you have selected your values, a read-only representation of them will appear in the value area. You can always edit your sets by bringing up the set selection dialog again.

Set selection dialog

The set selection dialog lets you select one or more values from a list.



On the left hand side of the dialog are all the available values to select. On the right hand side are the values you want to filter on. To add values to your selection, select the values you would like to add on the left hand side, and press the **Add >** button between the two panes. You can select values individually, or you can hold down the SHIFT key to select a list of values. If you hold down the CTRL key, you can select or deselect multiple items individually. To remove values from your selection, highlight the values you want to remove on the right hand side, using the same techniques as adding, and press the **< Remove** button.

Once you are happy with your selection, press the **OK** button at the bottom of the dialog. Your selected values will be placed in the value box in the filter dialog. If you want to cancel any changes you have made, press the **Cancel** button.

Removing Filter Conditions

To remove a filter condition, simply click the **Remove** button at the far right of the row. You cannot remove the final row in the filter table, which always shows "<Please select...>" in the field column, as this row enables you to add new filter conditions. Unless you actually specify a field, this row does not contribute to the filter specification.


Load and Save filters

If you have a filter you want to re-use, you can save filters to a file, and load them back from a previously saved file. Clicking on these buttons brings up a file selection dialog window.


Apply and Cancel

If you are happy with your filter definition, press the **Apply** button at the bottom of the screen. Your filter specification will be associated with your profile, and if you had enabled the filter, it will be applied to the profile immediately. If you press the **Cancel** button, all changes you have made to the filter will be discarded, and any previous filter will be restored as it was before you opened the filter dialog.

[Welcome to DROID](#)[!\[\]\(cbe80b694ebd74fcfe136a095b608235_img.jpg\) Run your profile](#)[Information collected
by DROID](#)[!\[\]\(cbe2492b119e39e02a1dab2af4a4b296_img.jpg\) Open and save
profiles](#)



DROID
Digital Record Object Identification




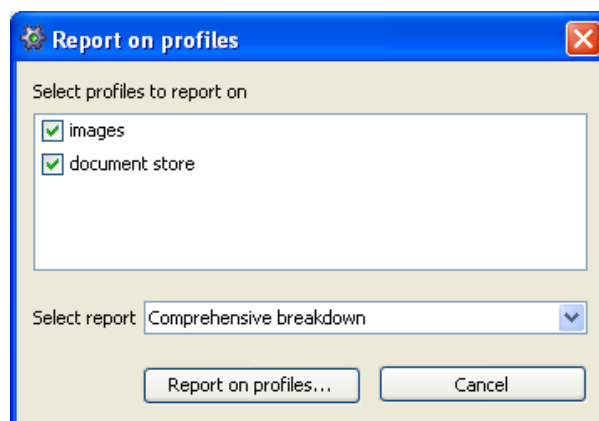
**Summary
reporting**

[Welcome to DROID](#)
[Filter your results](#)
[Exporting profiles](#)

DROID can create a variety of reports containing statistics about the files and folders in your profiles, and save the report as several different kinds of file.

Selecting profiles to report on

To create a summary report of one or more profiles, press the  **Report** button, or select the **Report / Generate Report** menu option. This will bring up the profile selection dialog, which allows you to select which profiles to report on.



If a selected profile has an active filter, this filter will be used when generating the report. So you can produce different reports on the same profiles, by using different filters. For example, you could filter out all files which are very large, giving you averages which are closer to the mean values normally encountered. Or you could filter out everything except document formats, letting you produce statistical reports on document types only.

