

# Registry Decoder

Instructions for Offline Analysis Component

Version 1.1

Date: 10/27/11

# 1. **Table of Contents**

2. Introduction .....	3
3. Obtaining Registry Decoder .....	4
4. Starting a New Case .....	4
Case Information Form .....	4
Adding Evidence.....	5
Processing .....	6
5. Investigation.....	6
General Notes .....	7
Hive Browsing .....	7
Path Based Analysis .....	8
Hive Searching.....	8
Hive Processing Plugins.....	10
Hive Timelining.....	11
Hive Differencing.....	11
6. Installation .....	11
7. Reporting Bugs.....	12
8. Contacting the Developers.....	12
9. Future Developments .....	12
10. Changelog.....	12
Version 1.1 .....	12

## 2. Introduction

Registry Decoder is a tool that attempts to automate the acquisition and analysis of Microsoft Windows registry files. There are two components of this tool, an online tool that collects files from a running machine and an offline tool that does the processing and analysis. This document contains the official instructions for the offline component.

For information about the online component please see <http://www.registrydecoder.com> and <http://code.google.com/p/regdecoderlive/>.

In the current version, the offline component is able to process a number of evidence types including:

1. Individual registry files
2. Full disk images
3. Partition images
4. Split images
5. Encase (E01) images
6. Databases created by the online acquisition component of Registry Decoder

The tool also performs a number of analysis tasks, including:

1. Hive Viewing
  - Similar to regedit and AccessData's Registry Viewer®.
2. Hive Searching
  - Performs full text searching across keys, values, and names
  - Creates tables of results
  - Provides automated reporting of the search term and matches
3. Plugins
  - Similar to RegRipper
  - Registry Decoder currently has 30 plugins and many more are in development
  - Provides automated reporting of plugin results
4. Hive Differencing
  - Can illustrate the differences between two registry hives using either search or plugin results as the data source
5. Timelining
  - Similar to *regtime.pl* from Harlan Carvey
6. Path-Based Analysis

- Allows exporting and viewing of paths and their key value pairs. Useful to identify if malware or other specific software pieces or events occurred on a computer

#### 7. Reporting

- Searches and plugins can be individually exported to HTML, PDF, or XLS
- “Bulk” Exports can be performed for all active analysis results tabs

The offline component of Registry Decoder has been tested against registry hives from 32 and 64-bit versions of Windows XP, Vista, and Windows 7. Searching and browsing of hives from Windows server operating systems (2003 and 2008) should also work but are not officially supported yet.

Please read the following sections for detailed instructions on how to perform investigations with Registry Decoder.

### 3. **Obtaining Registry Decoder**

To obtain the offline analysis component of Registry Decoder, please see the instructions at: <http://code.google.com/p/registrydecoder>.

