Registry Explorer

RECmd

SA Eric R. Zimmerman
Federal Bureau of Investigation
eric.zimmerman@ic.fbi.gov
saericzimmerman@gmail.com
801-514-4064

Revision history

07/01/2015 Rev. 1 – Initial release

Table of Contents

Requirements

Registry Explorer and RECmd require Microsoft .net framework version 4.5.2 full runtime or greater to be installed. It is available at https://www.microsoft.com/en-us/download/details.aspx?id=42642.

Why another Registry tool?

The need for Registry Explorer and RECmd rose out of writing a fully managed offline Registry hive parser in C#. Existing parsers did not offer the features I was looking for and as such, research and coding began. The Registry project serves as the basis for several programs including ShellBags Explorer, AppCompatParser, etc. Once the back end was mature, I wanted an easy to use and powerful way to expose the capabilities of the parser.

Registry Explorer fills the gaps in existing tools and expands the capabilities of Registry viewers in many unique and powerful ways. It is GUI based and contains powerful searching, filtering, and other visualization concepts that makes exploring Registry hives very easy while exposing all of the technical information contained in Registry hives.

RECmd was created in order to be able to script access to Registry hives, conduct new research, and automate searching across multiple Registry hives at once from the command line.

Because both tools use the same back end, both have the same searching and viewing capabilities including the full recovery of deleted keys and values. The parser also exposes value slack.

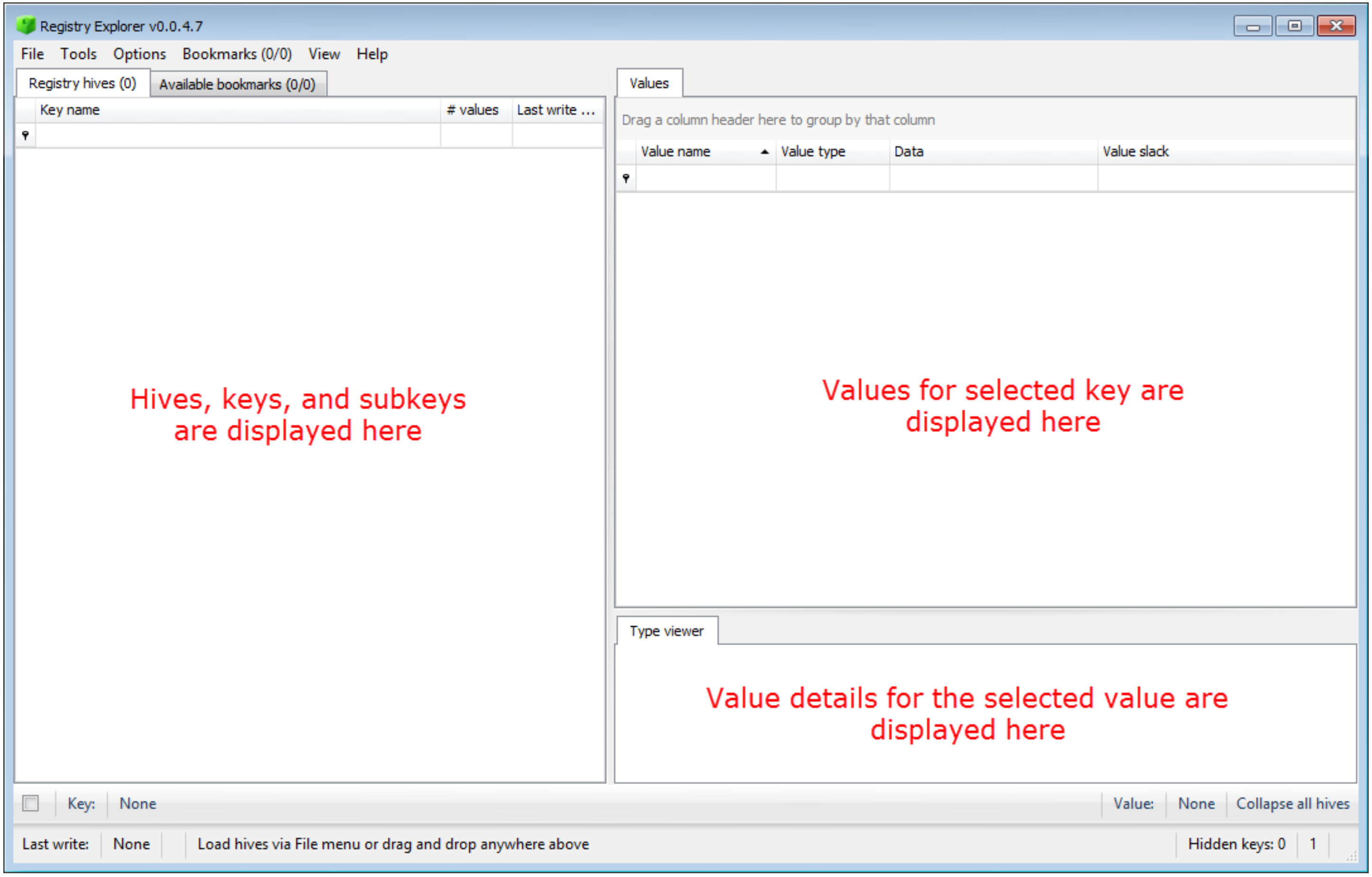
In summary, the capabilities of Registry Explorer and RECmd allows for quickly examining multiple hives at once and they can be leveraged to find new places where currently understood data is located in an easy to use and systematic way. It can be used in educational settings to not only understand the Registry from a functional level, but also from a deeply technical perspective.

Registry Explorer

Registry Explorer is a GUI based tool used to view the contents of offline Registry Hives. It has the ability to load multiple hives at once, search across all loaded hives using strings or regular expressions, exporting of data, and much more.

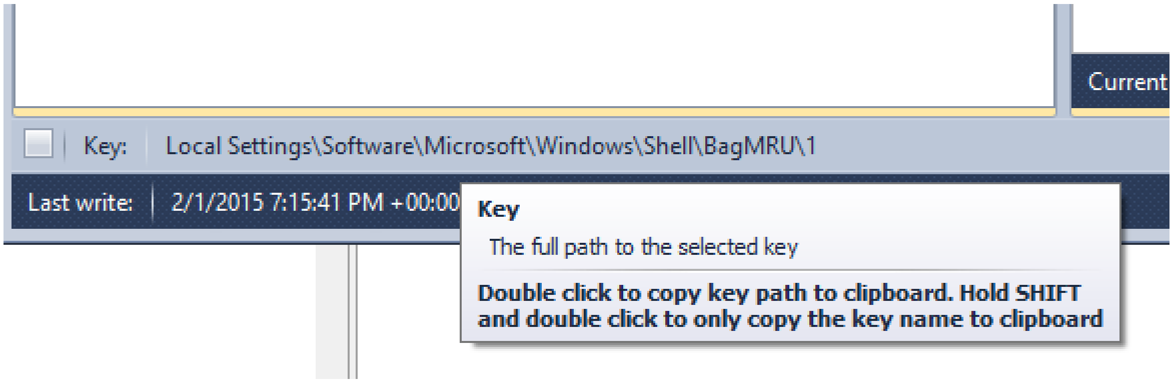
Getting started

After starting Registry Explorer, the main interface is displayed.


Registry Explorer v0.0.4.7
File Tools Options Bookmarks (0/0) View Help
Registry hives (0)
Available bookmarks (0/0)
Key name
# values
Last write ...
Values
Drag a column header here to group by that column
Value name
Value type
Data
Value slack
Hives, keys, and subkeys are displayed here
Values for selected key are displayed here
Type viewer
Value details for the selected value are displayed here
Key: None
Value: None
Collapse all hives
Last write: None
Load hives via File menu or drag and drop anywhere above
Hidden keys: 0
1

Settings for various things like program options, window size, slider positions, window positions, recent searches, etc. are all saved and reloaded between program executions. You can reset these options by deleting the relevant files under the Settings directory in the main Registry Explorer folder. The .layout files are for the trees and grids.

Tooltips are shown when hovering over different areas of the program. For example, hovering over the Key section of the status bar shows the following:


Current
Key: Local Settings\Software\Microsoft\Windows\Shell\BagMRU\1
Last write: 2/1/2015 7:15:41 PM +00:00

Key

The full path to the selected key

Double click to copy key path to clipboard. Hold SHIFT and double click to only copy the key name to clipboard

Interface sections

There are five sections to the main interface.

Registry hives

On the left side of the window is the Registry hives tab. This tab displays the Registry hives that have been loaded and the keys contained therein. Once at least one hive is loaded and a key is selected, a context menu is available by right clicking on a key. The context menu options will be discussed below in the Key context menu section.

Available bookmarks

Next to the Registry hives tab is the Available bookmarks tab. This tab will be discussed in detail below.

Values

The Values grid shows all of the values contained in the key that is selected in the Registry hives tab. Once a value is selected, a context menu is available by right clicking on a value. The context menu options will be discussed below.

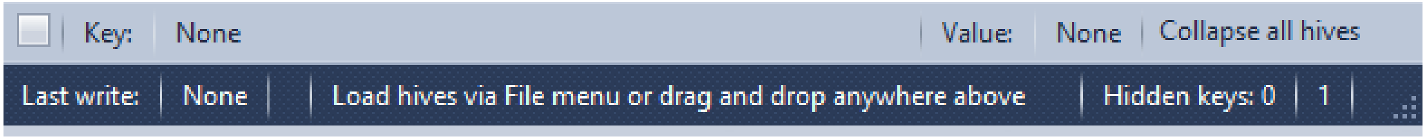
Value details

The Value details area contains one or more tabs that dynamically adjust depending the type of value selected. In every case, a type viewer will be displayed that shows the value of the selected key. If a value has slack, a separate tab will be shown that allows you to view the slack space in a hex viewer.

These concepts will be explained in more detail in the Using Registry Explorer section below.

Status bars

Across the bottom of the interface are several status bars as seen below.

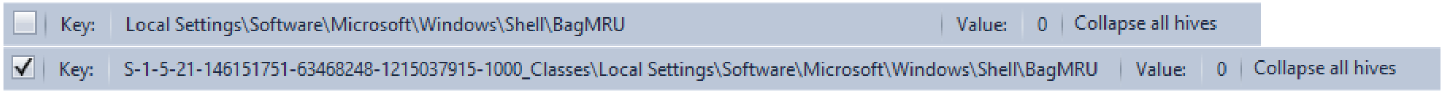


□ | Key: | None | Value: | None | Collapse all hives

Last write: | None | Load hives via File menu or drag and drop anywhere above | Hidden keys: 0 | 1

Top status bar

The top status bar contains details about the path to the selected key and the selected value. On the far left is a check box that toggles whether to show the root key name in the key path. By default, the root key path is not shown. The screen shot below shows what this option does when turned on and off.



□ | Key: | Local Settings\Software\Microsoft\Windows\Shell\BagMRU | Value: | 0 | Collapse all hives

☑ | Key: | S-1-5-21-146151751-63468248-1215037915-1000_Classes\Local Settings\Software\Microsoft\Windows\Shell\BagMRU | Value: | 0 | Collapse all hives

By hiding the root key name, longer key paths will not be truncated as different keys are selected.

To the far right of the top status bar is a button that, when clicked, will collapse all loaded hives back to their default state. This is a handy shortcut to clean up the Registry hives tree after interacting with it and expanding many keys and subkeys.

Double clicking the key path will copy the key path to the clipboard. Holding **Shift** and double clicking will copy only the key name to the clipboard.

Double clicking the value will copy the value's name to the clipboard. Holding **Shift** and double clicking will copy the value's data to the clipboard.

Bottom status bar

The bottom status bar contains the last write timestamp, the status of filters for values, a section for general status messages, an indicator of the total number of keys that are hidden from view, and the total number of messages available on the Messages form.

Double clicking on the Total messages counter will show the Messages form.

Double clicking the last write timestamp will copy it to the clipboard.

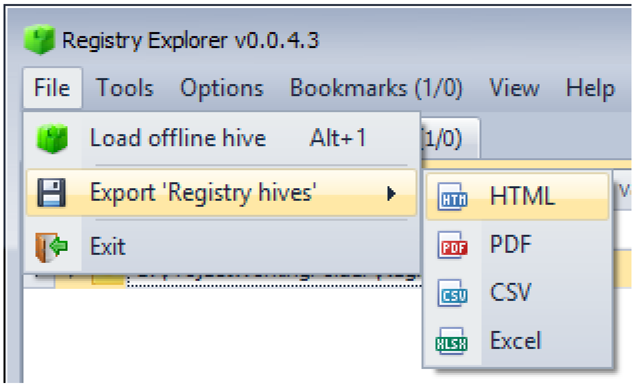
Main menu

The main menu contains options that allows for loading hives, searching hives, opening bookmarks, and so on. In many cases, the menu items will have shortcut keys associated with them. Pressing the keys shown by a menu item on the keyboard will activate that menu item.

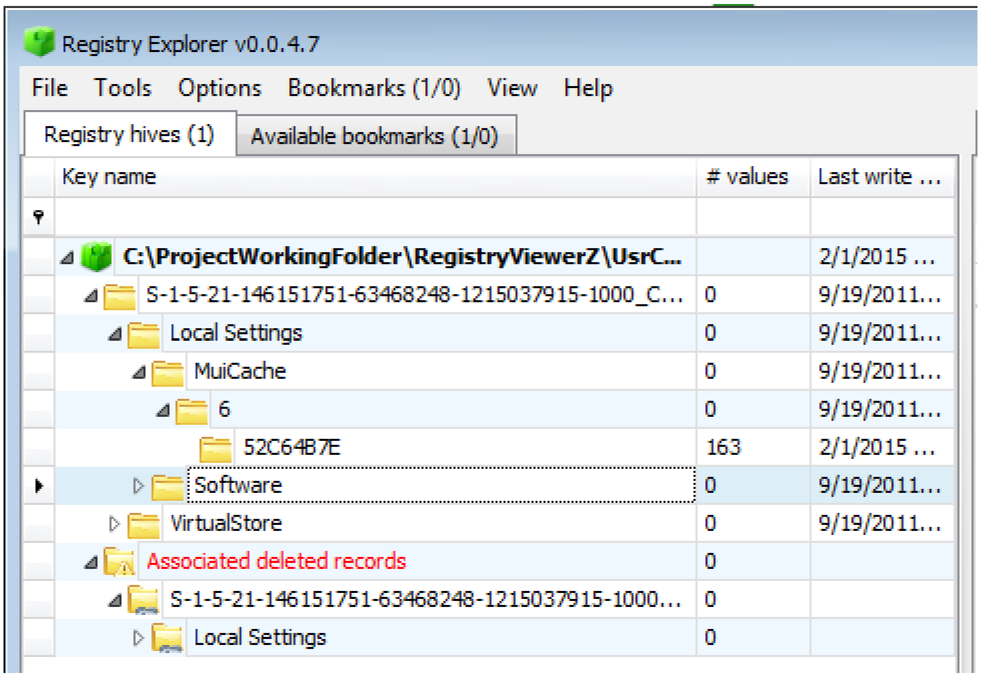
The various sections below will explain these submenus. Where things are obvious (like File | Exit), no additional information will be provided.

File

The File menu contains options for loading hives (you can also simply drag and drop one or more hives onto the main interface to load them) and exporting.

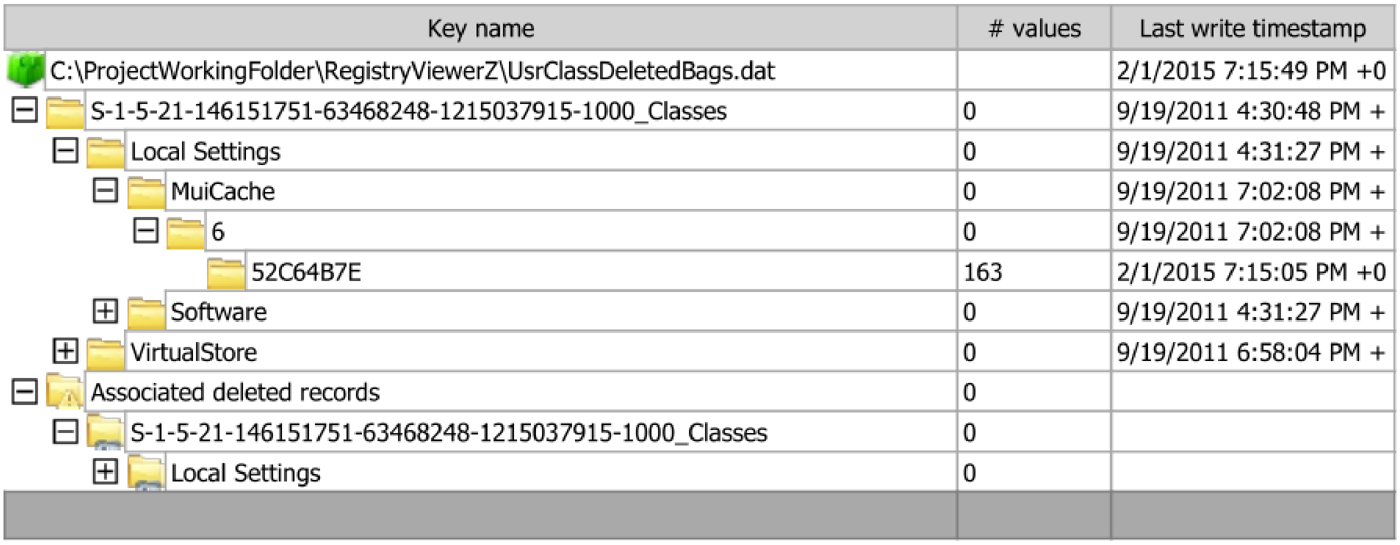


- **Load offline hive:** Allows for loading one or more hives. To select more than one file, select a file, then hold **Shift** and select the last file to load. You can also hold **Ctrl** and click files to select them individually.
- **Export 'Registry hives':** Exports what is shown in the Registry hives tab to a variety of formats. As an example, if the Registry hives tree looked like this:



Key name	# values	Last write …
C:\ProjectWorkingFolder\RegistryViewerZ\UsrC...		2/1/2015 …
S-1-5-21-146151751-63468248-1215037915-1000_C…	0	9/19/2011…
Local Settings	0	9/19/2011…
MuiCache	0	9/19/2011…
6	0	9/19/2011…
52C64B7E	163	2/1/2015 …
Software	0	9/19/2011…
VirtualStore	0	9/19/2011…
Associated deleted records	0	
S-1-5-21-146151751-63468248-1215037915-1000…	0	
Local Settings	0	

Exporting to PDF would generate a PDF file that contains the following:

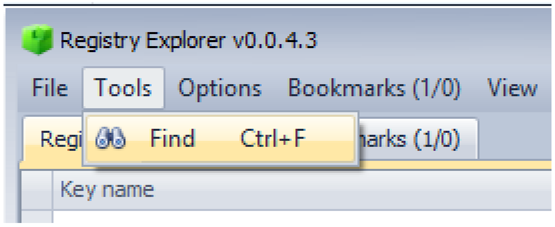


Key name	# values	Last write timestamp
C:\ProjectWorkingFolder\RegistryViewerZ\UsrClassDeletedBags.dat		2/1/2015 7:15:49 PM +0
S-1-5-21-146151751-63468248-1215037915-1000_Classes	0	9/19/2011 4:30:48 PM +
Local Settings	0	9/19/2011 4:31:27 PM +
MuiCache	0	9/19/2011 7:02:08 PM +
6	0	9/19/2011 7:02:08 PM +
52C64B7E	163	2/1/2015 7:15:05 PM +0
Software	0	9/19/2011 4:31:27 PM +
VirtualStore	0	9/19/2011 6:58:04 PM +
Associated deleted records	0	
S-1-5-21-146151751-63468248-1215037915-1000_Classes	0	
Local Settings	0	

This is useful for generating reports or other documentation that is easier to manipulate than simply taking a screen shot.