Available reports

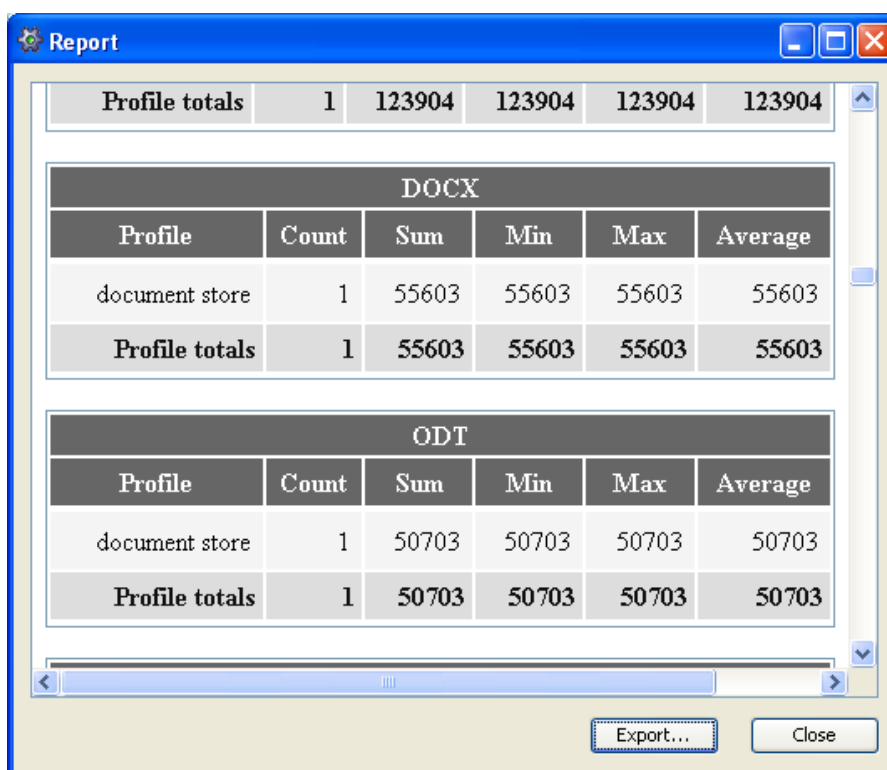
Once you have selected some profiles to report on, select which report you wish to generate. DROID ships with eleven pre-defined reports:

File count and sizes	The count, total size, and minimum, maximum and average size of all files in your profiles
Total count of files and folders	A count of all files and folders in your profiles
Total unreadable files	A count of all the unreadable files in your profiles

Total unreadable folders	A count of all the unreadable folders in your profiles
File count and sizes by file extension	The count, total size, and minimum, maximum and average size of all files in your profiles broken down by their file extensions.
File count and sizes by file format PUID	The count, total size, and minimum, maximum and average size of all files in your profiles broken down by their file format PUIDs.
File count and sizes by mime type	The count, total size, and minimum, maximum and average size of all files in your profiles broken down by their mime types.
File count and sizes by month last modified	The count, total size, and minimum, maximum and average size of all files in your profiles broken down by the month they were last modified in. Months are represented as a number from 1 (January) to 12 (December).
File count and sizes by year last modified	The count, total size, and minimum, maximum and average size of all files in your profiles broken down by the year they were last modified.
File count and sizes by year and month last modified	The count, total size, and minimum, maximum and average size of all files in your profiles broken down by the year and month they were last modified. Months are represented as a number from 1 (January) to 12 (December).
Comprehensive breakdown	A report combining all of the above reports into a single report. This report may take a long time to generate over large profiles.

Building your report

Next, press the **Report on profiles...** button to generate your report.



The screenshot shows a window titled 'Report' with a blue title bar. It contains two tables of statistics. The first table is for 'DOCX' files, and the second is for 'ODT' files. Both tables have columns for Profile, Count, Sum, Min, Max, and Average. The 'Profile totals' row is highlighted in grey.

DOCX					
Profile	Count	Sum	Min	Max	Average
document store	1	55603	55603	55603	55603
Profile totals	1	55603	55603	55603	55603

ODT					
Profile	Count	Sum	Min	Max	Average
document store	1	50703	50703	50703	50703
Profile totals	1	50703	50703	50703	50703

At the bottom of the window, there are two buttons: 'Export...' and 'Close'.

Statistics presented are broken down by profile, and aggregated across all the profiles you have selected to report on.

Exporting your report

Finally, if you want to save your report, press the **Export...** button. This will enable you to save your report in a variety of file formats. By default, all reports can be saved as:

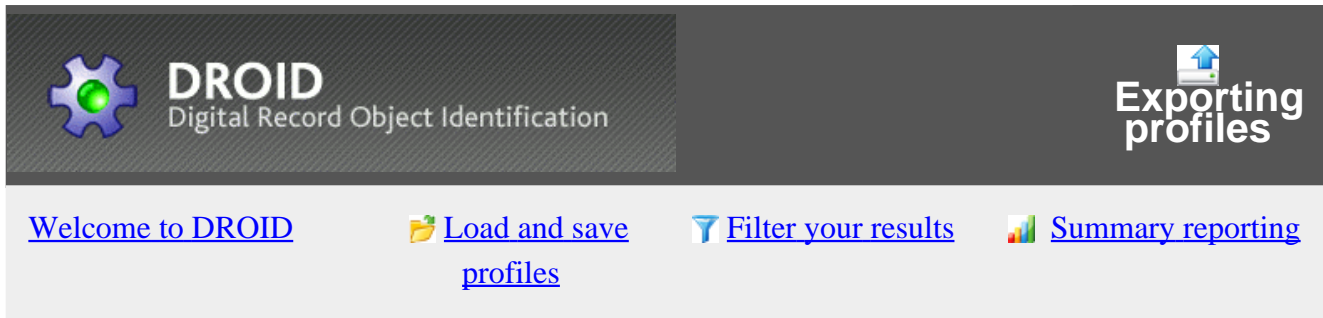
- PDF
- Web page
- Text
- DROID Report XML


In addition, some reports have special output formats defined. As shipped, DROID 6 includes a PLANETS XML export option for the Comprehensive Breakdown report. All reports are generated from the DROID Report XML, so you can use this format to transform into any other formats you require, using XSLT technology. All report definitions, and any associated transforms are located in the report_definitions sub-folder underneath your [user settings](#) folder.