### 4. **Starting a New Case**

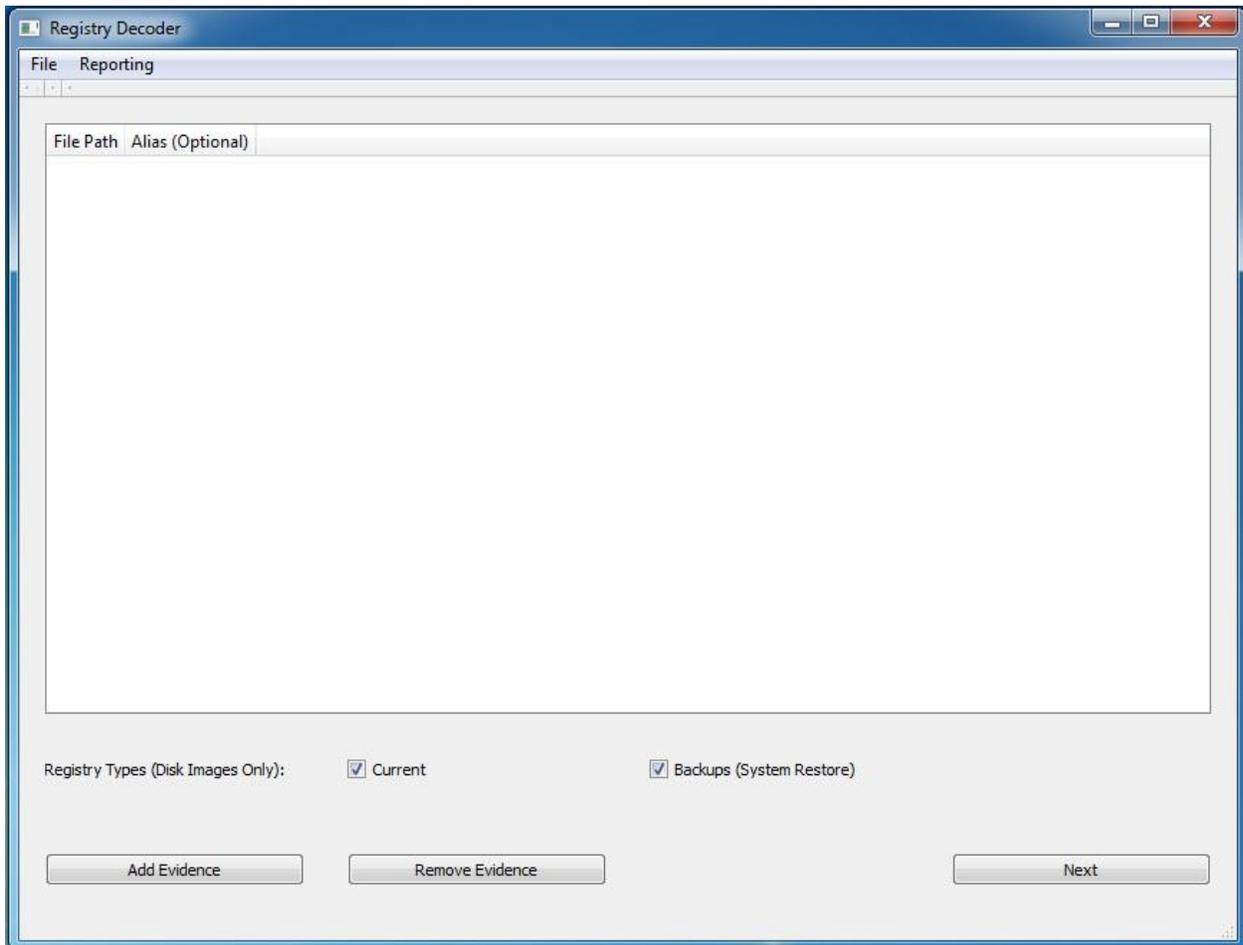
To start a new case, simply run Registry Decoder and click Next on the first form. This will then bring you to the case information form.

#### **Case Information Form**

This form is very simple and the only required field is the directory to which case data will be saved. We recommend that you fill in all fields, since they will be included within the auto-generated reports described later.

A case's files include a copy of the registry files analyzed and SQLite databases that store needed information. To proceed, simply fill out the form and then click 'Create Case'.

## Adding Evidence



The next form allows for adding of evidence to the case. To add evidence click the “Add Evidence” button and then choose any of the supported evidence types listed in the Introduction section.

If you wish to give a certain piece of evidence an alias, such as the name of the machine which a registry file came from or the name of the person to whom the disk image belongs, this information may be placed in the Alias field. This value will then be displayed throughout all interactions with the evidence as discussed throughout this document.

### *Adding Individual Registry Files and raw (dd) disk images*

Simply give the path to the file.

### *Adding a Folder from the Online Version*

To add a folder created by the online component, you must specify the path to the *registry-files/acquire\_files.db* file inside the folder.

### *Adding Encase (E01) Images*

Specify the path to the first file of the set (i.e. the one that ends in .E01). All files must be sequentially numbered and be in the same folder. This is the default naming convention for imaging tools such as FTK Imager.

### *Adding Split Images*

Specify the path to the first file of the set. Files must be in the format of the extension being only numbers, such as image.dd.001. This is the default naming convention for tools such as dcfldd.

### *Adding Evidence after a Case has been Created*

To add evidence to a case after initial processing, use the 'Start Case' option on the front screen, and then on the Case Information form choose the existing cases folder. You will be prompted if you are adding new evidence to a case, simply click 'Yes'.

From here you can add evidence as normal and it will become part of the analysis when its done being processed.

NOTE: There is no "undo" once evidence is added. Backing up cases is trivial and is recommended before adding new evidence.