Tools

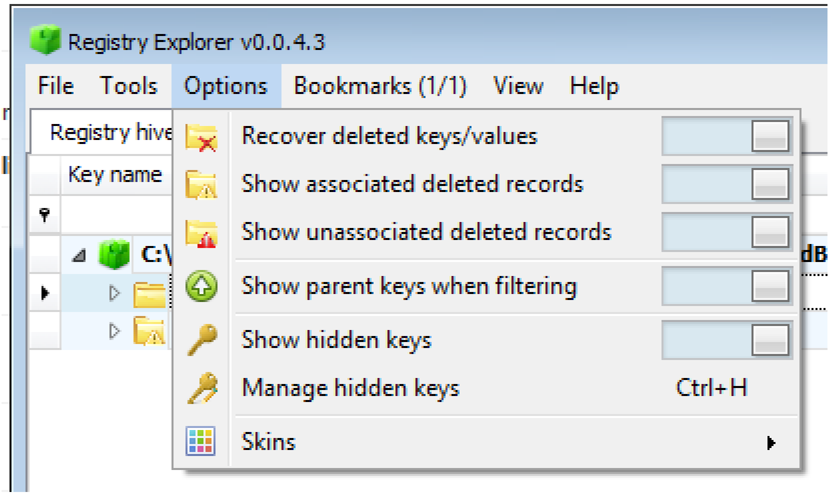
The Tools menu contains a single item, Find.



Using this option will be explained in full detail below in the Using Registry Explorer section

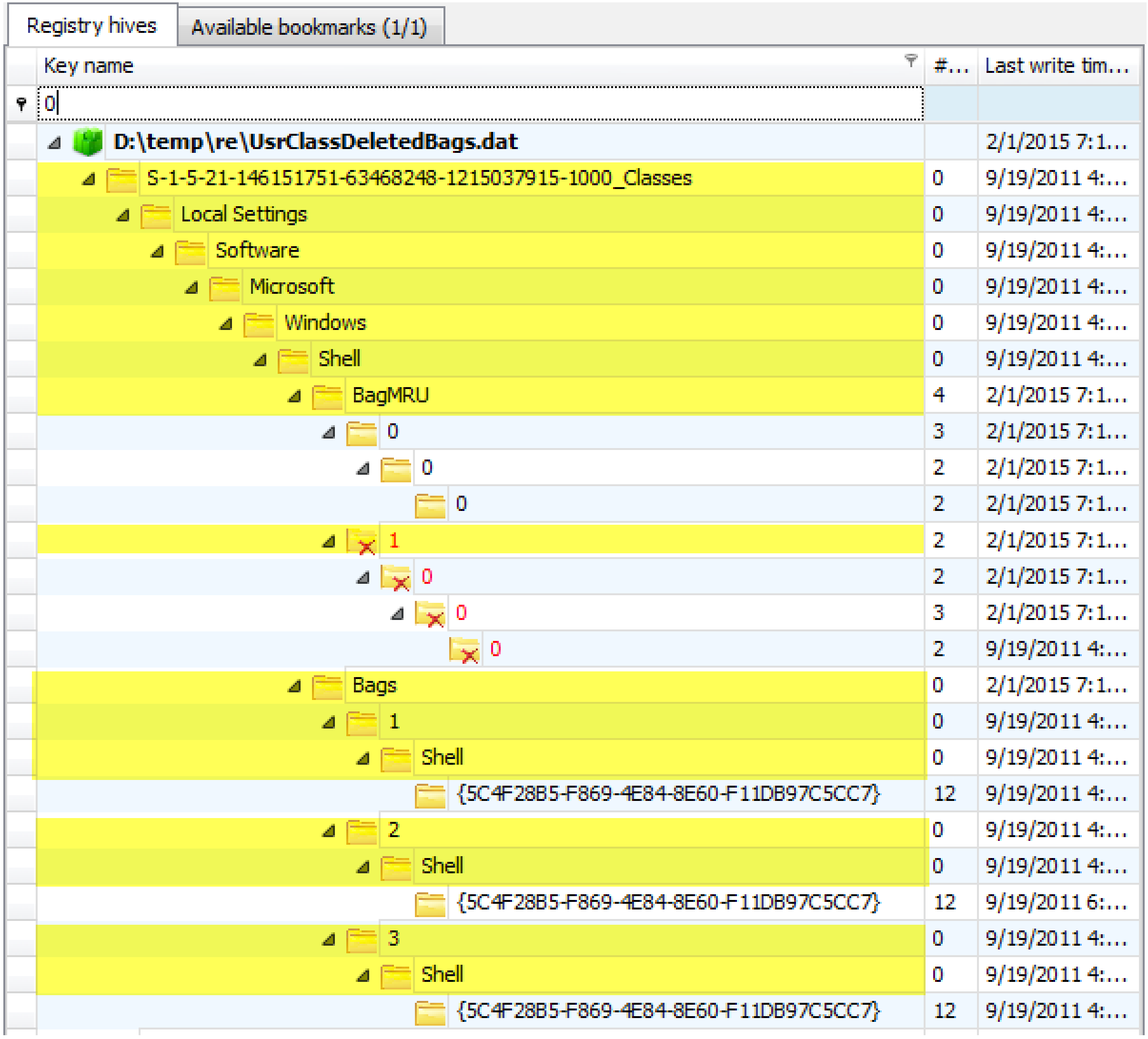
Options

This menu contains several options that control such things as recovering deleted records, viewing hidden keys, etc.


Registry Explorer v0.0.4.3
File
Tools
Options
Bookmarks (1/1)
View
Help
Registry hive
Key name
Recover deleted keys/values
Show associated deleted records
Show unassociated deleted records
Show parent keys when filtering
Show hidden keys
Manage hidden keys Ctrl+H
Skins

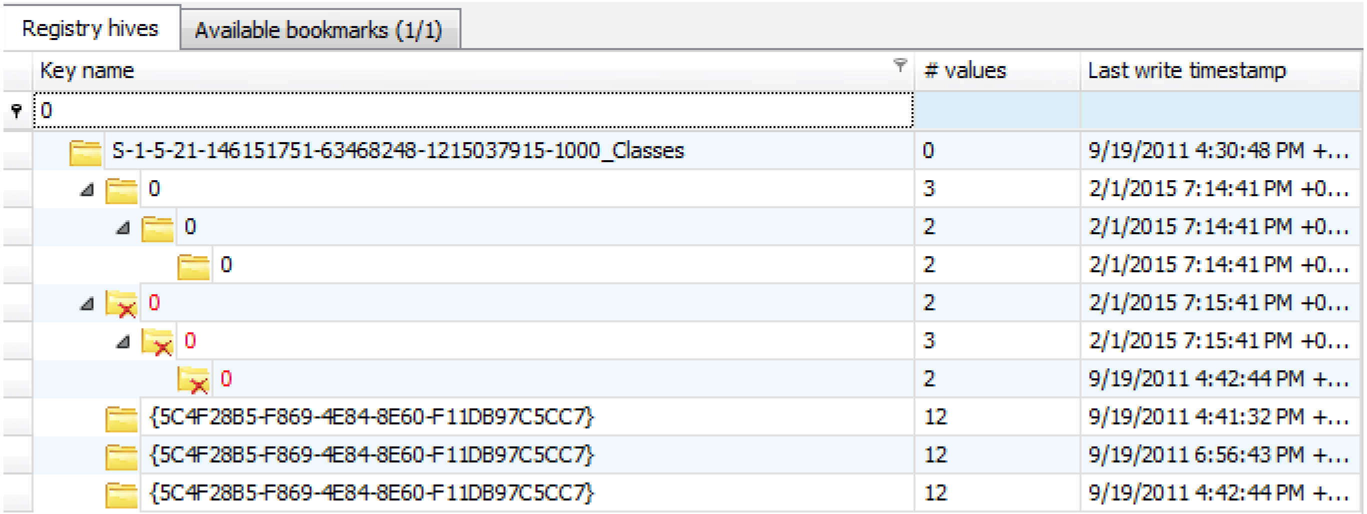
- **Recover deleted keys/values**: When enabled (the selector is to the right), Registry Explorer will recover any deleted records available during hive loading
- **Show associated deleted records**: When enabled, all associated deleted records will be shown in a special group under the main Registry hive data. Any recovered keys that could be associated with an active (that is, not deleted) key will also be shown in relation to the active key. This will be explained more in a subsequent section.
- **Show unassociated deleted records**: Similar to the previous option, but this group contains all of the keys that could not be associated with an active key.
- **Show parent keys when filtering**: This option changes the way the Registry hives tree works when using the column filters. When this option is enabled, any keys that match a filter will be displayed, along with the parent keys that the matching key belongs to as seen in the screen shot below.

  The keys highlighted in yellow are parent keys that may not contain the text entered in the filter column.



Key name	#...	Last write tim...
0		
D:\temp\re\UsrClassDeletedBags.dat		2/1/2015 7:1...
S-1-5-21-146151751-63468248-1215037915-1000_Classes	0	9/19/2011 4:...
Local Settings	0	9/19/2011 4:...
Software	0	9/19/2011 4:...
Microsoft	0	9/19/2011 4:...
Windows	0	9/19/2011 4:...
Shell	0	9/19/2011 4:...
BagMRU	4	2/1/2015 7:1...
0	3	2/1/2015 7:1...
0	2	2/1/2015 7:1...
0	2	2/1/2015 7:1...
1	2	2/1/2015 7:1...
0	2	2/1/2015 7:1...
0	3	2/1/2015 7:1...
0	2	9/19/2011 4:...
Bags	0	2/1/2015 7:1...
1	0	9/19/2011 4:...
Shell	0	9/19/2011 4:...
{5C4F28B5-F869-4E84-8E60-F11DB97C5CC7}	12	9/19/2011 4:...
2	0	9/19/2011 4:...
Shell	0	9/19/2011 4:...
{5C4F28B5-F869-4E84-8E60-F11DB97C5CC7}	12	9/19/2011 6:...
3	0	9/19/2011 4:...
Shell	0	9/19/2011 4:...
{5C4F28B5-F869-4E84-8E60-F11DB97C5CC7}	12	9/19/2011 4:...

If we turn off this option, we get a much different result as seen below.

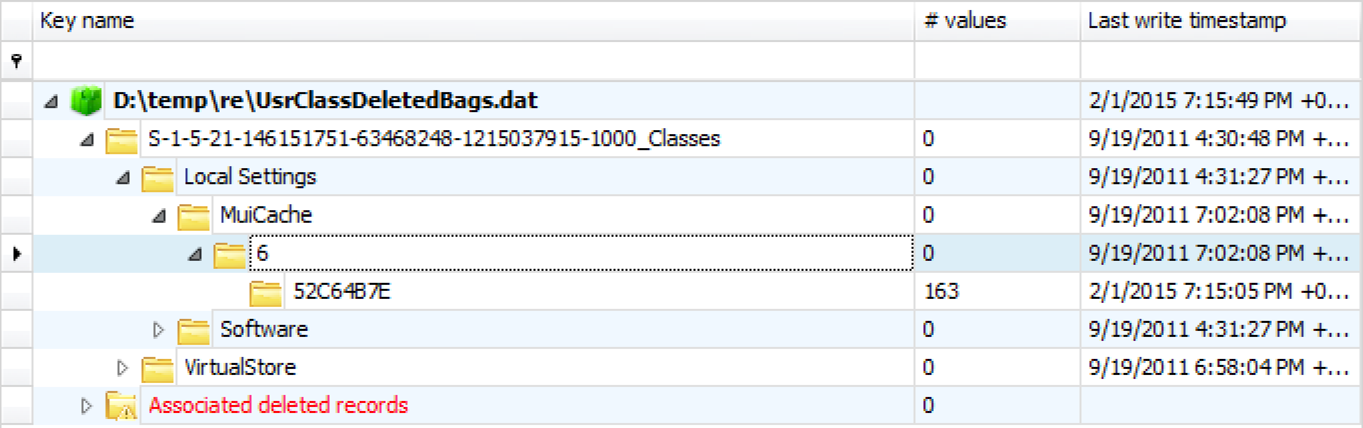


Key name	# values	Last write timestamp
0		
S-1-5-21-146151751-63468248-1215037915-1000_Classes	0	9/19/2011 4:30:48 PM +...
0	3	2/1/2015 7:14:41 PM +0...
0	2	2/1/2015 7:14:41 PM +0...
0	2	2/1/2015 7:14:41 PM +0...
0	2	2/1/2015 7:15:41 PM +0...
0	3	2/1/2015 7:15:41 PM +0...
0	2	9/19/2011 4:42:44 PM +...
{5C4F28B5-F869-4E84-8E60-F11DB97C5CC7}	12	9/19/2011 4:41:32 PM +...
{5C4F28B5-F869-4E84-8E60-F11DB97C5CC7}	12	9/19/2011 6:56:43 PM +...
{5C4F28B5-F869-4E84-8E60-F11DB97C5CC7}	12	9/19/2011 4:42:44 PM +...

Notice in this screen shot only the keys that match the filter criteria are shown. This can greatly reduce noise in the results in addition to lessening the need to scroll to the keys that match the filter criteria.

The other aspect this option controls is whether to show subkeys of keys that match a given filter. With the option on, any subkeys of keys that match the filter will continue to be shown. With the option off, subkeys of keys matching the filter are hidden.

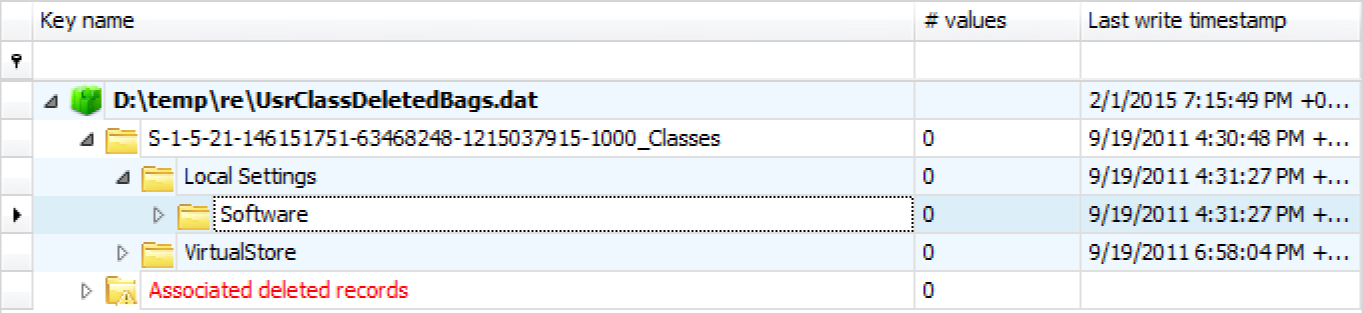
- **Show hidden keys:** When enabled, any keys that have been hidden will be shown in the Registry hives tree. In the screen shot below, several keys are shown.



Key name	# values	Last write timestamp
D:\temp\re\UsrClassDeletedBags.dat		2/1/2015 7:15:49 PM +0...
S-1-5-21-146151751-63468248-1215037915-1000_Classes	0	9/19/2011 4:30:48 PM +...
Local Settings	0	9/19/2011 4:31:27 PM +...
MuiCache	0	9/19/2011 7:02:08 PM +...
6	0	9/19/2011 7:02:08 PM +...
52C64B7E	163	2/1/2015 7:15:05 PM +0...
Software	0	9/19/2011 4:31:27 PM +...
VirtualStore	0	9/19/2011 6:58:04 PM +...
Associated deleted records	0	

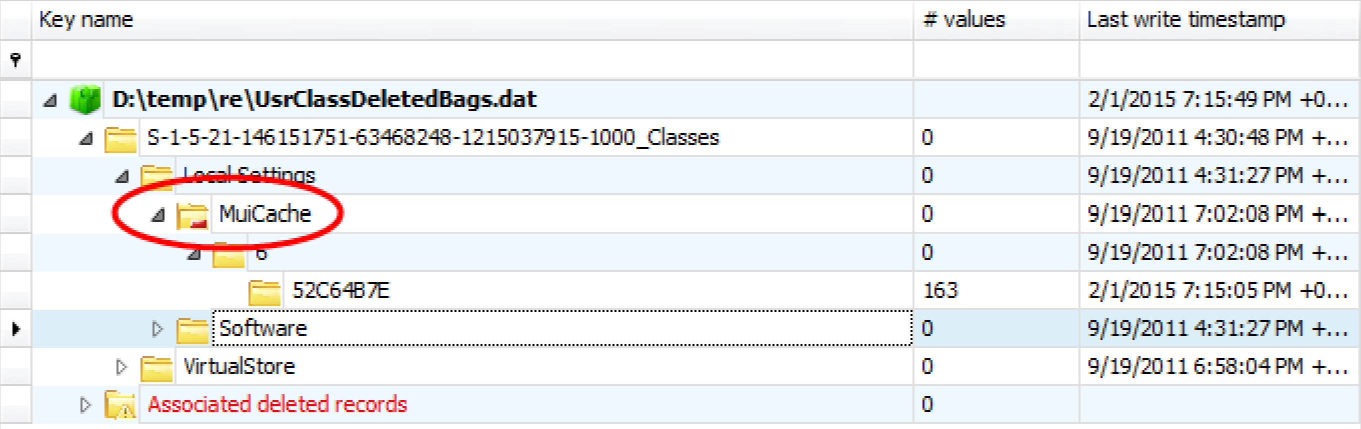
While we haven't discussed how to hide keys yet (it has its own section below), if we right click on a key, an option to hide the selected key (based on the key path, not just the key name) is shown.

For example, if we hide the MuiCache key, it will disappear from view, as seen below.



Key name	# values	Last write timestamp
D:\temp\re\UsrClassDeletedBags.dat		2/1/2015 7:15:49 PM +0...
S-1-5-21-146151751-63468248-1215037915-1000_Classes	0	9/19/2011 4:30:48 PM +...
Local Settings	0	9/19/2011 4:31:27 PM +...
Software	0	9/19/2011 4:31:27 PM +...
VirtualStore	0	9/19/2011 6:58:04 PM +...
Associated deleted records	0	

Notice the MuiCache key is no longer visible (assuming the Show hidden key option is off). If we enable this option, the MuiCache key will be shown in its original place, but the icon is different to show that it is in fact a hidden key.