[Welcome to DROID](#)

 [Filter your results](#)

 [Exporting profiles](#)



DROID can export all open profiles (which are not currently running) to a CSV file. To export a profile, press the  **Export** button, or select the **File / Export all...** menu item. This will bring up the export dialog window:



The profiles you have open are listed in the export window. If a profile is empty, or in the process of running, it will be greyed out. Select all the profiles you want to export into a single CSV file by checking the boxes next to them. If any of your profiles have active [filters](#), then the results will also be filtered. Each profile can have different filters defined and enabled.

You can also select whether the export should produce one row per file, or one row per format. When exporting one row per file, each row in the CSV file will represent a single file, folder or archival file profiled with DROID. If exporting one row per format, each row in the CSV file will be a single format identification made by DROID. Since a file can be identified as being more than one possible format, this option will produce CSV files with multiple rows for the same file (but with different identifications for it).

When you are happy you want to export your profiles, press the **Export profiles...** button. This will bring up a standard file-save dialog, in which you can specify where you want your CSV file to be saved.

CSV File Columns

ID

This is a unique number assigned to each file, folder or archival file processed by DROID.

PARENT_ID

This is the id of the archival file or folder in which this file is contained.

URI (Uniform Resource Identifier)

This provides a standard, cross-platform way of describing where resources are located. URIs are described in more detail in "[Information collected by DROID](#)".

File path

The file system location of the resource being profiled, if the resource was directly in a file system. Some files are not inside a file system - for example, files inside a zip file. In this case, the file path will be blank, as there is no file path to the resource.

Please note that file paths are platform dependent (they are different on windows and unix). DROID will write out file paths for the system on which it is currently running. This means that if you profile files on a unix machine, then export the profiles on a windows machine, the file paths will be written out as if they were windows file paths, and vice versa.

Name

The file name of the resource being profiled.

Method

This field gives the method by which a resource identification has been made. DROID can recognise resources by several methods:

- **Extension** - the file format was identified only by its filename [extension](#). This method may not be very reliable, as the filename extension may be wrong. It is only used when DROID cannot identify a resource by another method, and cannot usually identify what the version of a file format is, only its broad type.
- **Signature** - the file format was identified by finding a file format signature inside the file itself. This method is more reliable than filename extensions, and can identify the precise version of a file format.

- **Container** - the file format was identified by finding embedded files and patterns within them. This method is as reliable as signature identifications, and can be more reliable.

Job status

This field gives the identification status of a resource. It can have several values:




- **Not done** - the resource has not yet been profiled.
- **Done** - the resource was processed with no errors.
- **Access denied** - the operating system refused to give DROID read permissions to the resource.
- **Not found** - the resource was moved or deleted before DROID could read it.
- **Error** - an error occurred while trying to profile a resource. For example, if DROID is prevented from reading a file due to the user not having read permissions to it, this will result in a profiling error for that file.

Size (bytes)

The size in bytes of a file. Only files have a size - folders do not. However, note that some files can contain other files inside them, for example zip files. In this case, the zip file has a size (as it is a file), and so do the files inside it. The size reported in all cases is the uncompressed size of each file, as it would appear if extracted from the container file.

Type

DROID categorises the files and folders it profiles as being one of three types:

-  File
-  Folder
-  Archival file (e.g. zip)

Extension

The filename extension of a file, which is the last part of a filename following a full stop. Only files have extensions, as they indicate the type of the file. Even if a folder has a full stop in its name, it will not be assigned an extension.

Last modified


The date and time on which a resource was last modified. This is the only resource date-time available to DROID, as the Java 6 platform on which it runs will not report any others (e.g. creation

date-time). More file system information will be available via the Java 7 platform, which is not yet available at the time of writing. This would also require DROID to be modified to take advantage of the new file system interfaces in Java 7.


MD5 hash

If you have enabled [hash generation](#) in the preferences, then this column will contain the MD5 hash for each file and archival file processed. See "[Detecting duplicate files](#)" for more information on hashes.