### *General Notes*

Note that for disk images, the two checkboxes (current and backup) determine which files will be acquired.

Current files include those that would be active on the running machine:

- Everything under c:\windows\system32\config
- All ntuser.dat files

Backups will attempt to gather files from all system restore points on XP machines and the Reg-Back folder of Vista & Windows 7.

Once all evidence is added, click "Next".

## **Processing**

To begin processing on the next form, first review the given information and then click the 'Starting Processing' button. The evidence will then process, and when completed, will open the investigation tabs.

## **5. Investigation**

All investigation within Registry Decoder is performed within the tabbed analysis interface.

## General Notes

Before describing each analysis component, we first cover some general information about the analysis capabilities.

First, each analysis tab contains a tree view of the evidence loaded into the case. Whenever files are being chosen for analysis, they will be selected from this tree. These trees support a wide range of selection options, including:

- Selecting a single file
- Selecting a ‘group’ of files, which will then cause the chosen analysis to be run on all files in the group. This can include entire disk images or groups within a disk image.
- Selecting multiple files or groups throughout the file tree.
- Selecting ‘All files’ will run the analysis against all files in the case. Warning: This can produce a large amount of data depending on the number of evidence files!

Second, all tabs generated during analysis can be safely closed, but the initial set of tabs may not be closed. Tabs can be closed automatically using the CTRL+W shortcut.

Third, backups of cases can be made through the ‘File’ menu once a case has been loaded. The backup process will create a ZIP file with the chosen name of all files in a case directory. This directory can be later decompressed and opened by Registry Decoder on any machine.

Fourth, a case can be closed at anytime by choosing ‘Close Case’ from the File menu. After being clicked, the GUI will switch to the initial form where you can open a new case or create a new one.

## Hive Browsing

Hive browsing performs function similar to tools such as regedit or Access Data’s Registry Viewer®. To browse a file, simply choose it within the presented evidence tree and click ‘View’.

The automatically generated Browse tab presents an interface similar to that of other registry hive browsers. The left pane shows the hive keys as a tree. When a key is chosen, its full path within the registry and last written time are placed in the label at the bottom of the form. Data from this form can be copy/pasted, but not edited. The right panes show the keys, names, and data for the chosen key. These columns can be sorted. When a value is clicked in the right page, a hexdump of its contents will be shown in the bottom right table.

To instantly check if a path exists in a file, and jump to it if it exists, right click in any place on the tree and enter the path you wish to check. If it exists, the tree will be repositioned at that location. DO NOT include the “root” of the tree when searching, such as “\$\$\$PROTO.HIV” for XP hives.