Key name	# values	Last write timestamp
D:\temp\re\UsrClassDeletedBags.dat		2/1/2015 7:15:49 PM +0...
S-1-5-21-146151751-63468248-1215037915-1000_Classes	0	9/19/2011 4:30:48 PM +...
Local Settings	0	9/19/2011 4:31:27 PM +...
MuiCache	0	9/19/2011 7:02:08 PM +...
6	0	9/19/2011 7:02:08 PM +...
52C64B7E	163	2/1/2015 7:15:05 PM +0...
Software	0	9/19/2011 4:31:27 PM +...
VirtualStore	0	9/19/2011 6:58:04 PM +...
Associated deleted records	0	

When a key is hidden, the lower right corner will have a red dash to indicate this.

- **Manage hidden keys:** Brings up an interface to remove keys from the auto hide list. There are two options available when hiding keys: hide for session, and hide and add to auto hide. The Manage hidden keys interface only displays key paths that have previously been added to the auto hide list. Any key paths removed from the auto hide list will be unhidden when the Manage hidden keys interface is closed. Additional ways to unhide keys will also be discussed in a subsequent section.
- **Skins:** Allows for selecting a skin or theme that Registry Explorer will use.

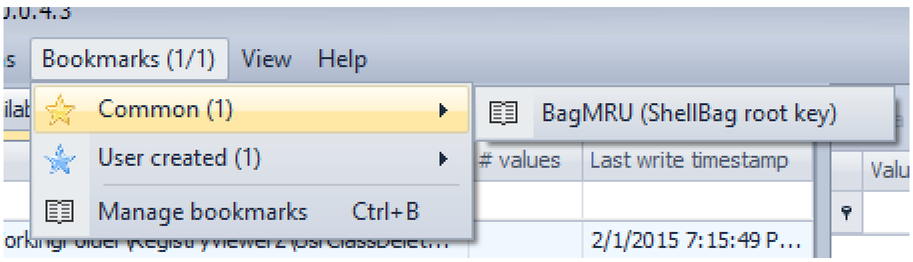
Bookmarks

The Bookmarks menu contains both common (included with Registry Explorer) and user created bookmarks to “of interest” Registry keys. Bookmarks can be created for any Registry key (we will see how to create our own bookmarks soon). Bookmarks that are included with Registry Explorer will show up under the ‘Common’ menu and any user created bookmarks will appear under the ‘User created’ menu.

Bookmarks live in a subdirectory of the main Registry Explorer program directory in a directory named Bookmarks. The Bookmarks directory contains two subdirectories, Common and User. To move a user created bookmark from the User created to Common submenu, simply move the bookmark file from the User directory to the Common directory.

The Manage bookmarks interface can be used to edit or delete bookmarks. Additionally, simply deleting the bookmark file from the Common or User directory will also remove the bookmark.

Bookmarks are simple json files and can also be edited with any text editor. Since they are simple json files, exchanging a good set of bookmarks with other users is as easy as sending someone else the bookmark files from the User directory. There is a project on Github, found here, that you can push your Bookmarks to.



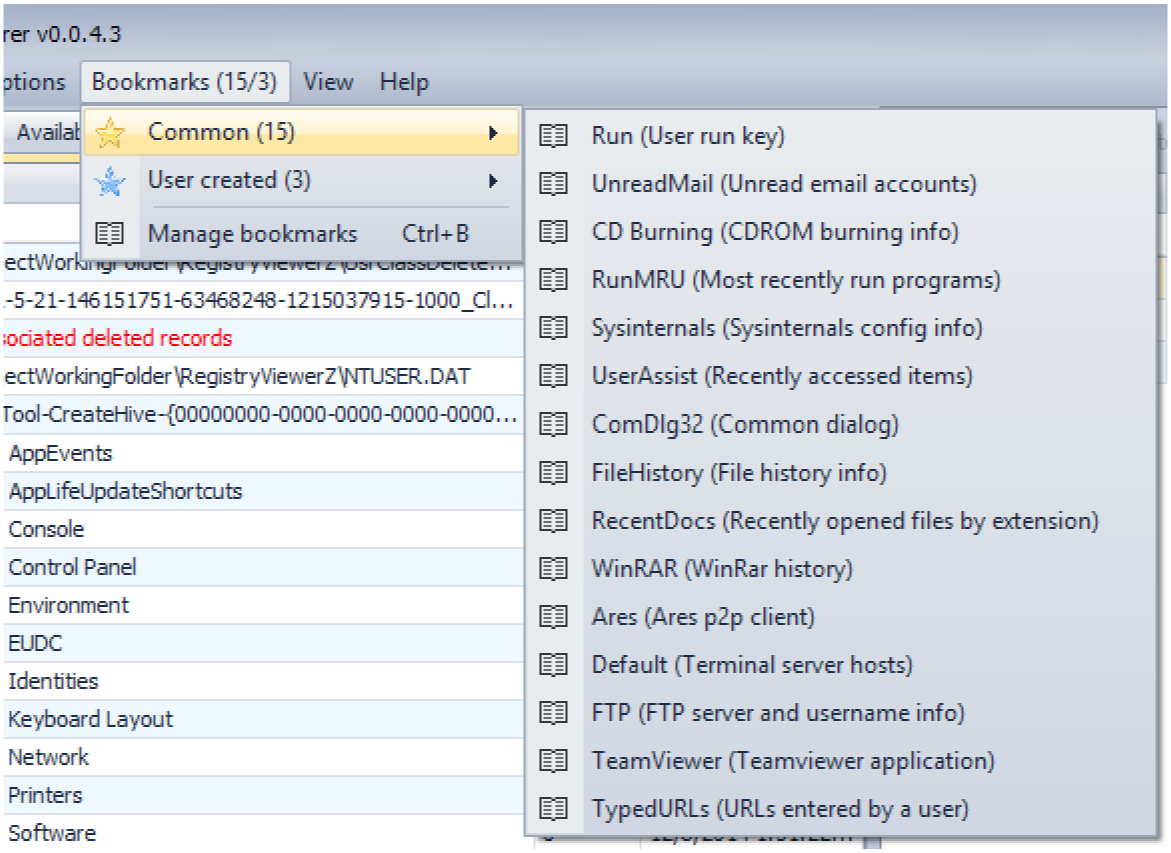
The main Bookmarks menu contains two numbers at the end. The first number is the total number of Common bookmarks ***that exist in the selected hive*** and the last number is the number of User bookmarks **that exist in the selected hive**. Clicking on any of the bookmarks will cause Registry Explorer to jump to the bookmarked key.

Bookmarks are tied to a Registry hive type and a key path within that hive type. When we discuss creating bookmarks below this will become clearer, but for now remember that each bookmark is associated with a certain flavor of Registry hive (NTUSER, UsrClass, SYSTEM, etc).

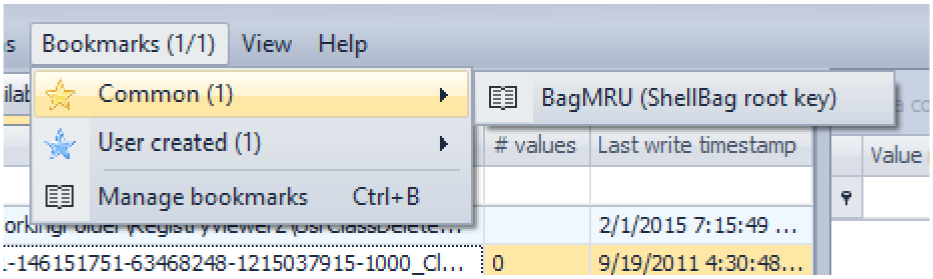
The Bookmarks menus dynamically adjust as hives are loaded and selected. For example, suppose you have the following bookmarks by hive type:

- NTUSER.DAT: 20 bookmarks
- USRCLASS.DAT: 8 bookmarks

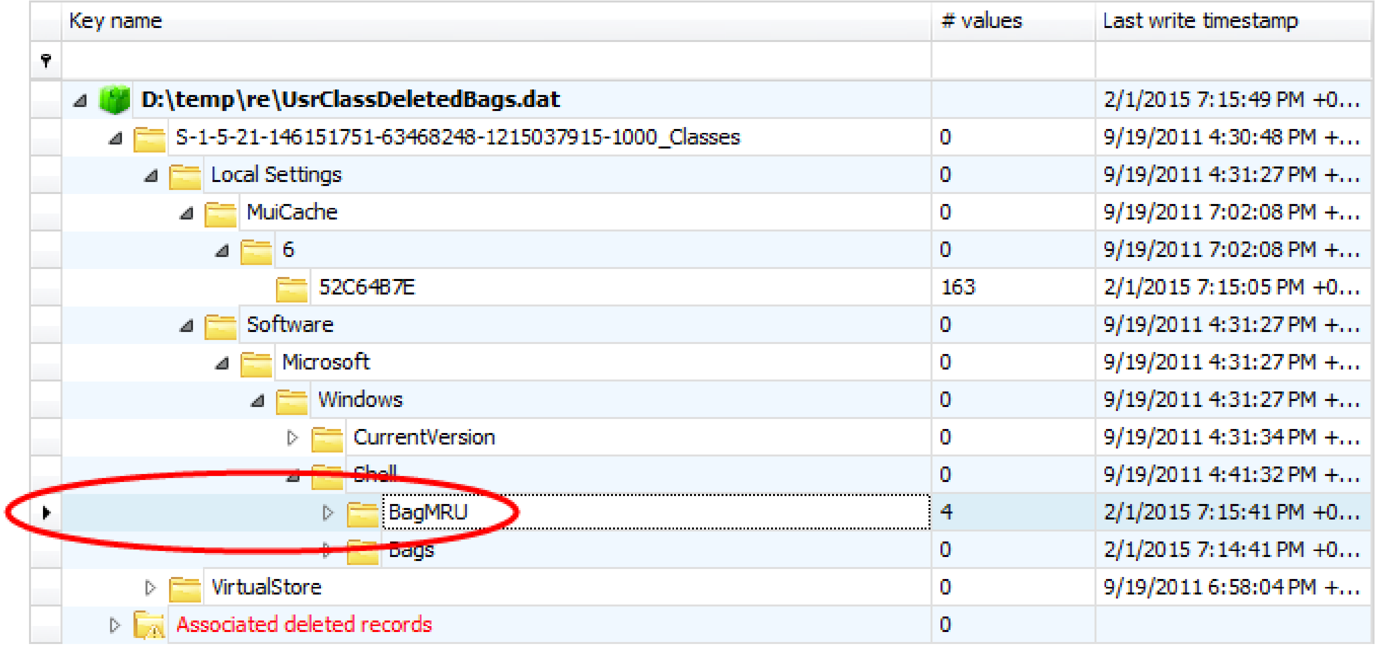
You then load an NTUSER and USRCLASS hive. The NTUSER hive contains 18 out of the 20 key paths as defined in the NTUSER.DAT related bookmarks (15 from common and 3 from user created). The USRCLASS hive contains two out of the eight bookmarks (one from common and one from user created). If you click on anything in the NTUSER.DAT hive, the Bookmarks menu will change to show you *only the bookmarks that actually exist* in the NTUSER hive, like this.


rer v0.0.4.3
ptions
Bookmarks (15/3)
View
Help
Availat
Common (15)
User created (3)
Manage bookmarks Ctrl+B
Run (User run key)
UnreadMail (Unread email accounts)
CD Burning (CDROM burning info)
RunMRU (Most recently run programs)
Sysinternals (Sysinternals config info)
UserAssist (Recently accessed items)
ComDlg32 (Common dialog)
FileHistory (File history info)
RecentDocs (Recently opened files by extension)
WinRAR (WinRar history)
Ares (Ares p2p client)
Default (Terminal server hosts)
FTP (FTP server and username info)
TeamViewer (Teamviewer application)
TypedURLs (URLs entered by a user)
-5-21-146151751-63468248-1215037915-1000_Cl...
sociated deleted records
ectWorkingFolder\RegistryViewerZ\NTUSER.DAT
Tool-CreateHive-{00000000-0000-0000-0000-0000...
AppEvents
AppLifeUpdateShortcuts
Console
Control Panel
Environment
EUDC
Identities
Keyboard Layout
Network
Printers
Software

If you then click on the USRCLASS hive, the Bookmarks menu will again dynamically adjust to show what is available in the USRCLASS hive.



Again, clicking a bookmark will jump to the key as defined in the bookmark. For example, clicking on the BagMRU bookmark results in the following key being selected (and of course all parent keys will be expanded so the bookmarked key is visible).



Key name	# values	Last write timestamp
D:\temp\re\UsrClassDeletedBags.dat		2/1/2015 7:15:49 PM +0...
S-1-5-21-146151751-63468248-1215037915-1000_Classes	0	9/19/2011 4:30:48 PM +...
Local Settings	0	9/19/2011 4:31:27 PM +...
MuiCache	0	9/19/2011 7:02:08 PM +...
6	0	9/19/2011 7:02:08 PM +...
52C64B7E	163	2/1/2015 7:15:05 PM +0...
Software	0	9/19/2011 4:31:27 PM +...
Microsoft	0	9/19/2011 4:31:27 PM +...
Windows	0	9/19/2011 4:31:27 PM +...
CurrentVersion	0	9/19/2011 4:31:34 PM +...
Shell	0	9/19/2011 4:41:32 PM +...
BagMRU	4	2/1/2015 7:15:41 PM +0...
Bags	0	2/1/2015 7:14:41 PM +0...
VirtualStore	0	9/19/2011 6:58:04 PM +...
Associated deleted records	0	

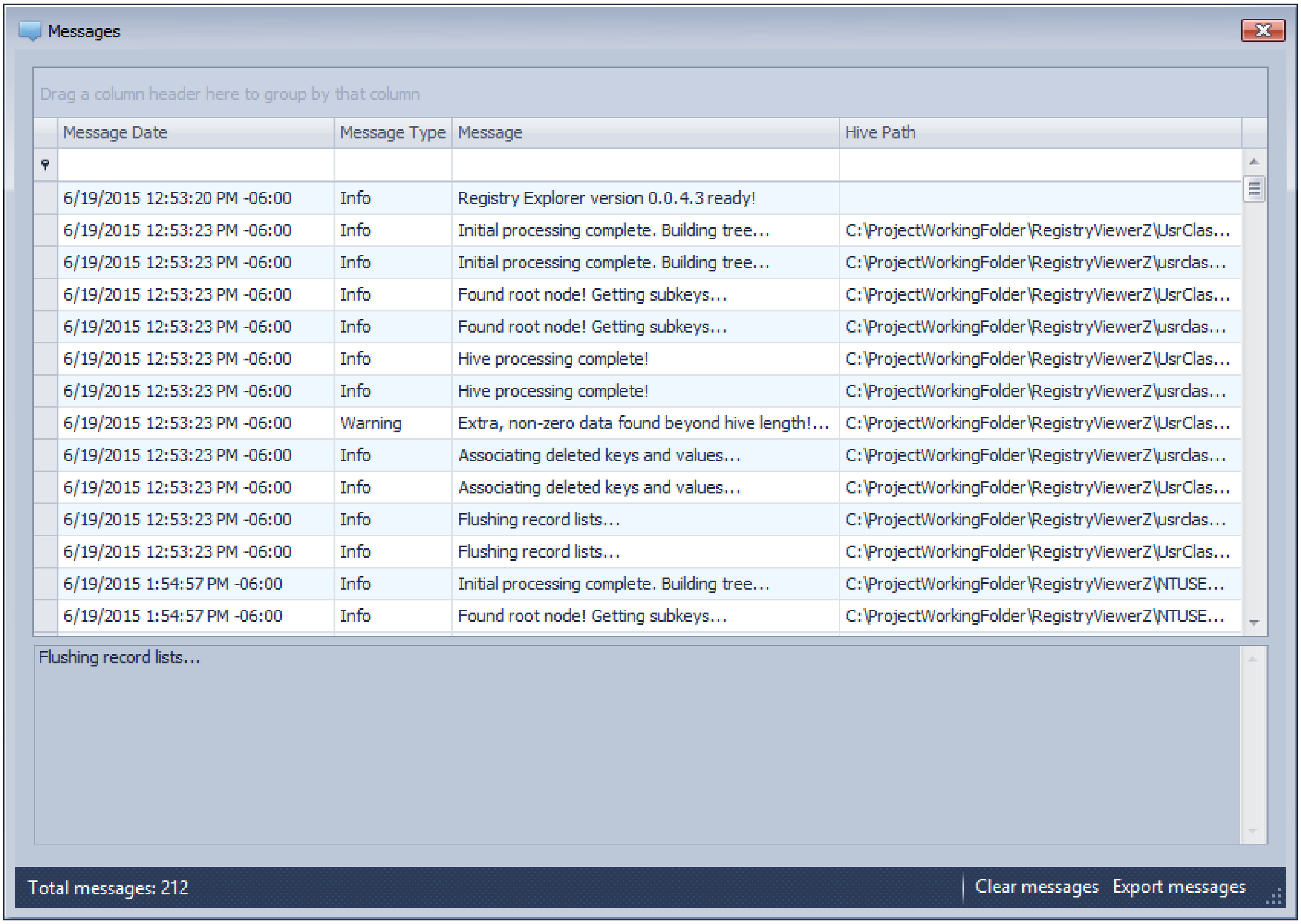
Because the Bookmarks menu dynamically adjusts itself based solely on what exists in the active hive, you do not have to click on bookmarks before you know whether they exist. This is a huge time saver and makes drilling down into hives much easier.

As you interact with loaded hives, the Bookmarks menu will show you at a glance how many bookmarks are available, but as we will soon see, Registry Explorer has an even easier way to interact with bookmarks (the Available bookmarks tab).

View

The View menu contains a single option, Messages, that toggles visibility of the Messages window. The Messages window displays status messages and other feedback as hives are loaded and so on.

Hives tend to process and load faster when the Messages window is hidden, so keep that in mind when loading many hives at once or when processing large hives.