PUID

The PRONOM Unique Identifier (PUID) identifies the precise file format of a profiled file. When a resource has been identified, it is assigned a PUID. A unique identifier exists for every file format that DROID can recognise, and these identifiers are maintained in the PRONOM  technical registry database, hosted at the UK National Archives.


Mime type


The mime-type of an identified file format is a high level format identifier assigned by the Internet Assigned Numbers . It is widely used in email and other internet protocols to identify the type of resource. Not all file formats identified by DROID have an assigned mime-type, and different [PUIDs](#) assigned by DROID can have the same mime-type

Format Name

The name of an identified file format, as listed in the PRONOM  technical registry against its [PUID](#).

Format Version

The version of an identified file format, as listed in the PRONOM  technical registry against its [PUID](#). Not all file formats have a defined version, so this field can be blank even when a file has a PUID.

[Welcome to DROID](#) [Load and save profiles](#) [Filter your results](#) [Summary reporting](#)

**DROID**
Digital Record Object Identification**Detecting
duplicate
files**[Welcome to DROID](#)[Information collected](#)[!\[\]\(cbe2492b119e39e02a1dab2af4a4b296_img.jpg\) Filter your results](#)

Duplicate files

It is very common to find that files are duplicated in different areas of your filing systems. Some estimates show that around 30% of all file storage consists of duplicate files. This can happen because many users save the same files from email attachments, or they take a backup copy of files while they are working on them, but don't end up changing most of them.

One method of duplicate detection is to use content hashes. If two files have the same hash value, then they are overwhelmingly likely to have identical content. The odds of two arbitrary files having the same hash value by accident are less than 1 in 18,000,000,000,000,000,000, which is very, very low. DROID can generate content hashes for your files, but note that DROID will not locate files with the same hash value for you, only generate them in the first place. If you export your profiles to a CSV file and import them into software like Excel or Access, you can query for files which have the same hash.

Another method of locating duplicate files (without using hashes) is to search for folder names containing words like, "backup", "temp" and "old", as users frequently name folders or files with these words if they intend them to be temporary copies. Another, more time consuming method, is by examining the names of files and folders. If there are areas with very similar (or identical names), then you may have duplicate information within them. However, both of these methods can only give you an indication that there may be duplication and a high degree of manual review will still be required to assure yourself that the file contents are really duplicated.

If you do find duplicate files, you must decide how to deal with them. Clearly you will need to keep at least one of them, but you will have to decide which, if any, can be safely removed. There are risks to digital continuity in deleting files, so you should take into account several considerations before deleting duplicates:

- not all users may have access to all copies of a duplicate file
- the duplicate files may have different metadata. For example, the location of a file in a filing structure can provide essential context. Or there may be important audit metadata showing that different people opened the different duplicates
- a duplicate file may provide context to the surrounding data it was stored next to. The loss of this context could render the meaning of the surrounding files unclear or unintelligible
- large areas of duplication may be for test or development environments, and therefore will be necessary for systems development.

You can mitigate some of the risks related to loss of context by leaving shortcuts (or symbolic links in a UNIX file system) to the “master file” when you delete a duplicate.

What are hashes

Hashes are long numbers, often represented as hexadecimal [text](#), which can be used as a signature to identify the content of a file. The hashes generated by DROID are called "MD5" [hashes](#), which are fairly fast to produce (relative to other hashing algorithms).

MD5 hashes are useful to locate duplicates in the files you profile, and to match with common files which have published hash values. However, MD5 hashes are not resistant to malicious attack - an attacker can create files which have the same hash but with different content. The goal of hashing in DROID is not to provide a cryptographic assurance of uniqueness, only to locate likely duplicates and to link to forensic hash databases (most of which use MD5).

Forensic Hash Databases

Forensic hash databases are published databases of hash values for files which are widely found. These can allow you to detect whether the files in your systems are common, well known files (such as Windows system files), whether they contain known illegal content, and in some cases, malware such as viruses. Knowing which files are well known outside your organisation can support information policy and decision making. For example, you may discover that a lot of storage space is being taken up with multiple copies of files which are easily replaced from install CDs. Files which are not well known probably contain unique content, and would be hard to replace if deleted.

There are a variety of content hash databases available. Two widely used hash databases are:

- [National Software Reference Library](#)
- [Hashkeeper](#)

Note that DROID does not link your files to these hash databases. It merely generates a compatible hash for each of your files, which you can then use to link with them. You will require additional technical assistance to perform these links.