## Path Based Analysis

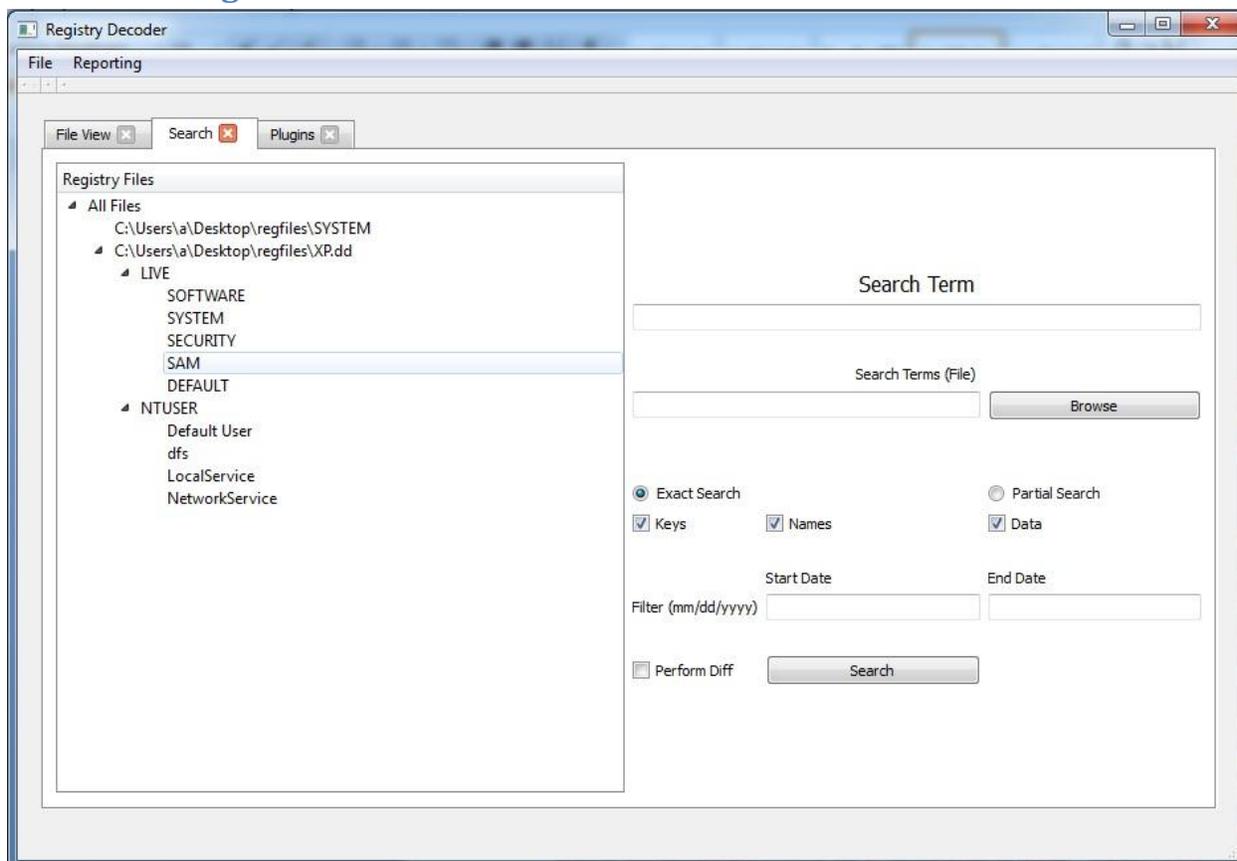
The path analysis form allows for determining if specific path(s) are within investigated registry files. It also allows exporting of information about paths along with their name/value pairs. This information is very useful in a number of situations such as:

- Showing that malware was or wasn't installed
- Showing if certain applications were installed and their corresponding parameters
- Exporting and analyzing data not covered by a plugin

To use this form, simply choose the registry files of interest and enter either a path to search or a newline-separated file with a list of paths. Results can be filtered by the key's last write time through using either the Start Date and/or End Date fields. The "Include Values" checkbox controls whether or not key/value pairs are included in the results and report.

Note: When entering the path, do NOT include the "root" of the tree, such as "\$\$\$PROTO.HIV" for XP hives.

## Hive Searching



This tab allows for searching of data contained within the analyzed hives. Files can be selected using the same methods as the Hive View. Individual search terms may be entered in the input

box under “Search Term” or a newline delimited file of search terms may be uploaded using the ‘Browse’ button.

Search behavior is influenced by the selection of either “Exact” or “Partial” search, which controls whether wildcards are added to the search parameter. When “Partial” is selected, users can also all their own wildcards. “\_” is used as a single character wildcard and “\*” is a full wildcard.

The “Keys”, “Name”, and “Data” check boxes control which registry entries are searched. The “Filter by Date” option allows for filtering of results by the last written time of the registry keys which match the search. The ‘Perform Diff’ option is explained in the ‘Hive Differencing Section’.

Once a search is performed, if match(es) are found, they are populated into a table in an automatically generated form. This form displays the search term used along with the resulting hits. The table used to display the search hits contains columns of last write time, key, name, and data. These show the respective values for the hits and the last write time of the rows key. The column which produced the search hit is displayed in bold. These columns can be sorted by clicking on the respective header.