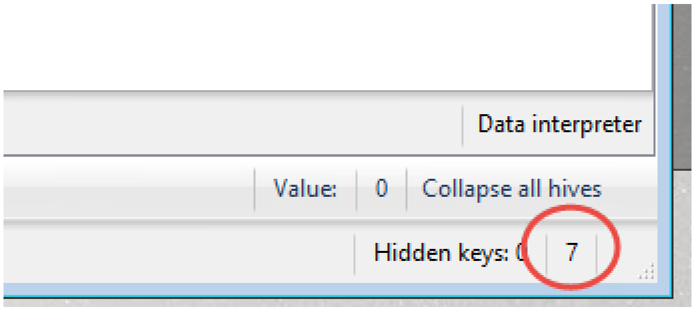


Message Date	Message Type	Message	Hive Path
6/19/2015 12:53:20 PM -06:00	Info	Registry Explorer version 0.0.4.3 ready!	
6/19/2015 12:53:23 PM -06:00	Info	Initial processing complete. Building tree...	C:\ProjectWorkingFolder\RegistryViewerZ\UsrClas...
6/19/2015 12:53:23 PM -06:00	Info	Initial processing complete. Building tree...	C:\ProjectWorkingFolder\RegistryViewerZ\usrclas...
6/19/2015 12:53:23 PM -06:00	Info	Found root node! Getting subkeys...	C:\ProjectWorkingFolder\RegistryViewerZ\UsrClas...
6/19/2015 12:53:23 PM -06:00	Info	Found root node! Getting subkeys...	C:\ProjectWorkingFolder\RegistryViewerZ\usrclas...
6/19/2015 12:53:23 PM -06:00	Info	Hive processing complete!	C:\ProjectWorkingFolder\RegistryViewerZ\UsrClas...
6/19/2015 12:53:23 PM -06:00	Info	Hive processing complete!	C:\ProjectWorkingFolder\RegistryViewerZ\usrclas...
6/19/2015 12:53:23 PM -06:00	Warning	Extra, non-zero data found beyond hive length!...	C:\ProjectWorkingFolder\RegistryViewerZ\UsrClas...
6/19/2015 12:53:23 PM -06:00	Info	Associating deleted keys and values...	C:\ProjectWorkingFolder\RegistryViewerZ\usrclas...
6/19/2015 12:53:23 PM -06:00	Info	Associating deleted keys and values...	C:\ProjectWorkingFolder\RegistryViewerZ\UsrClas...
6/19/2015 12:53:23 PM -06:00	Info	Flushing record lists...	C:\ProjectWorkingFolder\RegistryViewerZ\usrclas...
6/19/2015 12:53:23 PM -06:00	Info	Flushing record lists...	C:\ProjectWorkingFolder\RegistryViewerZ\UsrClas...
6/19/2015 1:54:57 PM -06:00	Info	Initial processing complete. Building tree...	C:\ProjectWorkingFolder\RegistryViewerZ\NTUSE...
6/19/2015 1:54:57 PM -06:00	Info	Found root node! Getting subkeys...	C:\ProjectWorkingFolder\RegistryViewerZ\NTUSE...

Flushing record lists...

Total messages: 212

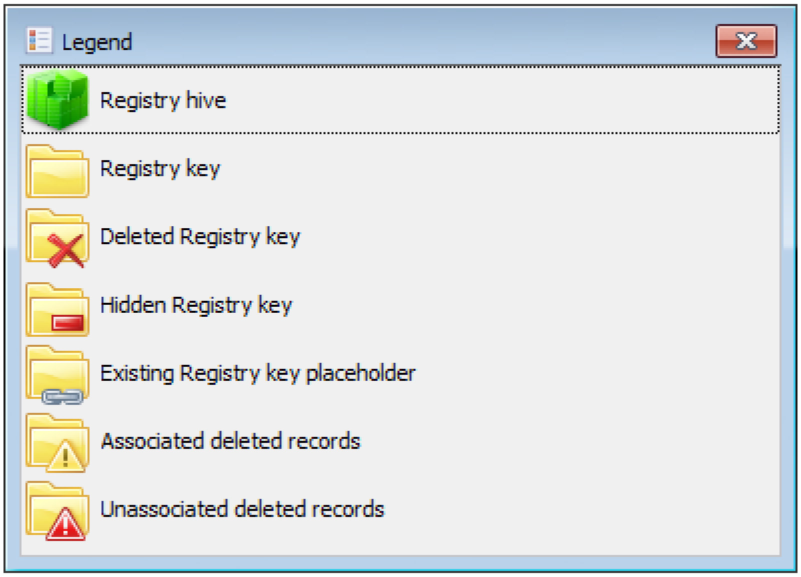
The total number of messages is also shown on the main window’s bottom status bar to the far right. Double clicking the message count will show the Messages form.



Help

The Help menu contains three options: Quick help, Legend, and About. The Legend shows the various icons seen in the Registry hives tree and a description about them.

The legend contains descriptions for the different icons used for various Registry objects such as hives, keys, and existing key placeholders. The legend can be seen below.



Registry hive

Registry key

Deleted Registry key

Hidden Registry key

Existing Registry key placeholder

Associated deleted records

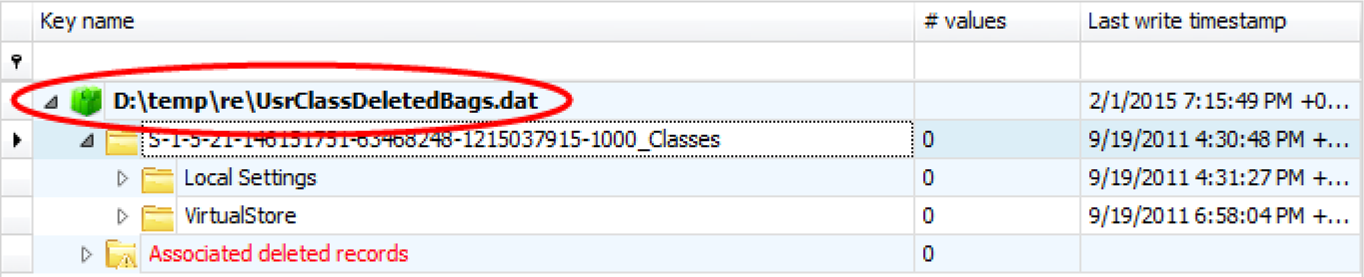
Unassociated deleted records

Using Registry Explorer

Loading hives

To load hives into Registry Explorer, either select one or more hives and drag/drop them onto the main interface. You can also use **File | Load offline hive** or press **Alt+1** to select hives.

After selecting a hive, Registry Explorer will fully process the hive. Once that is done, the hive will be displayed on the main interface. The top level node for a hive is the full file path to the hive as seen below. The hive node has a green icon and is also in **bold** to differentiate it from keys.



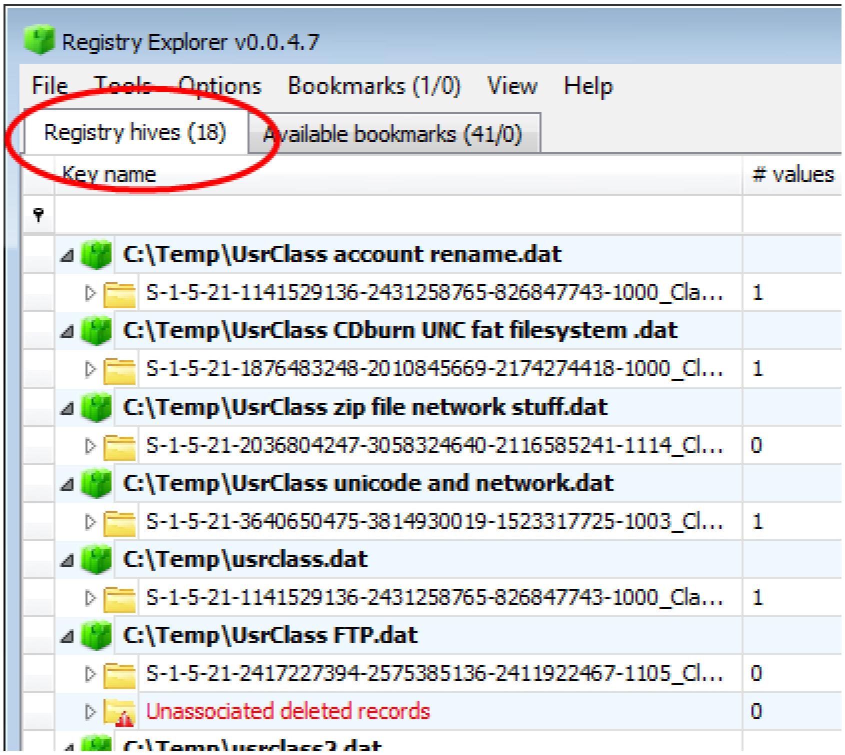
Key name	# values	Last write timestamp
D:\temp\re\UsrClassDeletedBags.dat		2/1/2015 7:15:49 PM +0...
S-1-5-21-146151751-63468248-1215037915-1000_Classes	0	9/19/2011 4:30:48 PM +...
Local Settings	0	9/19/2011 4:31:27 PM +...
VirtualStore	0	9/19/2011 6:58:04 PM +...
Associated deleted records	0	

The last write timestamp for the hive is the timestamp value from the header of the hive.

Below the hive name is the root key for the hive. The root key name can vary for every hive that is loaded. All other keys in the active (that is, not deleted) portion of the Registry will be displayed under the root key.

If the option to recover deleted records is enabled, up to two different virtual keys may be created: one for Associated deleted records and one for unassociated deleted records. These virtual keys will not be shown if there aren't any deleted records of that type available. As discussed above, these keys can also be hidden using the relevant option under the Options menu.

The number after Registry hives in parenthesis is the total number of hives loaded. In the example below, there are 18 hives loaded.



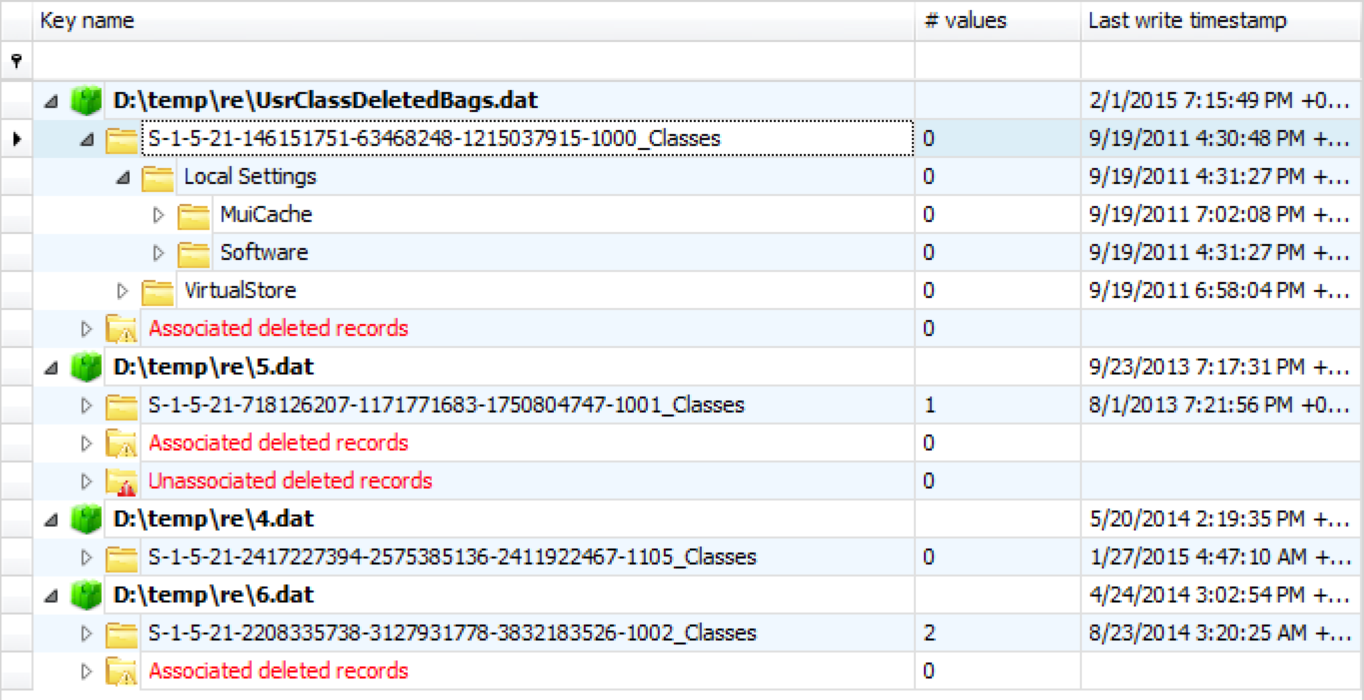
Key name	# values
C:\Temp\UsrClass account rename.dat	
S-1-5-21-1141529136-2431258765-826847743-1000_Cla…	1
C:\Temp\UsrClass CDburn UNC fat filesystem .dat	
S-1-5-21-1876483248-2010845669-2174274418-1000_Cl…	1
C:\Temp\UsrClass zip file network stuff.dat	
S-1-5-21-2036804247-3058324640-2116585241-1114_Cl…	0
C:\Temp\UsrClass unicode and network.dat	
S-1-5-21-3640650475-3814930019-1523317725-1003_Cl…	1
C:\Temp\usrclass.dat	
S-1-5-21-1141529136-2431258765-826847743-1000_Cla…	1
C:\Temp\UsrClass FTP.dat	
S-1-5-21-2417227394-2575385136-2411922467-1105_Cl…	0
Unassociated deleted records	0
C:\Temp\usrclass2.dat	

Selecting keys

Selecting keys in Registry Explorer works much the same as it does in regedit or selecting directories in Windows Explorer. Clicking the small arrow to the left of the key name or double clicking a key will expand that key, displaying any subkeys that are present. If the arrow is not visible, the key does not have any subkeys.

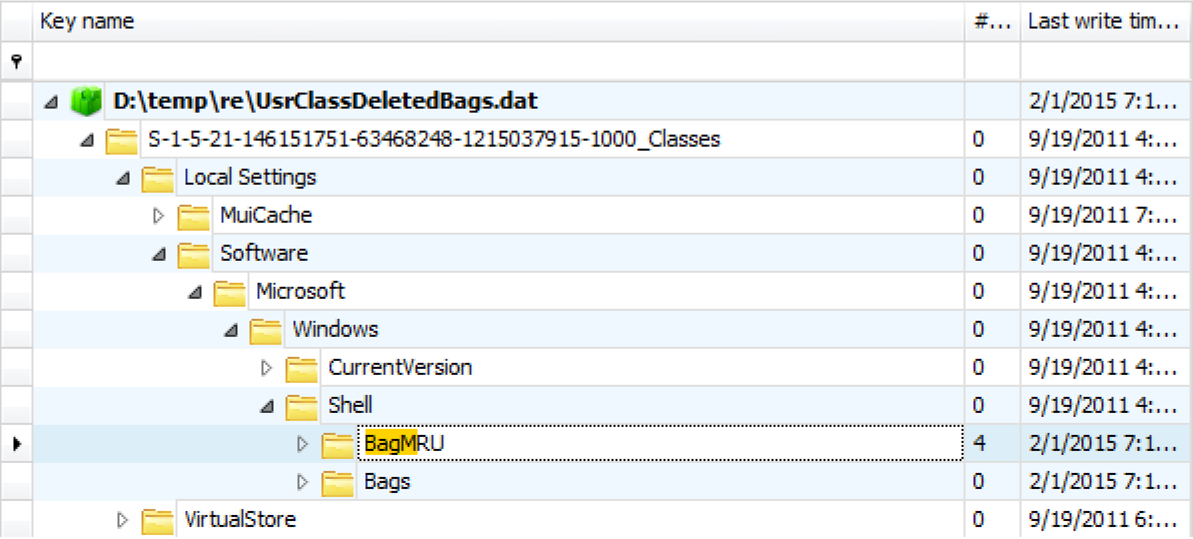
Keys can be double clicked and expanded, drilling down into the key hierarchy, until the key you are interested in is located. Alternatively, you can simply start typing a key's name and the keys will be dynamically expanded as matching keys are found in the tree.

For example, assume Registry Explorer looks like this:



Key name	# values	Last write timestamp
D:\temp\re\UsrClassDeletedBags.dat		2/1/2015 7:15:49 PM +0...
S-1-5-21-146151751-63468248-1215037915-1000_Classes	0	9/19/2011 4:30:48 PM +...
Local Settings	0	9/19/2011 4:31:27 PM +...
MuiCache	0	9/19/2011 7:02:08 PM +...
Software	0	9/19/2011 4:31:27 PM +...
VirtualStore	0	9/19/2011 6:58:04 PM +...
Associated deleted records	0	
D:\temp\re\5.dat		9/23/2013 7:17:31 PM +...
S-1-5-21-718126207-1171771683-1750804747-1001_Classes	1	8/1/2013 7:21:56 PM +0...
Associated deleted records	0	
Unassociated deleted records	0	
D:\temp\re\4.dat		5/20/2014 2:19:35 PM +...
S-1-5-21-2417227394-2575385136-2411922467-1105_Classes	0	1/27/2015 4:47:10 AM +...
D:\temp\re\6.dat		4/24/2014 3:02:54 PM +...
S-1-5-21-2208335738-3127931778-3832183526-1002_Classes	2	8/23/2014 3:20:25 AM +...
Associated deleted records	0	

If you want to look at the contents of the BagMRU key, click on either the hive path or the root key, then start typing *BagMRU*. As each letter is typed, Registry Explorer will search for matching keys and select them. After a few keystrokes, the following key is selected.



Key name	#...	Last write tim...
D:\temp\re\UsrClassDeletedBags.dat		2/1/2015 7:1...
S-1-5-21-146151751-63468248-1215037915-1000_Classes	0	9/19/2011 4:...
Local Settings	0	9/19/2011 4:...
MuiCache	0	9/19/2011 7:...
Software	0	9/19/2011 4:...
Microsoft	0	9/19/2011 4:...
Windows	0	9/19/2011 4:...
CurrentVersion	0	9/19/2011 4:...
Shell	0	9/19/2011 4:...
BagMRU	4	2/1/2015 7:1...
Bags	0	2/1/2015 7:1...
VirtualStore	0	9/19/2011 6:...

Notice also the part of the key that matched what was typed is highlighted. While a bookmark can be used to quickly jump to a particular key, using this technique can save a lot of time when you know the name of the key you are interested in.

Filtering keys

The top of the Registry hives tree contains areas to enter text to filter that column. One thing to note is that *only expanded keys are included in the filter results*. To filter against all keys in a hive, use the context menu option to expand all subkeys (or press **Alt+Down**) before filtering. The next section will cover the context menu in detail.

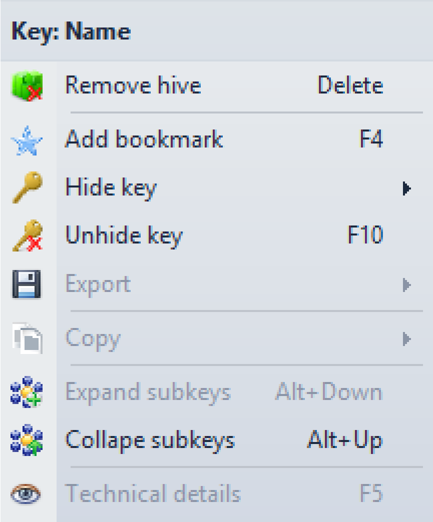
The **Options | Show parent keys when filtering** option affects what is shown when filtering keys. See the Options section for a full discussion on how this option works.

While it may seem that filtering is the quickest way to find a certain key, it is quite often faster to type the name of a key you are interested in (or better yet, using **Tools | Find** if it exists in more than one place).