[Welcome to DROID](#)[Information collected](#)[!\[\]\(fe3aebe81acea8d45108cd2768939da7_img.jpg\) Filter your results](#)

**DROID**
Digital Record Object Identification

Change
preferences

[Welcome to DROID](#)[Startup configuration](#)[Update file format signatures](#)

The preferences window allows you to configure various settings used by DROID after it has started up. There are also some settings which control [how DROID starts up](#), which are not configured through this window. There are three main tabs in the preferences window:

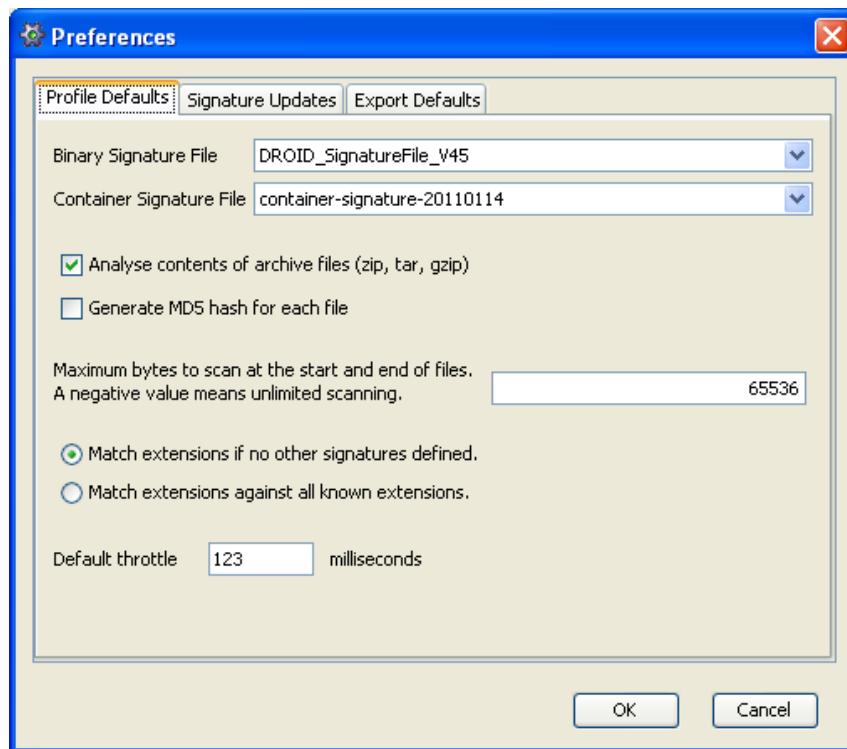
- [Profile Defaults](#)
- [Signature Updates](#)
- [Export Defaults](#)

Profile defaults

[top](#)

The Profile Defaults tab contains the settings used when creating a new profile.

- [Binary Signature File](#)
- [Container Signature File](#)
- [Analyse contents of archive files](#)
- [Generate MD5 hash for each file](#)
- [Maximum bytes to scan](#)
- [Match extensions if no other signatures defined](#)
- [Match extensions against all known extensions](#)
- [Default throttle](#)



Binary Signature File

The binary signature file is the version of the binary signatures to use when trying to identify file formats. Binary signatures are patterns to find inside the file in order to identify its file format. Once a profile has been created, you cannot change the binary signature file it uses. It is recorded with the profile when you save it.

Container Signature File

The container signature file is the version of the container signatures to use when trying to identify file formats. Container signatures identify file formats by looking for embedded files inside the main files. Binary signatures can be applied to these embedded files. This improves the accuracy of recognition for these types of file formats, as the search is confined to the specified embedded files, not to the entire containing file. Once a profile has been created, you cannot change the container signature file it uses. It is recorded with the profile when you save it.

Analyse contents of archive files.

If this option is checked, then DROID will look inside zip, tar and gzip files and profile the files it finds inside them.

Generate MD5 hash for each file

If this option is checked, then DROID will produce an MD5 hash for the content of the file. By default this option is off, as generating hashes slows down profiling significantly. Read "[Detecting duplicate files](#)" for more information on why you may want to generate hashes.

Maximum bytes to scan

When DROID is trying to identify files, it looks inside the files for common patterns. This setting allows you to configure how far from the beginning or end of the file DROID will scan before it stops trying to identify the file. This has a large effect on the performance of DROID, and a smaller effect on the accuracy of its identifications.

Almost all patterns which identify the format of files are found fairly close to the start or end of the file. By default, this setting is 65536 bytes (64KB). You can make it smaller, and DROID's performance will increase, but the accuracy of identifications may go down. Alternatively, you can make it bigger, and the performance of DROID will go down, and the identification accuracy may go up.

Setting this value to a negative number (e.g. -1), will cause DROID to scan the entire file (possibly more than once, if different patterns trigger those scans). This setting gives the maximum possible accuracy DROID can achieve, but can cause DROID to profile very slowly, particularly if you have large files.