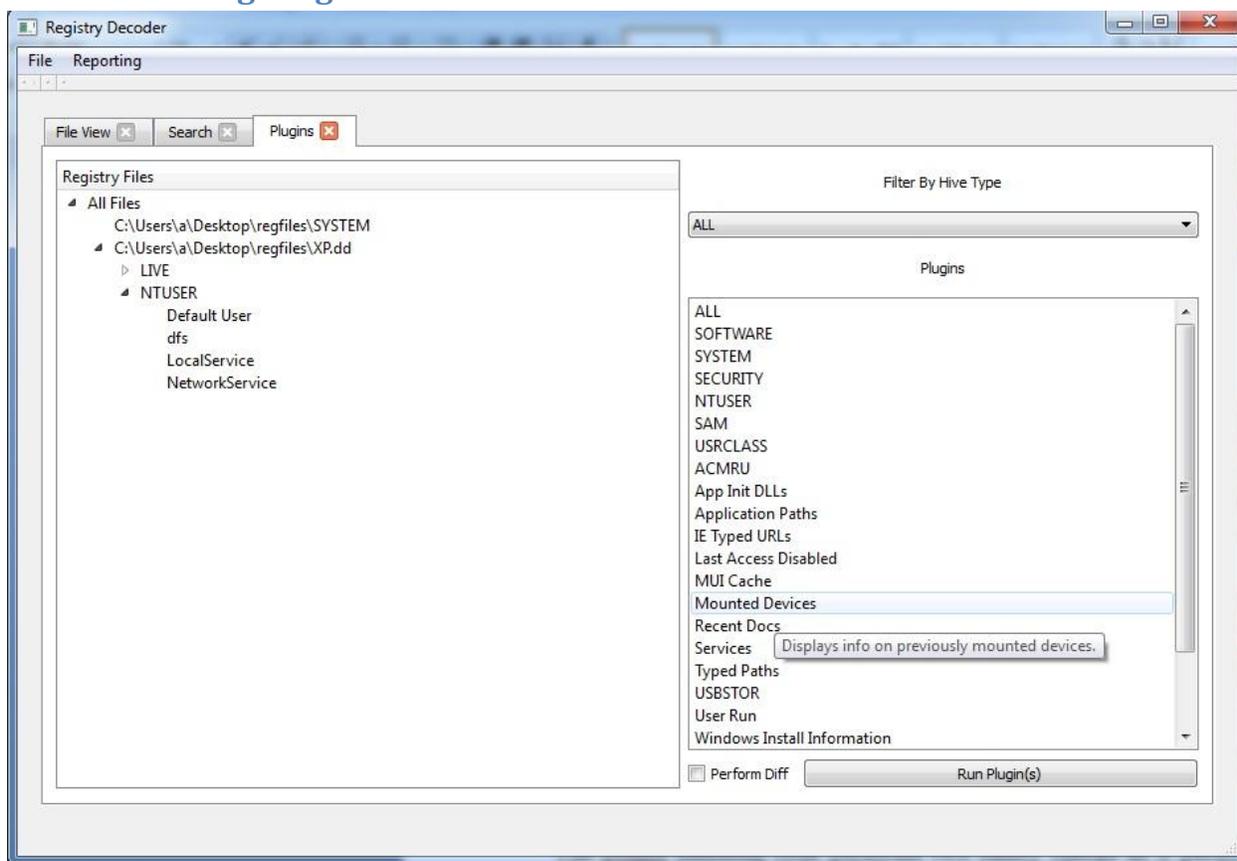
Users can also right click on a particular search entry and choose ‘Switch to File View’ and a Browse tab will then be automatically generated and opened at the key of the chosen search hit. This allows you to instantly locate relevant data within a hive structure.

Reporting of displayed search hits can be automatically performed by choosing the report format and then clicking on the input to choose a filename. The ‘Create Report’ button will then create the report in the specified file.

Creating a report of all active search tabs can be done by choosing ‘Reporting’ from the top menu and then ‘Export Searches’. The auto-generated tab then requires a filename and after clicking ‘Create Report’ a report of all open search tabs will be made.

Results that are irrelevant or unwanted within a report can be removed by pressing the ‘delete’ key on a particular entry. This will remove the result from both the GUI view as well as any reports that are generated. Users can also use the CTRL and Shift keys in the normal ways to select a subset of the result for reporting.

## Hive Processing Plugins



This tab allows provides access to Registry Decoder's plugin system. The functionality of many of these plugins are similar to the plugins provided in RegRipper, although we also provide functionality not present in RegRipper. Plugins allow for very specific analysis of a certain subset of registry data. Common examples include recovering USB device history, MRU lists, or Typed URLs.