Key context menu

As in other places, the context menu changes dynamically depending on what you right click on. For example, if you right click on a hive's full path, you will see the option to remove the hive from Registry Explorer. Right clicking anywhere else but the hive's path will hide this option from view. Similarly, if a key is hidden, an option to unhide the key will be shown, else it will be hidden, and so on.

The key context menu looks similar to this:



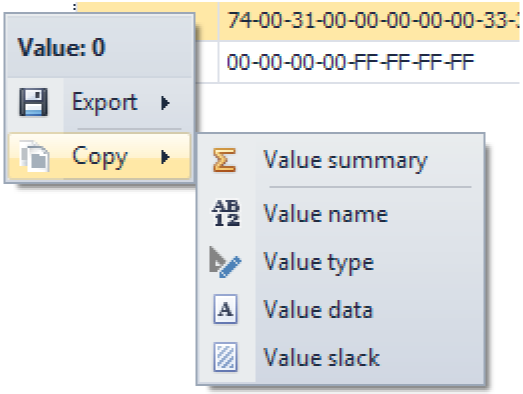
The name of the currently selected key is shown at the top. Most options also have shortcuts which can be used in lieu of using the mouse.

- **Remove hive**: Removes the loaded hive from Registry Explorer. This option is only shown when a hive is selected (denoted by the full path to the hive name, shown in **bold**, and with a different icon).
- **Add bookmark**: Creates a new user bookmark. Full details will be discussed below.
- Hide key
    - **For this session only**: Hides keys matching the selected key's path from all loaded hives until Registry Explorer is restarted
    - **Hide and add to auto hide**: Same as the above option, except the key's path is remembered between restarts of Registry Explorer. This option is useful to hide non-useful keys in the Registry that get in your way.
- **Unhide key**: Unhide previously hidden keys with the same path as the selected key. If a key has been auto hidden, this option will remove it from the auto hide list.
- **Export**
    - **To .reg format**: Exports the selected key and its values to plaintext format. This file can then be imported into the active Registry by double clicking on the generated file.

  - **To .reg format recursively**: The same as above, except all keys and values for the selected key and all subkeys are exported.
- Copy
  - **Key name**: Copies the selected key's key name to the clipboard. Double clicking the key path in the status bar while holding **Shift** also copies the key name to the clipboard.
  - **Key path**: Copies the selected key's key path to the clipboard. Double clicking the key path in the status bar also copies the key path to the clipboard.
  - **Last write time**: Copies the selected key's last write timestamp to the clipboard. Double clicking the last write timestamp in the status bar also copies the last write timestamp to the clipboard. Double clicking the status bar while holding **Shift** will copy the key name and last write timestamp to the clipboard.
- **Expand subkeys**: Recursively expands the selected key and all subkeys
- **Collapse subkeys**: Collapse all subkeys below the selected key
- **Technical details**: Displays full technical details about the selected key, its subkeys, values, security records, and hive header. This option will be fully explored below.

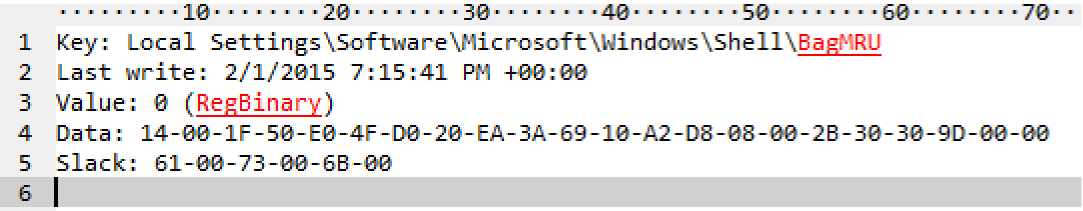
Value context menu

A typical value context menu may look like this:



The name of the currently selected value is shown at the top.

- Export
  - **Value data**: Exports selected value's data in *binary form* to a file
  - **Value slack**: Exports selected value's slack data in binary form to a file. If a value has no slack, this option is disabled
- Copy
  - **Value summary**: Copies a summary of the selected value to the clipboard. An example is shown below.



```
Key: Local Settings\Software\Microsoft\Windows\Shell\BagMRU
Last write: 2/1/2015 7:15:41 PM +00:00
Value: 0 (RegBinary)
Data: 14-00-1F-50-E0-4F-D0-20-EA-3A-69-10-A2-D8-08-00-2B-30-30-9D-00-00
Slack: 61-00-73-00-6B-00
```

  - **Value name**: Copies the selected value's name to the clipboard

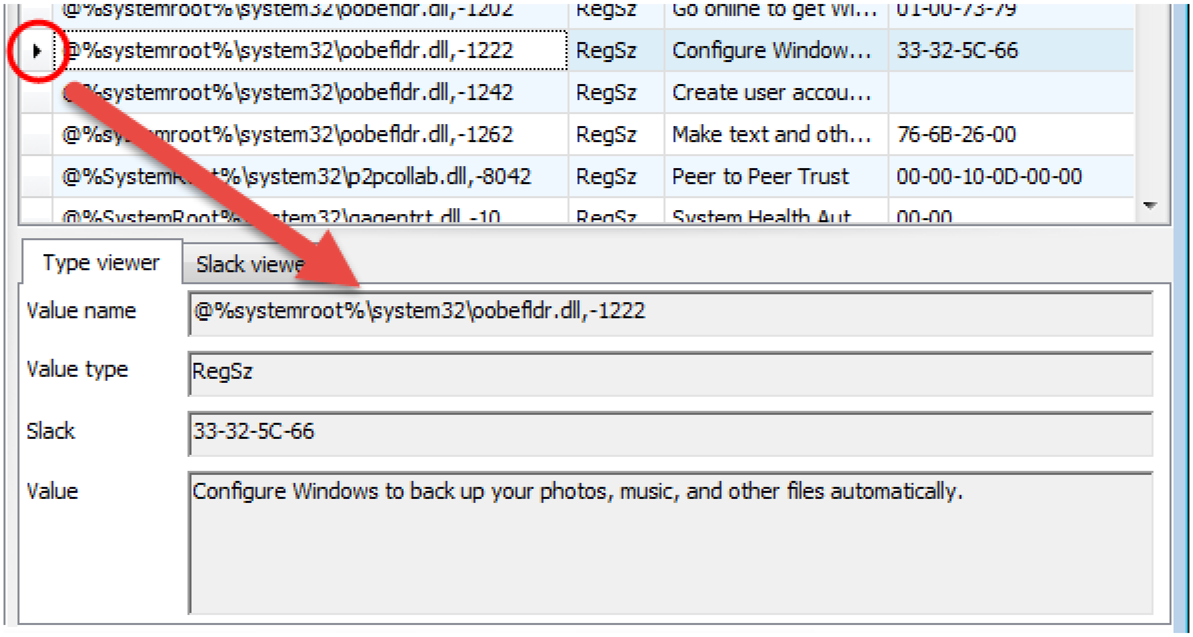
- **Value type**: Copies the selected value's type to the clipboard (RegBinary or RegSz for example)
- **Value data**: Copies the selected value's value data to the clipboard. For RegBinary values, the hex values, separated by a hyphen, are copied to the clipboard as a string
- **Value slack**: Copies the selected value's value slack to the clipboard. This option formats the data the same as the Value data option. If a value has no slack, this option is disabled.

Value details

The Value details area will change depending on the type of value selected.

Type viewer

For all values except RegBinary values, a simple string representation of the value is shown as seen below.



@%systemroot%\system32\oobefldr.dll,-1202	RegSz	Go online to get Wi…	01-00-73-79
@%systemroot%\system32\oobefldr.dll,-1222	RegSz	Configure Window…	33-32-5C-66
@%systemroot%\system32\oobefldr.dll,-1242	RegSz	Create user accou…	
@%systemroot%\system32\oobefldr.dll,-1262	RegSz	Make text and oth…	76-6B-26-00
@%SystemRoot%\system32\p2pcollab.dll,-8042	RegSz	Peer to Peer Trust	00-00-10-0D-00-00
@%SystemRoot%\system32\qagentrt.dll,-10	RegSz	System Health Aut…	00-00

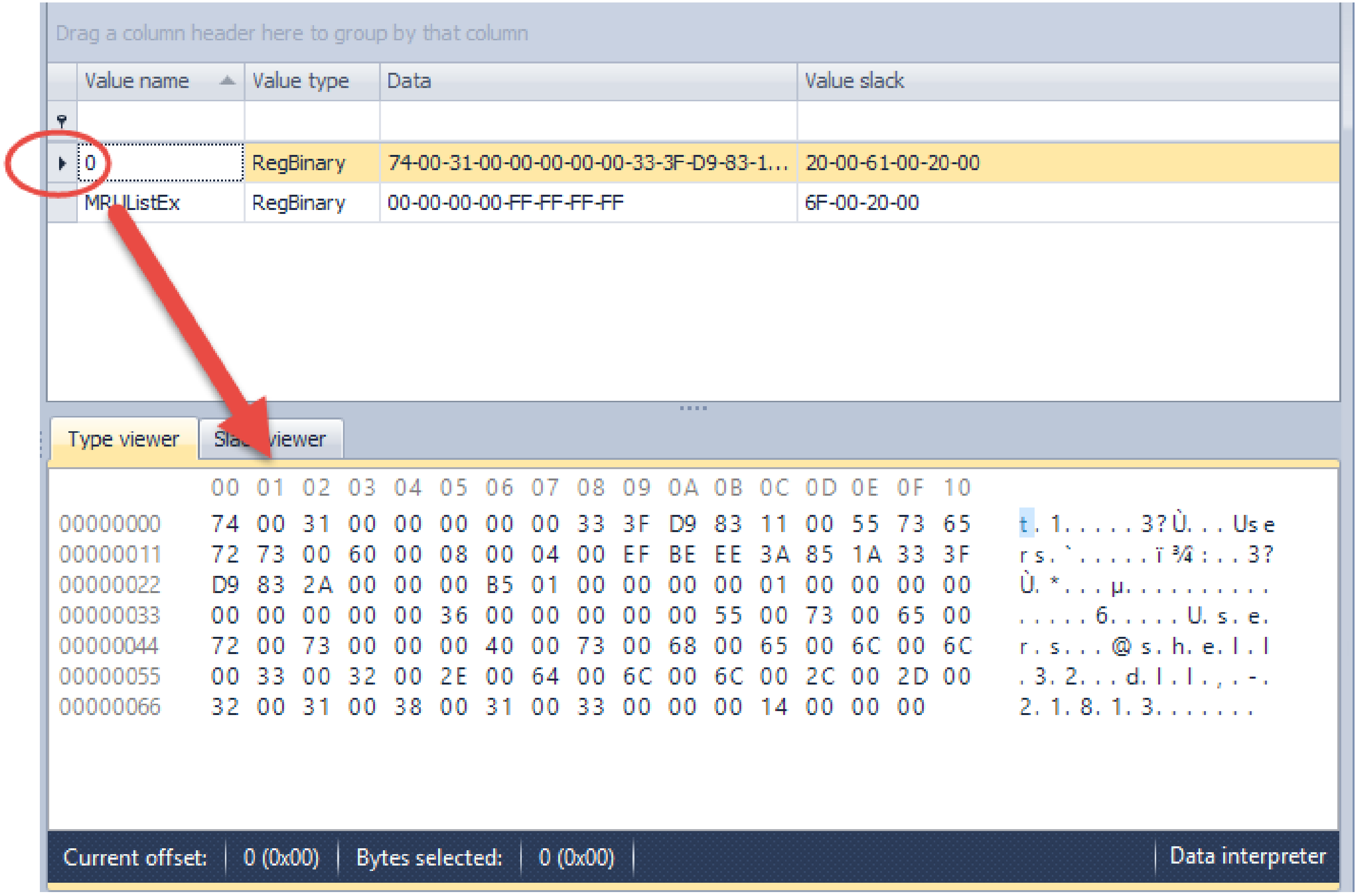
Value name: @%systemroot%\system32\oobefldr.dll,-1222

Value type: RegSz

Slack: 33-32-5C-66

Value: Configure Windows to back up your photos, music, and other files automatically.

For RegBinary keys, a hex viewer will be shown to display the value's binary data.

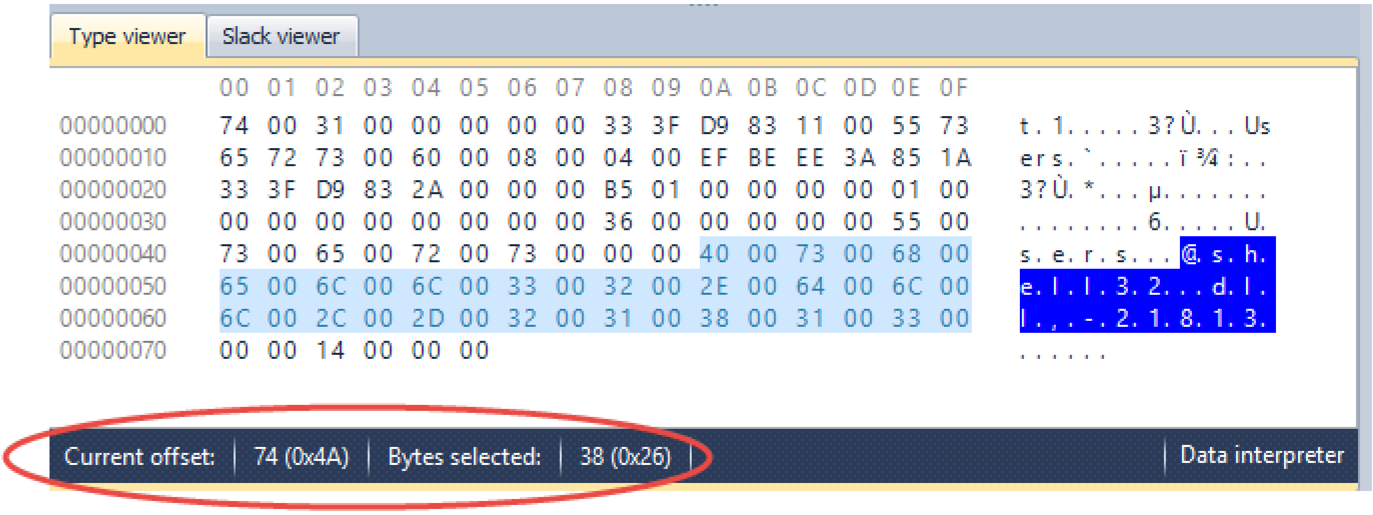


Value name	Value type	Data	Value slack
0	RegBinary	74-00-31-00-00-00-00-00-33-3F-D9-83-1...	20-00-61-00-20-00
MRUListEx	RegBinary	00-00-00-00-FF-FF-FF-FF	6F-00-20-00

```
          00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10
00000000  74 00 31 00 00 00 00 00 33 3F D9 83 11 00 55 73 65   t.1.....3?Ù...Use
00000011  72 73 00 60 00 08 00 04 00 EF BE EE 3A 85 1A 33 3F   rs.`.....ï¾:..3?
00000022  D9 83 2A 00 00 00 B5 01 00 00 00 00 01 00 00 00 00   Ù.*...µ...........
00000033  00 00 00 00 00 36 00 00 00 00 00 55 00 73 00 65 00   .....6.....U.s.e.
00000044  72 00 73 00 00 00 40 00 73 00 68 00 65 00 6C 00 6C   r.s...@.s.h.e.l.l
00000055  00 33 00 32 00 2E 00 64 00 6C 00 6C 00 2C 00 2D 00   .3.2...d.l.l.,.-.
00000066  32 00 31 00 38 00 31 00 33 00 00 00 14 00 00 00      2.1.8.1.3.......
```

Current offset: 0 (0x00) | Bytes selected: 0 (0x00) | Data interpreter

Selecting a byte or a range of bytes will update the Current offset and Bytes selected values at the bottom of the hex viewer.

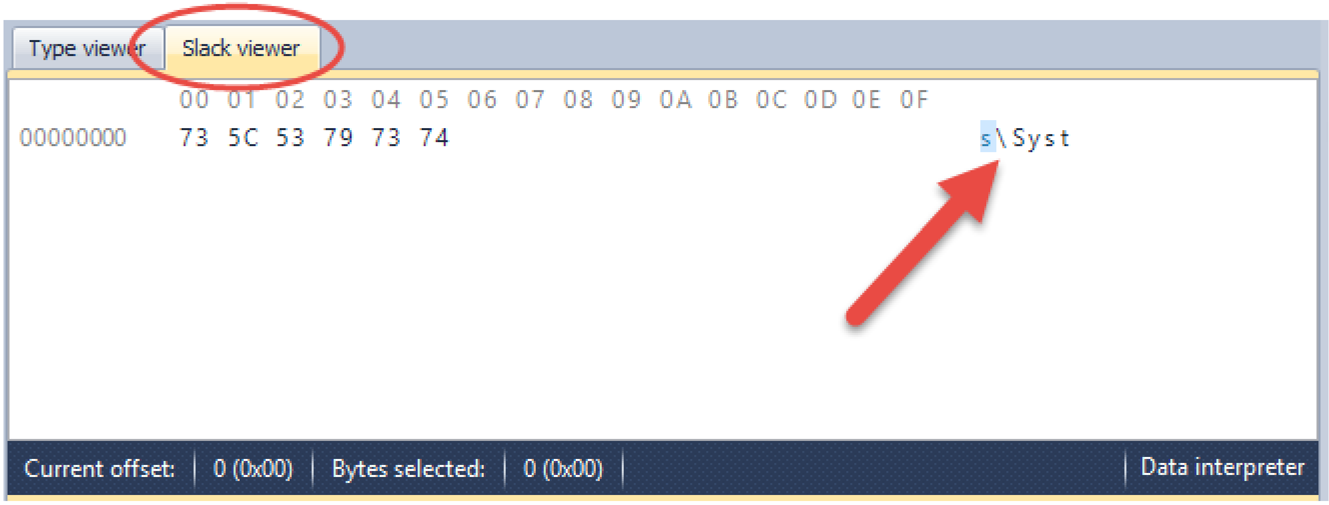


```
          00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000  74 00 31 00 00 00 00 00 33 3F D9 83 11 00 55 73   t.1.....3?Ù...Us
00000010  65 72 73 00 60 00 08 00 04 00 EF BE EE 3A 85 1A   ers.`.....ï¾:..
00000020  33 3F D9 83 2A 00 00 00 B5 01 00 00 00 00 01 00   3?Ù.*...µ.......
00000030  00 00 00 00 00 00 00 00 36 00 00 00 00 00 55 00   ........6.....U.
00000040  73 00 65 00 72 00 73 00 00 00 40 00 73 00 68 00   s.e.r.s...@.s.h.
00000050  65 00 6C 00 6C 00 33 00 32 00 2E 00 64 00 6C 00   e.l.l.3.2...d.l.
00000060  6C 00 2C 00 2D 00 32 00 31 00 38 00 31 00 33 00   l.,.-.2.1.8.1.3.
00000070  00 00 14 00 00 00                                 ......
```