If you do have files which are not being identified, you can increase this value, or set it to -1, to see if this has any effect on identification accuracy. If it still can't identify those files on a large setting (or the unlimited -1 setting), then you should probably choose a smaller value to increase the identification performance.

Match extensions if no other signatures defined

If this option is selected, DROID will only attempt to match file extensions for formats for which no other binary or container signatures are defined. The thinking here is that, if a presumably more accurate signature failed to match the file format, then matching against the extension alone would give potentially misleading results. This is the default behavior, and is the same as previous versions of DROID.

Match extensions against all known extensions

If this option is selected, DROID will attempt to match a file extensions against all known extensions. This may be helpful if you find you have files in newer formats, for which there is currently no good binary or container signature, or you have unusual files for which the existing signatures fail to match. Using this option will produce identifications for more files, and more multiple identifications for files.

Default throttle

This is the delay in milliseconds that DROID should pause between identifying files read from the file system. Specifying a higher delay will cause DROID to work slower, placing less load on your computer, network or disk storage. It does not cause a pause between identifying files inside archival files.

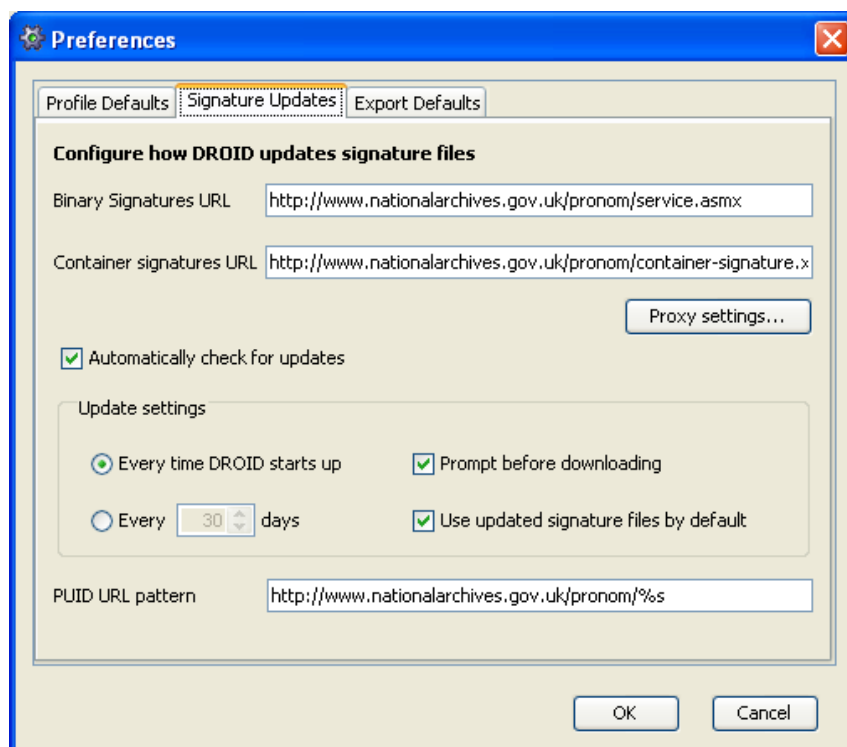
Unless you need to slow DROID down, this should be set to zero. Unlike the other profile preferences, this value can be dynamically adjusted while running using the throttle slider control on the main window. The throttle setting can be different for each profile, and will be saved with the profile.

Signature updates

[top](#)

This tab allows you to configure where DROID should try to update its signatures from, and how often.

- [Binary Signatures URL](#)
- [Container Signatures URL](#)
- [Proxy settings](#)
- [Automatically check for updates](#)
- [Update settings](#)
- [PUID URL pattern](#)



Binary Signatures URL

The web address where DROID can find a PRONOM-compatible web service to download binary signature updates.

Container signatures URL

The web address where DROID can find the container signature file. Container signature files do not require specialised PRONOM web services to download from - a file available on a standard web server is sufficient.

Proxy settings

Clicking this button will allow you to configure network proxy settings if your network has a proxy. Please consult your network administrator for details of any proxy settings you may need to make.

Automatically check for updates

If this box is checked, then DROID will automatically attempt to look for updated signatures according to the schedule you specify:

Update settings

- **Every time DROID starts up** - DROID will try to check for signature updates when it starts.
- **Every X days** - on starting up, DROID will check for signature updates after the number of days since the last check. Note that if you leave DROID running for more days than is specified, it will not automatically try to update its signatures. The check is still only made on startup, but only if the number of days since the last time it checked has elapsed.
- **Prompt before downloading** - If this box is checked, DROID will prompt you before downloading new signatures.
- **Use updated signature files by default** - If this box is checked, DROID will automatically use a newer signature file by default. If it isn't checked, it will still download the file, so it is available for selection in the profile default tab, but it will not automatically make it the default signature file to use.