On the Plugin tab, the 'Hive Type' dropdown menu allows you to filter which plugins are shown in the 'Plugin' dropdown by hive type. The 'Plugin' listbox controls which plugins are run. Individual plugins can be run, multiple plugins can be selected, or the set of plugins that can analyze a specific hive type (software, system, etc) can be chosen.

Hovering over a particular plugin will provide a tooltip containing information about what data the plugin collects.

The plugin handling code performs two sanity checks as it performs analysis. The first is that plugins are only run against hive types that they support, in order to ensure forensic soundness. Second, even if the same plugin is chosen multiple times, for instance if SOFTWARE is chosen as well as Application Paths, the plugin will only be run once.

Once a plugin is run, tabs are automatically generated with the results. These results can be exported and explored in the same manner as the search results. Unwanted report results can also be deleted from the results table, just as in the Search output, by pressing the ‘delete’ key on individual entries.

### Hive Timelining

The Timeline tabs allows for timelining of hives based on the last write time of keys. Keys can be filtered by using the Start Date and/or End Date fields. This analysis does not generate a result and instead writes directly to an output file as *regtime.pl* by Harlan Carvey. The output file can be then be used conjunction with the Sleuthkit suite of timelining tools.

### Hive Differencing

Registry Decoder can also perform “differencing” of registry files using either search terms or plugins. Both of these forms contain a checkbox, ‘Perform Diff’, that if clicked, will present a second tree from which to select a registry. An investigator can then chose one hive in the original tree and one hive in the newly generated tree and perform analysis based on the differences between the two. Differencing between search and plugin results can be performed in the same way.

To show the differences between the registry files (or plugin/search results), a color scheme is used. Results from the search or plugin that are only in the first file chosen are RED, results that are common between both files are BLACK, and results that are only in the second file are BLUE. These results can then be sorted and the colors will be preserved.

The Differencing facility provides powerful analysis capabilities, since investigators can quickly sift through time-varying data on a single machine (by comparing a current registry hive to a historical one) or investigate activity across multiple computers (such as users under investigation sharing USB drives or recently accessed documents).

We currently do not support reporting of difference data, but these capabilities will be included in a future release.

## 6. Installation

Windows users can simply download the pre-compiled binary and begin analysis. This binary is created by PyInstaller and requires no external dependencies.

Users who want to work from a source checkout are required to install Python 2.6.x or 2.7.x as well as a number of third-party applications and libraries:

1. Python ReportLab (<http://www.reportlab.com/software/opensource/>)
2. The Sleuthkit 3.2 (<http://www.sleuthkit.org/>)
3. PyTSK (<http://code.google.com/p/pytsk/>)
4. Reglookup and PyRegFi (<http://projects.sentinelchicken.org/reglookup/>)

5. PyQT (<http://www.riverbankcomputing.co.uk/software/pyqt/intro>)
6. python-xlwt
7. libewf and ewf.py from PyTSK

We are currently working with Linux package maintainers in order to get Registry Decoder within their software repositories.

## 7. Reporting Bugs

To report bugs in the offline component of Registry Decoder, please either send an email to [registrydecoder@digdeeply.com](mailto:registrydecoder@digdeeply.com) along with the error log created by the application or file an official bug report at: <http://code.google.com/p/registrydecoder/issues/list>.

## 8. Contacting the Developers

To contact the developers of Registry Decoder, please email [registrydecoder@digdeeply.com](mailto:registrydecoder@digdeeply.com).

## 9. Future Developments

To see the ongoing development of Registry Decoder and the roadmap of future features please visit the project tracker at:

<http://code.google.com/p/registrydecoder/issues/list>

## 10. Changelog

The following section documents major changes and bug fixes that occurred between stable releases.

### Version 1.1

- 1) Exploration of paths by right clicking in Browse view
- 2) Full wildcard search
- 3) Limit reported items by selecting within the results table
- 4) Add support for Encase (E01) and split images
- 5) Plugins can now be run from the command line
- 6) Timelining
- 7) Path based analysis
- 8) Reports now contain the local time of creation
- 9) Evidence can be added to an existing case
- 10) Cases can be closed & re-opened without closing the GUI
- 11) Name and value pairs can be sorted
- 12) Dual boot machines are now support in the file view
- 13) File view reworked to be clearer & easier to use