Current offset: 74 (0x4A) | Bytes selected: 38 (0x26) | Data interpreter

Slack viewer

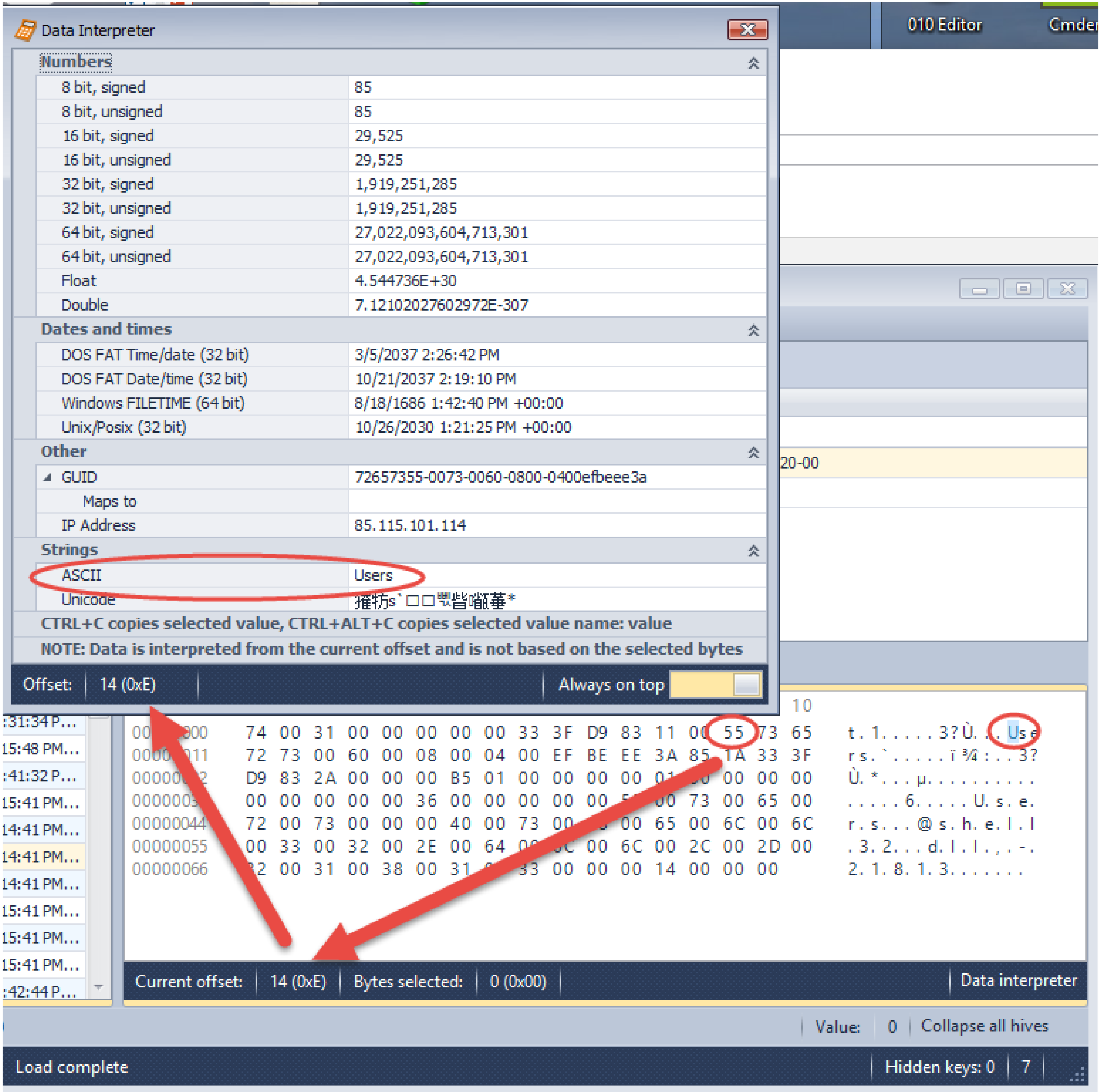
For values that have value slack, a Slack viewer tab will be added. This viewer works the same as the Type viewer for RegBinary values.



```
         00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000 73 5C 53 79 73 74                                s\Syst
```

Data interpreter

In the lower right corner of the hex viewer is a Data interpreter button. Clicking this button will bring up the Data interpreter that converts the raw hexadecimal data into a variety of formats including dates and times, GUIDs, IP addresses, and more. The Data interpreter window is shown below.



Data Interpreter

Numbers	
8 bit, signed	85
8 bit, unsigned	85
16 bit, signed	29,525
16 bit, unsigned	29,525
32 bit, signed	1,919,251,285
32 bit, unsigned	1,919,251,285
64 bit, signed	27,022,093,604,713,301
64 bit, unsigned	27,022,093,604,713,301
Float	4.544736E+30
Double	7.12102027602972E-307
Dates and times	
DOS FAT Time/date (32 bit)	3/5/2037 2:26:42 PM
DOS FAT Date/time (32 bit)	10/21/2037 2:19:10 PM
Windows FILETIME (64 bit)	8/18/1686 1:42:40 PM +00:00
Unix/Posix (32 bit)	10/26/2030 1:21:25 PM +00:00
Other	
GUID	72657355-0073-0060-0800-0400efbeee3a
Maps to	
IP Address	85.115.101.114
Strings	
ASCII	Users
Unicode	獲牦s`ロロ뫯ꏬ䉺華*

CTRL+C copies selected value, CTRL+ALT+C copies selected value name: value

NOTE: Data is interpreted from the current offset and is not based on the selected bytes

Offset: 14 (0xE) | Always on top

```
                                                            10
00000000   74 00 31 00 00 00 00 00 33 3F D9 83 11 00 55 73 65   t.1.....3?Ù...Use
00000011   72 73 00 60 00 08 00 04 00 EF BE EE 3A 85 1A 33 3F   rs.`.....ï¾:..3?
00000022   D9 83 2A 00 00 00 B5 01 00 00 00 00 01 00 00 00 00   Ù.*...µ..........
00000033   00 00 00 00 00 36 00 00 00 00 00 55 00 73 00 65 00   .....6.....U.s.e.
00000044   72 00 73 00 00 00 40 00 73 00 68 00 65 00 6C 00 6C   r.s...@s.h.e.l.l
00000055   00 33 00 32 00 2E 00 64 00 6C 00 6C 00 2C 00 2D 00   .3.2...d.l.l.,.-.
00000066   32 00 31 00 38 00 31 00 33 00 00 00 14 00 00 00      2.1.8.1.3.......
```

Current offset: 14 (0xE) | Bytes selected: 0 (0x00) | Data interpreter

Value: 0 | Collapse all hives

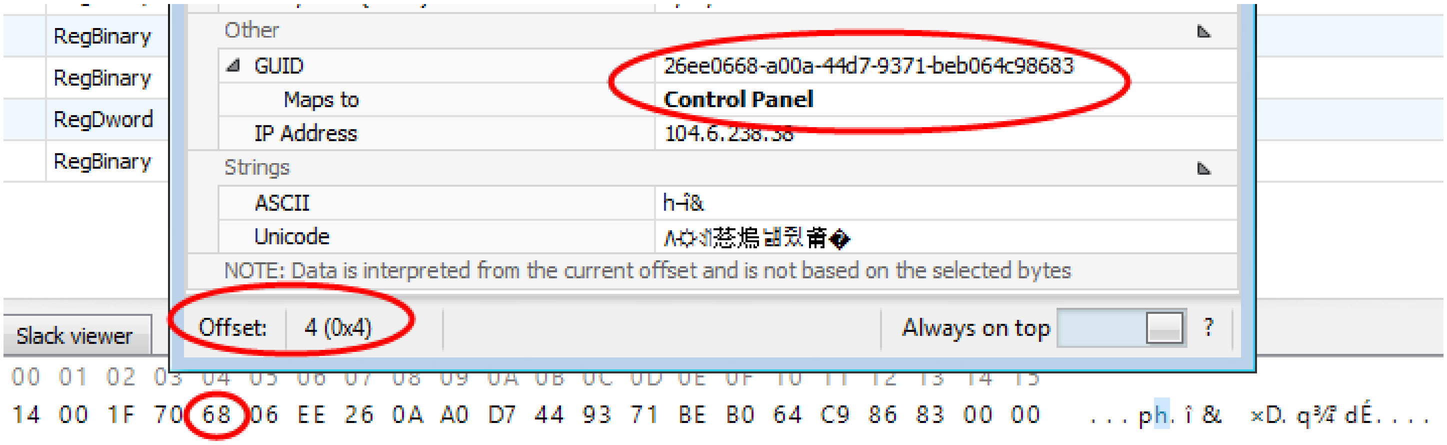
Load complete | Hidden keys: 0 | 7

In the example above, a RegBinary value is selected and the 14th byte has been selected (click on a byte to select it). To the right of the hex display is an ASCII interpretation of the binary data. In this case, 55 corresponds to the ‘U’ character.

The Data interpreter also shows the same offset, 14, but it goes a step further and decodes the ASCII string ‘Users’ from bytes 55 73 65 72 73. Registry Explorer will look for a single Null terminator for ASCII strings (00) and double

Null terminators (00 00) for Unicode strings. If no Null terminators are found, the bytes will be interpreted from the current offset to the end of the data.

The Data interpreter can also convert GUIDs to known folder/location names as seen below.



RegBinary
RegBinary
RegDword
RegBinary

Other
GUID: 26ee0668-a00a-44d7-9371-beb064c98683
Maps to: **Control Panel**
IP Address: 104.6.238.38

Strings
ASCII: h-î&
Unicode: ʎ✡刂慈焉 胡烈甬�

NOTE: Data is interpreted from the current offset and is not based on the selected bytes

Offset: 4 (0x4)