PUID URL pattern

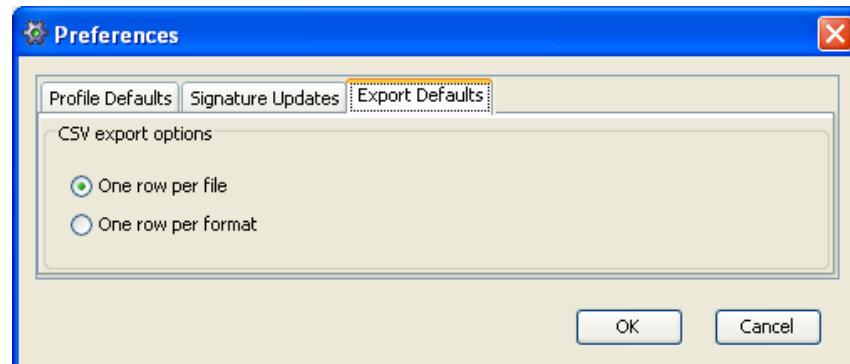
The PUID URL pattern is the web location which DROID will connect to when clicking on a PUID hyperlink in the profile tab. This should contain the URL to connect to, and the place into which the PUID should be inserted is indicated by %s. For example, if we were to click on a hyperlinked PUID of "fmt/111" in the main profile results, DROID would launch your web browser at the address: "http://www.nationalarchives.gov.uk/pronom/fmt/111".

Export Defaults

[top](#)

The Export Defaults tab contains miscellaneous settings, to configure how DROID exports CSV files by default.

- [Default CSV export options](#)



Default CSV export options

DROID can export profiles to CSV files in two ways: one row per file, and one row per format. See [exporting profiles](#) for more information on each of these options. The settings here only configure the *default* for export. When you export, you can select which you prefer on a per-export basis.

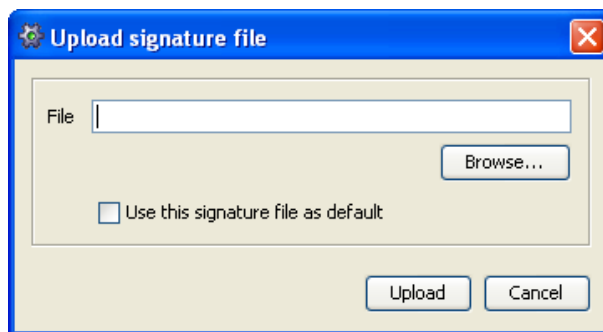
[Welcome to DROID](#)

[Startup configuration](#)

[Update file format signatures](#)



DROID can update its file format signatures using the PRONOM[®] technical registry service hosted at the National Archives. To manually update your file format signatures, select the **Tools / Check for signature updates...** menu item.




Either enter the file name and path of the signature file you want to upload, or press the **Browse...** button to bring up a file selection dialog. If you check the "Use this signature file as default", it will become the default for new profiles. If not, the signature will be available to select in the preferences should you want to use it at some point in the future, but will not be used by default. When you have selected the file to upload, press the **Upload** button, or **Cancel** to exit the dialog making no changes.

It is also possible for DROID to automatically check for and update its signatures on a regular schedule. This can be configured in the [preferences](#) section.

If you have a [binary signature](#) file locally that you want to use, you can upload this file into DROID using the **Tools / Upload signature file...** menu item. This will bring up a file open dialog from which you can select the signature file that you want DROID to use. Currently this feature cannot be used to upload [container signatures](#), only binary signature files.

Alternatively, you can manually copy a binary or container signature file into your droid working area (which is by default located in a folder called ".droid", underneath your home folder). For binary signature files, place them in the "signature_files" sub-folder. For container signature files, place them in the "container_sigs" sub folder. Note that container signature files have a number appended to them, which is the date on which they are updated, in the format year, month date. For example, the file "container-signature-20110114.xml" is the container signature file last updated on January 14th, 2011.

Binary signatures

Most file formats have either "magic numbers"  embedded within them, or they have regular structures which always appear in files of that format. The binary signatures define what patterns to find in order to identify a file format. The signatures can specify simple sequences of bytes, alternate sequences, or several different possible bytes at the same position in the file. These patterns may be found at known positions (e.g. at the start or 20 bytes in from the end of the file), or they may appear at variable positions, either within a known range of possible positions, or anywhere in the file at all. Signatures can combine all of these features, sometimes more than one in the same signature.

Taking a very contrived example, a signature may specify that there are three bytes at the start of the file: 01, 02 and 03. Then 5 bytes on there is either 04 or 05. After that, somewhere between 10 and 1000 bytes on is either 06 07 08 09, or 10, 11, 12, 13. Finally, the bytes ff, ee, dd are found right at the end of the file.

Container signatures

Some file formats are actually collections of embedded files, each with their own formats. For example, Microsoft Office 2007 files are actually zip files, which contain various other files. Taken together, they define an Office 2007 file. Binary signatures are not very good at identifying files like this, as the contents of the embedded files are obscured by the containing format (in this case, zip). In order to provide more accurate identifications, DROID can open files which use either the ZIP format, or the OLE2 format, and look inside them to see what embedded files are present.

Container signatures are just specifications of which files to look for inside the containing formats, and optionally, one or more [binary signatures](#) to run against those files if they are found.

[Welcome to DROID](#)[Startup configuration](#)[Change preferences](#)

**DROID**
Digital Record Object Identification**Startup
configuration**[Welcome to DROID](#)[Update file format signatures](#)[Change preferences](#)

There are some [settings](#) which DROID needs to be told at the point it starts up. These either relate to how Java is started in the first place, or to fundamental settings which DROID needs to know before it can do anything else. You can configure these settings in several ways, by [editing the startup scripts](#), or by [setting system environment variables](#). If any of the settings are not configured at all, they have defaults in each case.

Editing startup scripts

There are two startup scripts which can be used to run DROID:

- `droid.bat` on Windows
- `droid.sh` on Mac or Linux

Each of these files is a simple text file, which has documented sections to configure each of the settings near the start of the file. Settings provided by the scripts will override any system environment variables you have set in the system as a whole.

Note: be very careful if you edit the Mac or Linux startup scripts on Microsoft Windows systems, as this will often add Windows-specific line-endings to the file which may prevent it from working correctly on Mac or Linux systems. Some text editors let you specify the type of line-ending to write out.

System environment variables

DROID will attempt to pick up settings from system environment variables if they are not provided by a startup script. To configure system environment variables, please see the documentation for your operating system. The environment variables to use are documented against each of the settings below.

Settings

- [Maximum memory to use](#)
- [User settings folder](#)
- [Temporary files folder](#)
- [Logging folder](#)

- [Log configuration](#)
- [Console log level](#)

Maximum memory to use

Java programs can only use as much memory as the Java Runtime Environment sets aside for them. By default, this value is often quite low. We recommend that DROID runs with up to 512Mb of memory potentially available, which is the default when DROID runs if no further configuration is done.

This is the only setting which will only take effect if DROID is run using the startup scripts. This is because it is a setting Java itself needs to know, before DROID is even run.

- Default value (running through scripts and the DROID.exe file): **512Mb**
- Environment variable: ***droidMemory***

User settings folder

The user settings folder is where DROID stores user settings, including preferences, report definitions, filter definitions and file format signatures. If not otherwise configured, all the other folders and files configurable below are also stored under this folder.

- Default value: a folder called "**.droid6**" directly under the user's home directory,
- Environment variable: ***droidUserDir***

Temporary files folder

The temporary files folder is where DROID creates and deletes temporary files required during processing. Two sub-folders are created underneath this folder:

1. **/tmp** - for decompressing archival files to be further processed, and any other temporary files required. These are deleted when no longer required during processing.
2. **/profiles** - to store temporary copies of profiles DROID currently has open. Each temporary profile is deleted when it is closed.

Normally there should be no files left in these areas after closing DROID down, but in case of any problems, any files in these areas can be safely deleted if DROID is not running.


- Default value: the droid user settings folder.
- Environment variable: ***droidTempDir***

Logging folder

The logging folder is where DROID writes its log files out during operation.

- Default value: a folder called "**logs**" under the droid user settings folder.
- Environment variable: ***droidLogDir***

Log configuration

DROID uses a logging system called log4j  . You can configure different log settings by providing the path to a log configuration file, or by editing the settings in the default log configuration file.

- Default value: a file called "**log4j.properties**" under the droid user settings folder.
- Environment variable: ***log4j.configuration***

Console log level

DROID logs events to the console by default at an INFO level. If DROID is running in quiet mode from the command line, then this property is overridden to be at an ERROR level. You can configure the default log level to the console using this property in the start up scripts. Valid logging levels, from most verbose to least verbose, are DEBUG, INFO, WARN, ERROR, FATAL.

- Default value: INFO
- Environment variable: ***consoleLogThreshold***

[Welcome to DROID](#) [Update file format signatures](#) [Change preferences](#)

Index

W

- Welcome to DROID (2)
- Whats New (3)
- License (6)
- Third-party components.html (8)
- Project team (10)
- FAQ (11)