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15
14 00 1F 70 68 06 EE 26 0A A0 D7 44 93 71 BE B0 64 C9 86 83 00 00    . . . ph. î &  ×D. q¾ dÉ. . . .
```

In this particular case, a GUID was found at offset 0x04, 26ee0668-a00a-44d7-9371-beb064c98683, that maps to 'Control panel.'

To copy values from the Data interpreter to the clipboard, press **Ctrl+C**. To copy both the value type and the value (in type: value format), press **Ctrl+Alt+C**.

Interacting with deleted keys

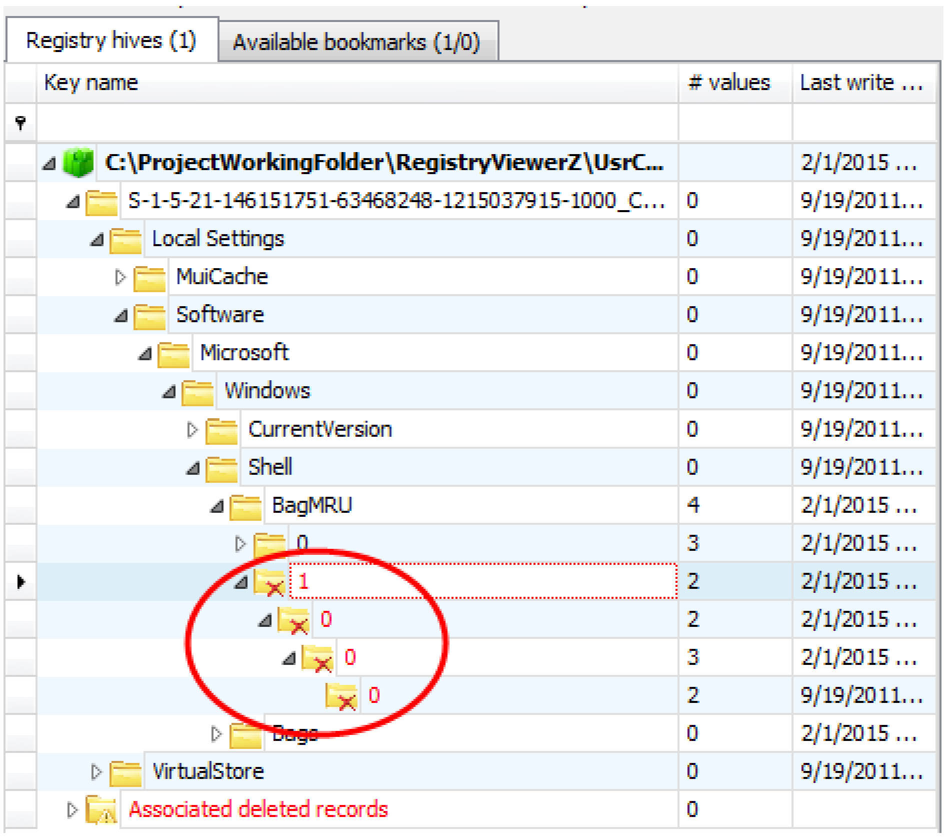
Registry Explorer is capable of recovering both deleted Registry keys and values. It also reassociates deleted values with their parent keys and subkeys to their parent keys.

In some cases, it is not possible to reassociate recovered keys to an active Registry key because the deleted key's parent cell index does not correspond to a key's offset in the active Registry.

Registry Explorer shows recovered deleted keys in up to three ways: "Inlined' with existing keys (that is, deleted keys are shown where they used to exist), Associated deleted records (the same info as Inlined keys, but the parent keys are placeholders), and Unassociated deleted records (no parent key could be found in the active Registry).

Inlined with existing keys

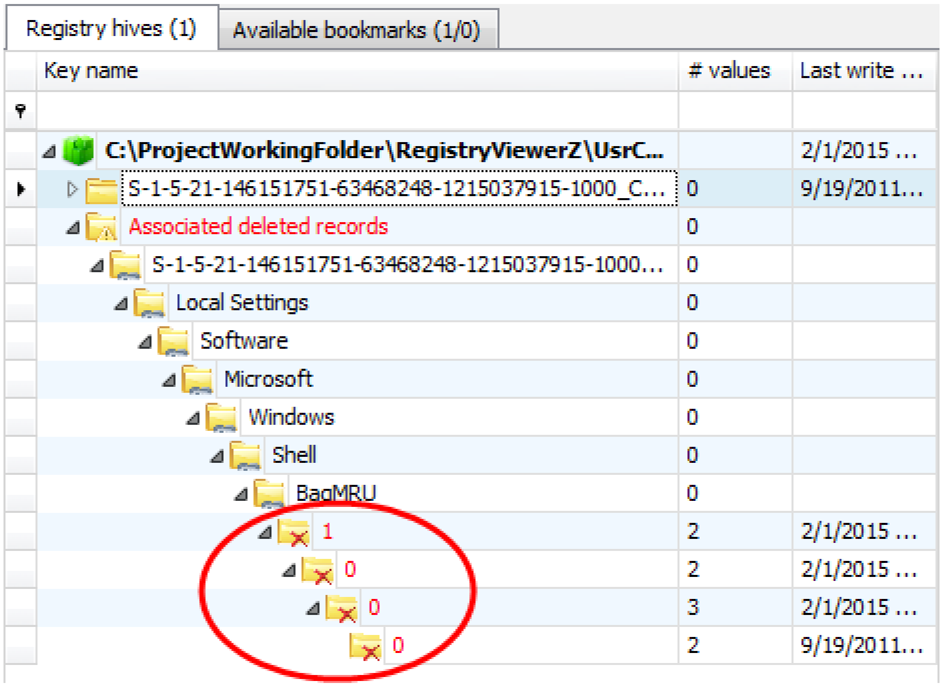
When Registry Explorer can reassociate a key with an active parent key, it is shown under the root key under its parent key. The icon for the deleted key (and all its subkeys) is the same as an active key, but a red X is shown in the lower right corner to denote it is a deleted key. The font for deleted keys is red.



Key name	# values	Last write ...
C:\ProjectWorkingFolder\RegistryViewerZ\UsrC...		2/1/2015 ...
S-1-5-21-146151751-63468248-1215037915-1000_C...	0	9/19/2011...
Local Settings	0	9/19/2011...
MuiCache	0	9/19/2011...
Software	0	9/19/2011...
Microsoft	0	9/19/2011...
Windows	0	9/19/2011...
CurrentVersion	0	9/19/2011...
Shell	0	9/19/2011...
BagMRU	4	2/1/2015 ...
0	3	2/1/2015 ...
1	2	2/1/2015 ...
0	2	2/1/2015 ...
0	3	2/1/2015 ...
0	2	9/19/2011...
Bags	0	2/1/2015 ...
VirtualStore	0	9/19/2011...
Associated deleted records	0	

Associated deleted records

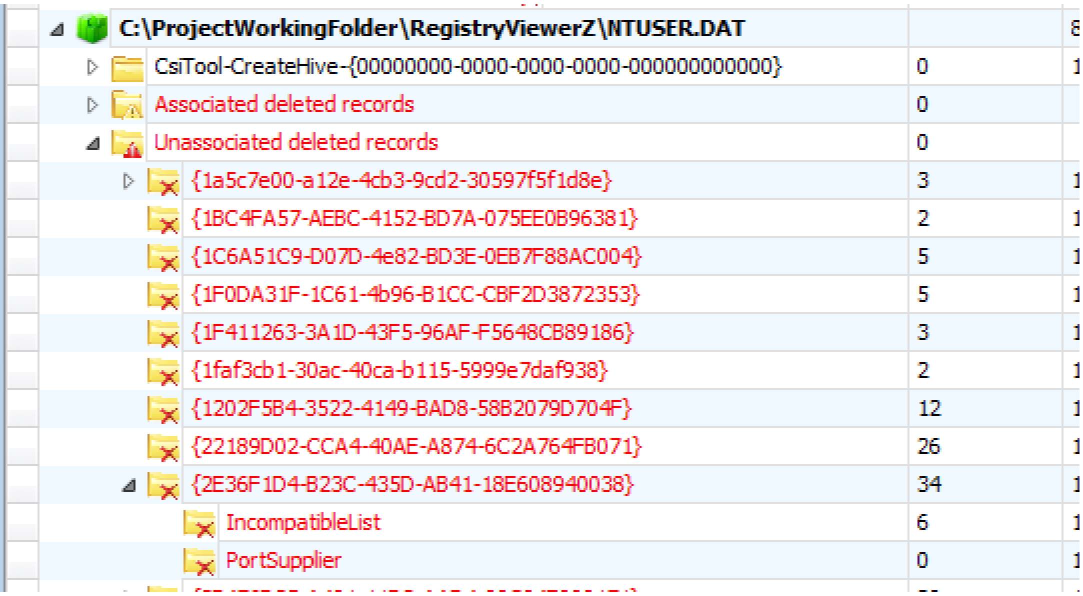
All associated deleted records are also shown under a virtual key called ‘Associated deleted’ records. Under this key, placeholder keys (keys with a link icon in the lower right) are created that denote active keys, down to the point where the deleted key can be found. In the example below, the same path as seen above is reflected down to the ‘BagMRU’ key. At this point, the icon and font color changes to indicate the key is in fact deleted and has been reassociated.



Key name	# values	Last write ...
C:\ProjectWorkingFolder\RegistryViewerZ\UsrC...		2/1/2015 ...
S-1-5-21-146151751-63468248-1215037915-1000_C...	0	9/19/2011...
Associated deleted records	0	
S-1-5-21-146151751-63468248-1215037915-1000...	0	
Local Settings	0	
Software	0	
Microsoft	0	
Windows	0	
Shell	0	
BagMRU	0	
1	2	2/1/2015 ...
0	2	2/1/2015 ...
0	3	2/1/2015 ...
0	2	9/19/2011...

Unassociated deleted records

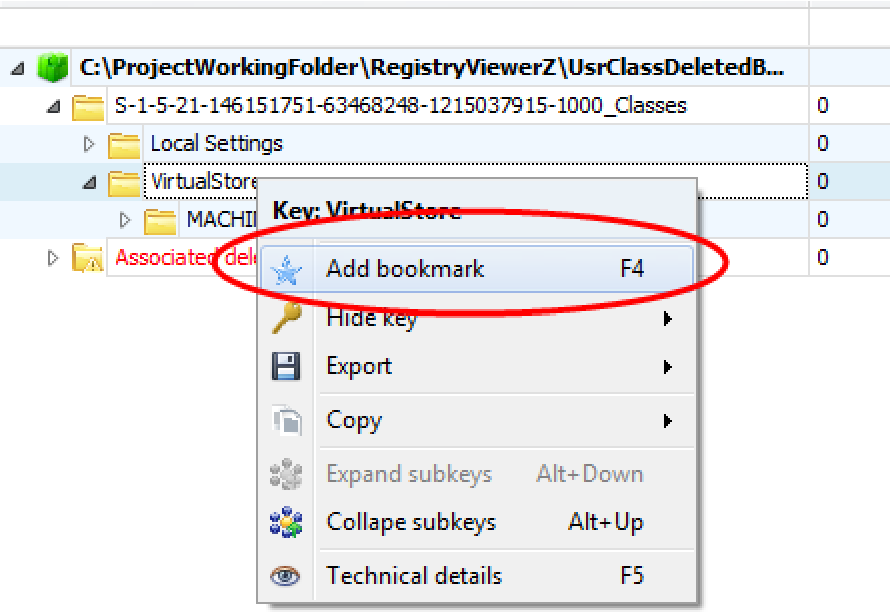
In the cases where an active key could not be found, the recovered deleted key will be placed under another virtual key called ‘Unassociated deleted records’ that functions in a similar way to the Associated deleted records. The primary difference between the two is that there will not be any active parent keys shown for unassociated records. Unassociated records can be explored like any other records (looking at values, viewing Technical details, etc.).



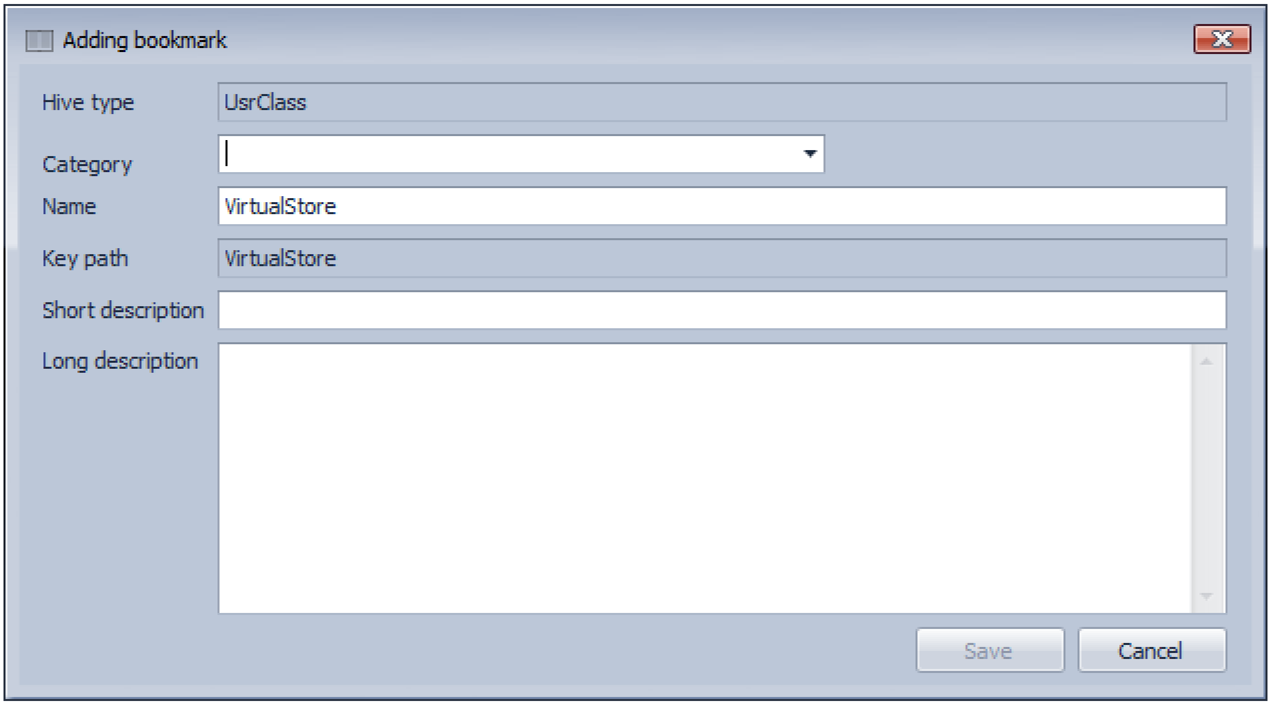
C:\ProjectWorkingFolder\RegistryViewerZ\NTUSER.DAT		8
CsiTool-CreateHive-{00000000-0000-0000-0000-000000000000}	0	1
Associated deleted records	0	
Unassociated deleted records	0	
{1a5c7e00-a12e-4cb3-9cd2-30597f5f1d8e}	3	1
{1BC4FA57-AEBC-4152-BD7A-075EE0B96381}	2	1
{1C6A51C9-D07D-4e82-BD3E-0EB7F88AC004}	5	1
{1F0DA31F-1C61-4b96-B1CC-CBF2D3872353}	5	1
{1F411263-3A1D-43F5-96AF-F5648CB89186}	3	1
{1faf3cb1-30ac-40ca-b115-5999e7daf938}	2	1
{1202F5B4-3522-4149-BAD8-58B2079D704F}	12	1
{22189D02-CCA4-40AE-A874-6C2A764FB071}	26	1
{2E36F1D4-B23C-435D-AB41-18E608940038}	34	1
IncompatibleList	6	1
PortSupplier	0	1

Creating bookmarks

To create a bookmark, right click on a key and select Add bookmark.



Since Registry Explorer knows the hive type and key path already, these values will be prepopulated. In the example below, a UsrClass hive is active and the VirtualStore key is selected.


Adding bookmark

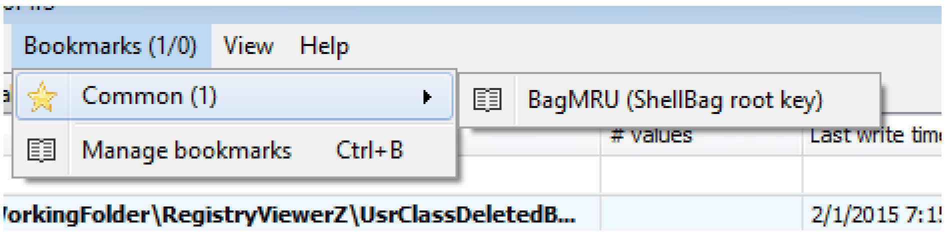
Hive type	UsrClass
Category	
Name	VirtualStore
Key path	VirtualStore
Short description	
Long description	

The Category field allows you to place this particular Registry key into a high-level group. This will eventually be used for reporting. Several preexisting categories are included, but typing a new Category will add it to the list.

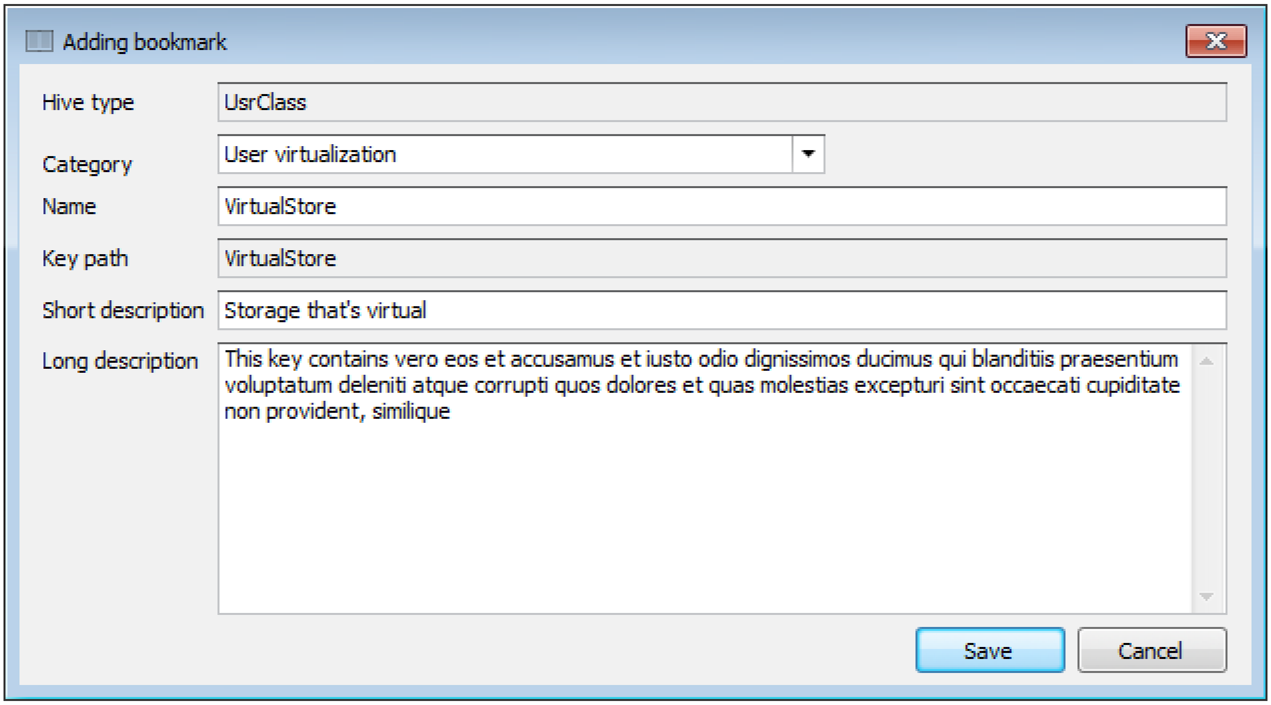
The Short description serves as a summary for what the key means or why it is relevant. The value entered for Short description will show up after the name of the bookmark in the bookmarks menu.

The Long description should contain technical information, links to web pages with more information, or any other information you want to convey.

Recall the bookmarks menu is dynamic and will update according to the keys that are available for the selected hive. If, before adding the bookmark, the Bookmarks menu looked like this:



And our new bookmark looks like this:



Hive type: UsrClass

Category: User virtualization

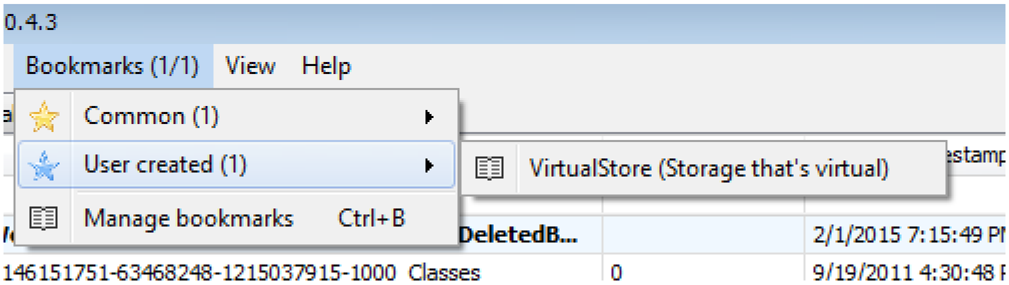
Name: VirtualStore

Key path: VirtualStore

Short description: Storage that's virtual

Long description: This key contains vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique

The Bookmarks menu will look like this once the Save button is clicked:

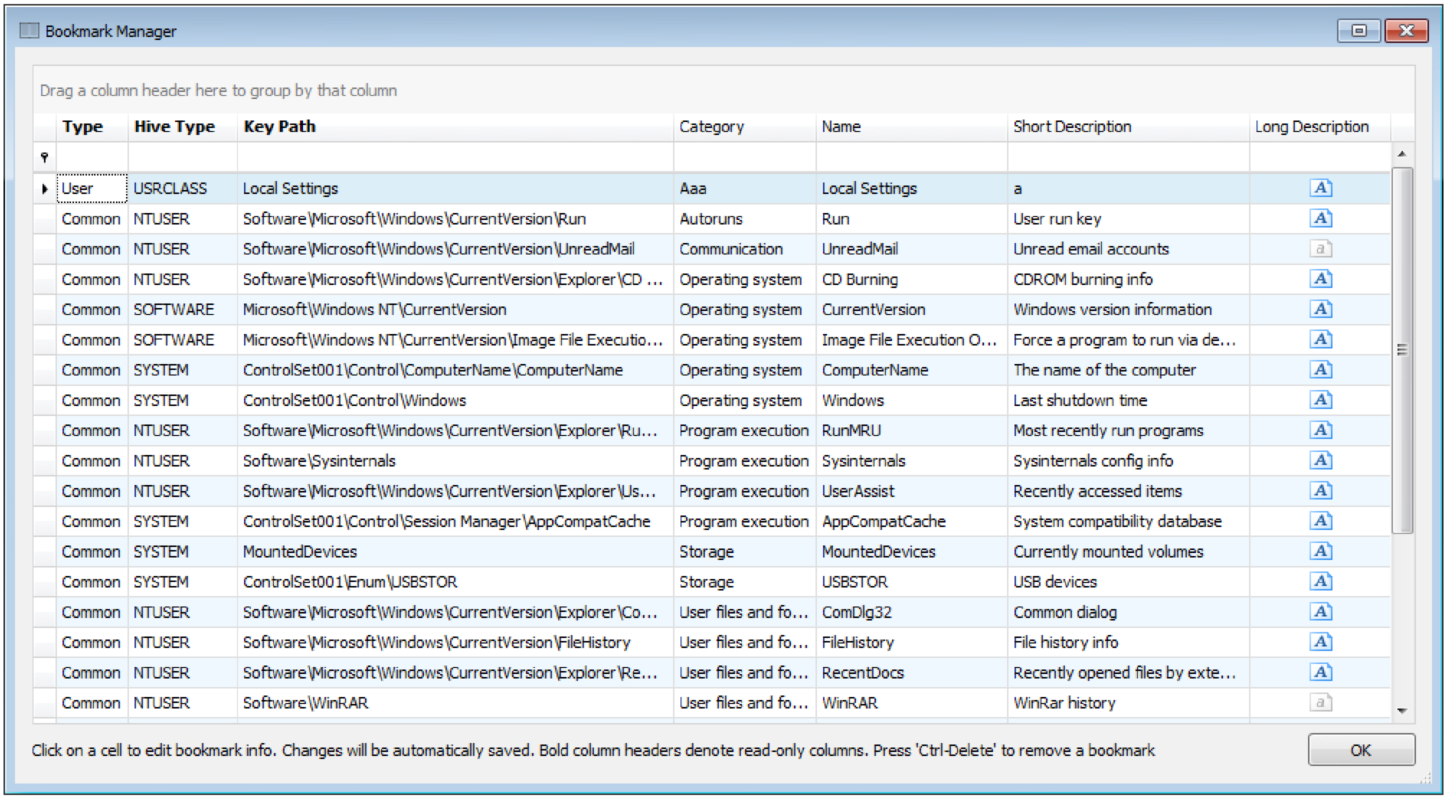


A ‘User created’ menu is now visible as is our VirtualStore bookmark (with the short description shown in parenthesis after the key name).

Selecting the 'VirtualStore' bookmark expands all child keys to the bookmarked key in the selected Registry hive.

Managing bookmarks

Recall bookmarks are kept in two folders, one for included bookmarks and one for user created bookmarks. Registry Explorer contains a Bookmark manager that is available under the Bookmarks menu.



Type	Hive Type	Key Path	Category	Name	Short Description	Long Description
User	USRCLASS	Local Settings	Aaa	Local Settings	a	A
Common	NTUSER	Software\Microsoft\Windows\CurrentVersion\Run	Autoruns	Run	User run key	A
Common	NTUSER	Software\Microsoft\Windows\CurrentVersion\UnreadMail	Communication	UnreadMail	Unread email accounts	a
Common	NTUSER	Software\Microsoft\Windows\CurrentVersion\Explorer\CD ...	Operating system	CD Burning	CDROM burning info	A
Common	SOFTWARE	Microsoft\Windows NT\CurrentVersion	Operating system	CurrentVersion	Windows version information	A
Common	SOFTWARE	Microsoft\Windows NT\CurrentVersion\Image File Executio...	Operating system	Image File Execution O...	Force a program to run via de...	A
Common	SYSTEM	ControlSet001\Control\ComputerName\ComputerName	Operating system	ComputerName	The name of the computer	A
Common	SYSTEM	ControlSet001\Control\Windows	Operating system	Windows	Last shutdown time	A
Common	NTUSER	Software\Microsoft\Windows\CurrentVersion\Explorer\Ru...	Program execution	RunMRU	Most recently run programs	A
Common	NTUSER	Software\Sysinternals	Program execution	Sysinternals	Sysinternals config info	A
Common	NTUSER	Software\Microsoft\Windows\CurrentVersion\Explorer\Us...	Program execution	UserAssist	Recently accessed items	A
Common	SYSTEM	ControlSet001\Control\Session Manager\AppCompatCache	Program execution	AppCompatCache	System compatibility database	A
Common	SYSTEM	MountedDevices	Storage	MountedDevices	Currently mounted volumes	A
Common	SYSTEM	ControlSet001\Enum\USBSTOR	Storage	USBSTOR	USB devices	A
Common	NTUSER	Software\Microsoft\Windows\CurrentVersion\Explorer\Co...	User files and fo...	ComDlg32	Common dialog	A
Common	NTUSER	Software\Microsoft\Windows\CurrentVersion\FileHistory	User files and fo...	FileHistory	File history info	A
Common	NTUSER	Software\Microsoft\Windows\CurrentVersion\Explorer\Re...	User files and fo...	RecentDocs	Recently opened files by exte...	A
Common	NTUSER	Software\WinRAR	User files and fo...	WinRAR	WinRar history	a

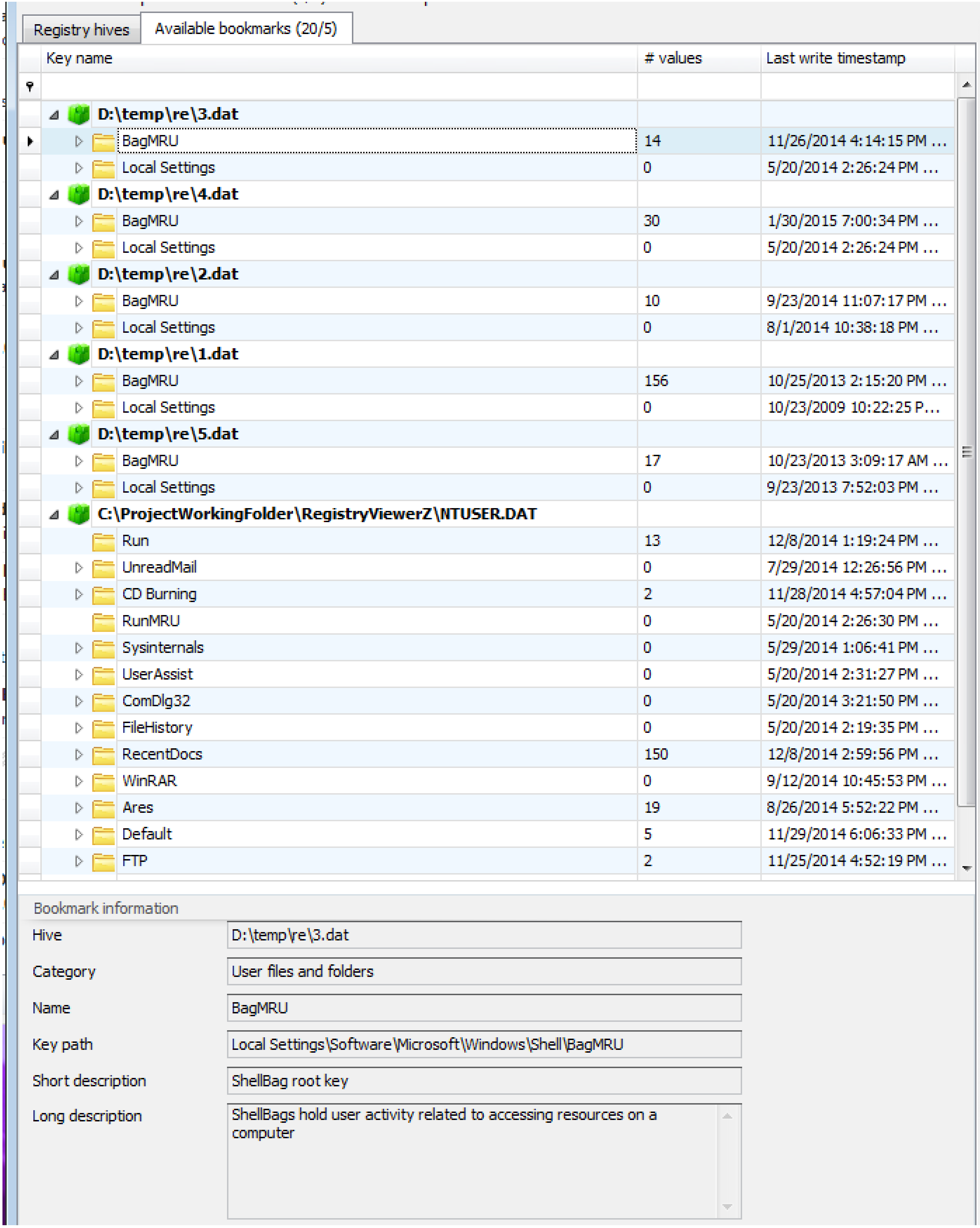
Click on a cell to edit bookmark info. Changes will be automatically saved. Bold column headers denote read-only columns. Press 'Ctrl-Delete' to remove a bookmark

The column headers in **bold** (Type, Hive Type and Key Path) are read only. To edit any of the other columns, click on that column's value and adjust. The bookmark is saved automatically and the Bookmarks menu will be updated accordingly.

Available bookmarks

The Available bookmarks tab is an optimized way to view all available bookmarks across all loaded hives. Using the Available bookmarks tab allows you to see all bookmarks that exist without the distraction of parent keys or having to drill down into different hives to review things.

After loading one or more hives, click on the Available bookmarks tab. An example of this is shown below.



Key name	# values	Last write timestamp
D:\temp\re\3.dat		
BagMRU	14	11/26/2014 4:14:15 PM ...
Local Settings	0	5/20/2014 2:26:24 PM ...
D:\temp\re\4.dat		
BagMRU	30	1/30/2015 7:00:34 PM ...
Local Settings	0	5/20/2014 2:26:24 PM ...
D:\temp\re\2.dat		
BagMRU	10	9/23/2014 11:07:17 PM ...
Local Settings	0	8/1/2014 10:38:18 PM ...
D:\temp\re\1.dat		
BagMRU	156	10/25/2013 2:15:20 PM ...
Local Settings	0	10/23/2009 10:22:25 P...
D:\temp\re\5.dat		
BagMRU	17	10/23/2013 3:09:17 AM ...
Local Settings	0	9/23/2013 7:52:03 PM ...
C:\ProjectWorkingFolder\RegistryViewerZ\NTUSER.DAT		
Run	13	12/8/2014 1:19:24 PM ...
UnreadMail	0	7/29/2014 12:26:56 PM ...
CD Burning	2	11/28/2014 4:57:04 PM ...
RunMRU	0	5/20/2014 2:26:30 PM ...
Sysinternals	0	5/29/2014 1:06:41 PM ...
UserAssist	0	5/20/2014 2:31:27 PM ...
ComDlg32	0	5/20/2014 3:21:50 PM ...
FileHistory	0	5/20/2014 2:19:35 PM ...
RecentDocs	150	12/8/2014 2:59:56 PM ...
WinRAR	0	9/12/2014 10:45:53 PM ...
Ares	19	8/26/2014 5:52:22 PM ...
Default	5	11/29/2014 6:06:33 PM ...
FTP	2	11/25/2014 4:52:19 PM ...

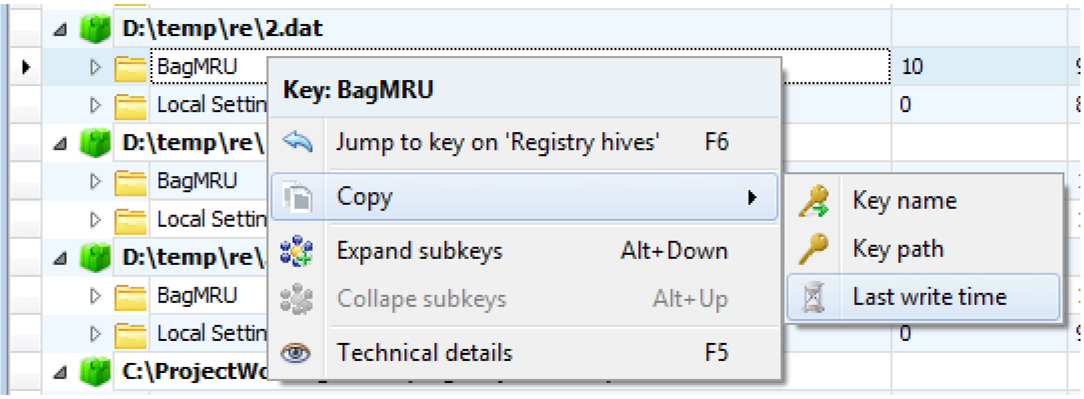
Bookmark information

Hive	D:\temp\re\3.dat
Category	User files and folders
Name	BagMRU
Key path	Local Settings\Software\Microsoft\Windows\Shell\BagMRU
Short description	ShellBag root key
Long description	ShellBags hold user activity related to accessing resources on a computer

When the root folder for a bookmark is selected (BagMRU in the example above), information about the bookmark is shown at the bottom of the window in the Bookmark information section.

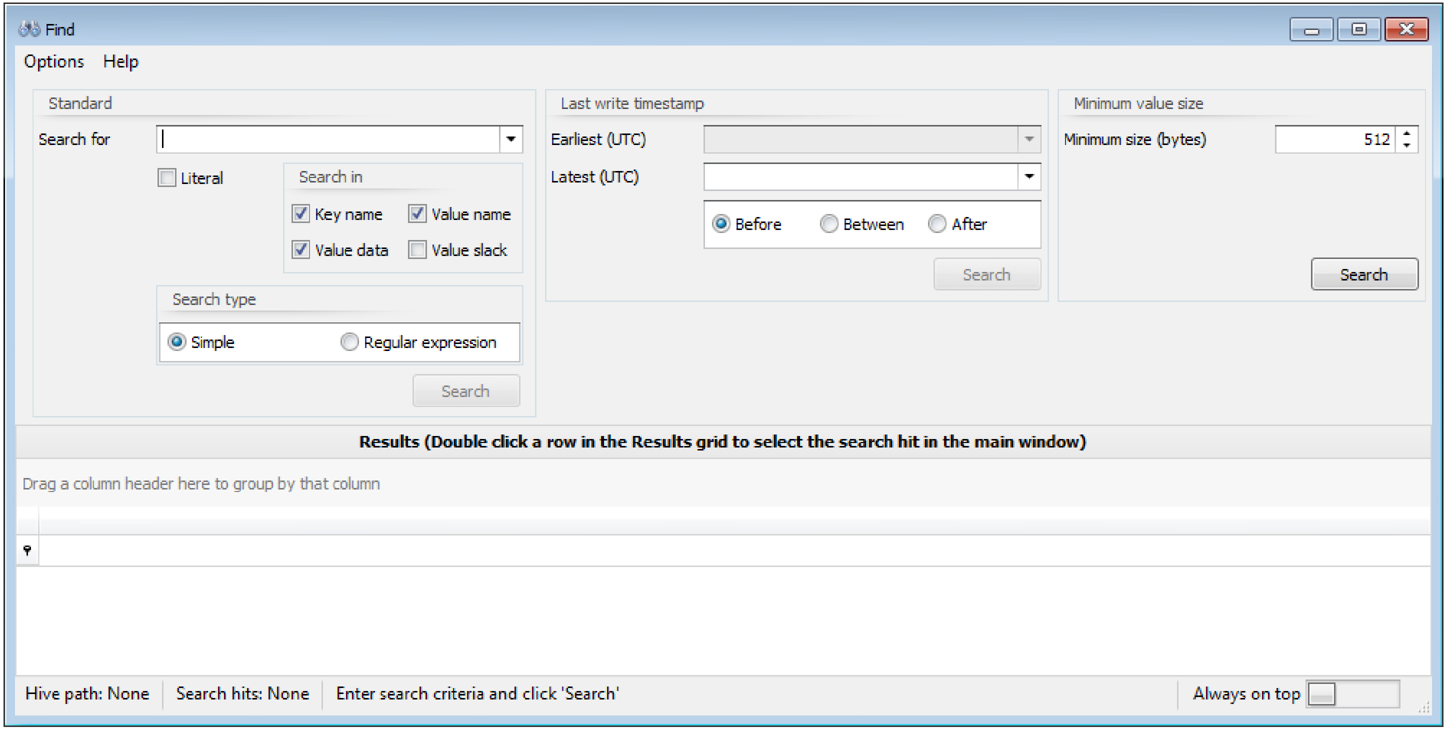
The numbers at the end of the Available bookmarks tab indicate the total number of common bookmarks (20 in this case) and the total number of user created bookmarks (5 in this case). Available bookmarks dynamically updates as hives are loaded/unloaded, bookmarks are created/removed, etc.

Right clicking on a key brings up a context menu. The options work the same way as on the Registry hives tab. The 'Jump to key' option will change the active tab to the 'Registry hives' tab and select the bookmarked key. This is useful to see the bookmarked key in context with other keys.



Searching

Registry Explorer contains powerful searching capabilities including standard string searches and regular expression based searches. It can also search for keys where the last write timestamp is before, between or after a given timestamp or pair of timestamps, or for values that have a data size greater than a certain number of bytes.



Standard

Search for

☐ Literal

Search in

☑ Key name ☑ Value name

☑ Value data ☐ Value slack

Search type

● Simple ○ Regular expression

Last write timestamp

Earliest (UTC)

Latest (UTC)

● Before ○ Between ○ After

Minimum value size

Minimum size (bytes) 512

Results (Double click a row in the Results grid to select the search hit in the main window)

Drag a column header here to group by that column

Hive path: None | Search hits: None | Enter search criteria and click 'Search' | Always on top ☐

Registry Explorer allows you to search all hives at once across key names, value names, value data and/or value slack. Searching is done against each hive asynchronously and results will appear as they are available.

Options menu

Clear recent

When conducting a standard search, search terms in the 'Search for' box are remembered between program executions. Use this option to clear these recent searches.

Convert

The convert menu contains options to convert the entered string in the 'Search for' box to its ASCII or Unicode hexadecimal value. This is useful when searching for patterns in Value data.

For example, entering 'Eric' (without the quotes) and using the conversion options results in the following being shown in the 'Search for' box:

- ASCII: 45-72-69-63
- Unicode: 45-00-72-00-69-00-63-00

The converted value can now be used to search for the initial string in its encoded form.

Help menu

Search tips

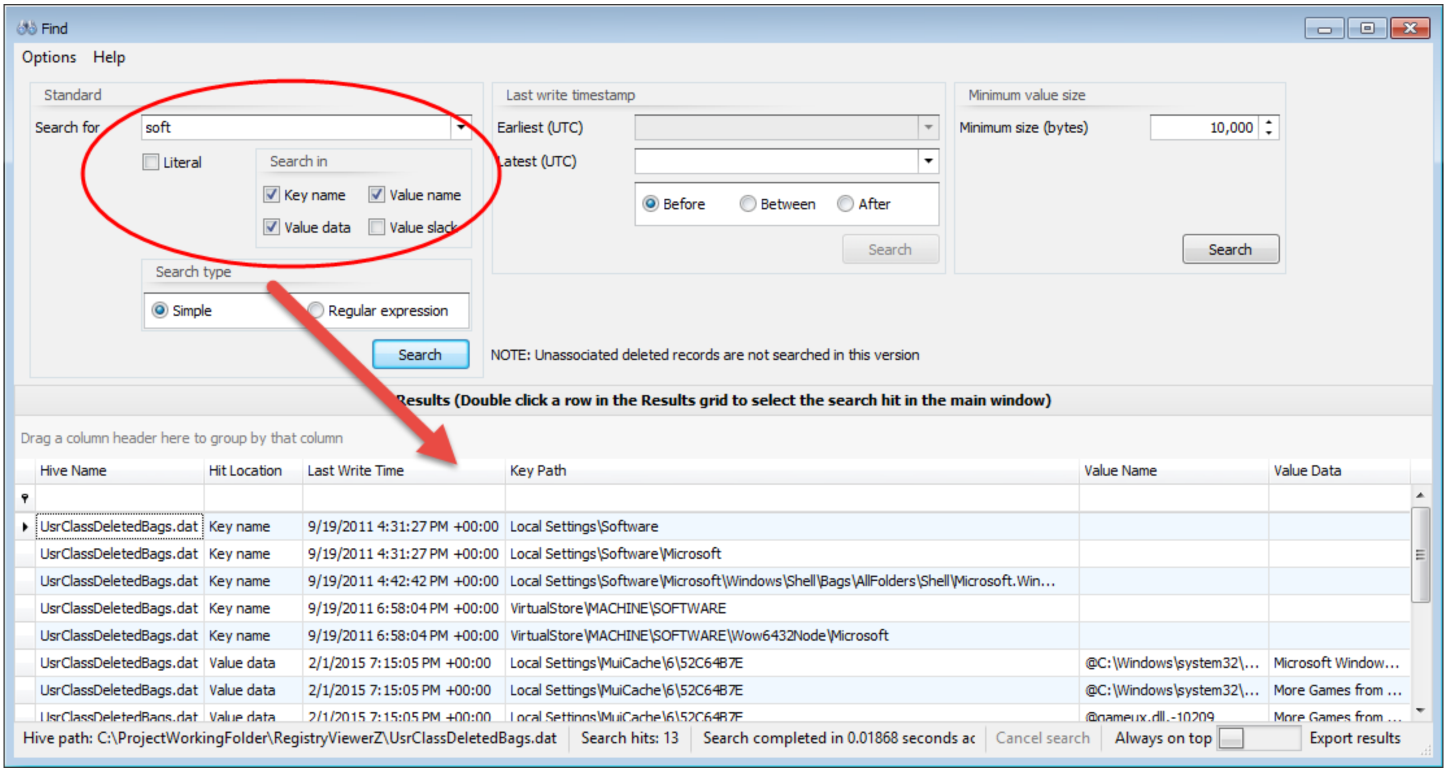
Shows several tips for different kinds of searches

Regular expressions

Launches a web page with information about creating .net regular expressions

Standard search

To conduct a standard search, simply enter a value in the 'Search for' box and click 'Search.' You can also press the Enter key after entering a search term to perform the search.



Standard

Search for: soft

□ Literal

Search in: ☑ Key name ☑ Value name ☑ Value data □ Value slack

Search type: ● Simple ○ Regular expression

Last write timestamp

Earliest (UTC):

Latest (UTC):

● Before ○ Between ○ After

Minimum value size

Minimum size (bytes): 10,000

NOTE: Unassociated deleted records are not searched in this version

Results (Double click a row in the Results grid to select the search hit in the main window)

Drag a column header here to group by that column

Hive Name	Hit Location	Last Write Time	Key Path	Value Name	Value Data
UsrClassDeletedBags.dat	Key name	9/19/2011 4:31:27 PM +00:00	Local Settings\Software		
UsrClassDeletedBags.dat	Key name	9/19/2011 4:31:27 PM +00:00	Local Settings\Software\Microsoft		
UsrClassDeletedBags.dat	Key name	9/19/2011 4:42:42 PM +00:00	Local Settings\Software\Microsoft\Windows\Shell\Bags\AllFolders\Shell\Microsoft.Win...		
UsrClassDeletedBags.dat	Key name	9/19/2011 6:58:04 PM +00:00	VirtualStore\MACHINE\SOFTWARE		
UsrClassDeletedBags.dat	Key name	9/19/2011 6:58:04 PM +00:00	VirtualStore\MACHINE\SOFTWARE\Wow6432Node\Microsoft		
UsrClassDeletedBags.dat	Value data	2/1/2015 7:15:05 PM +00:00	Local Settings\MuiCache\6\52C64B7E	@C:\Windows\system32\...	Microsoft Window...
UsrClassDeletedBags.dat	Value data	2/1/2015 7:15:05 PM +00:00	Local Settings\MuiCache\6\52C64B7E	@C:\Windows\system32\...	More Games from ...
UsrClassDeletedBags.dat	Value data	2/1/2015 7:15:05 PM +00:00	Local Settings\MuiCache\6\52C64B7E	@gameux.dll,-10209	More Games from ...

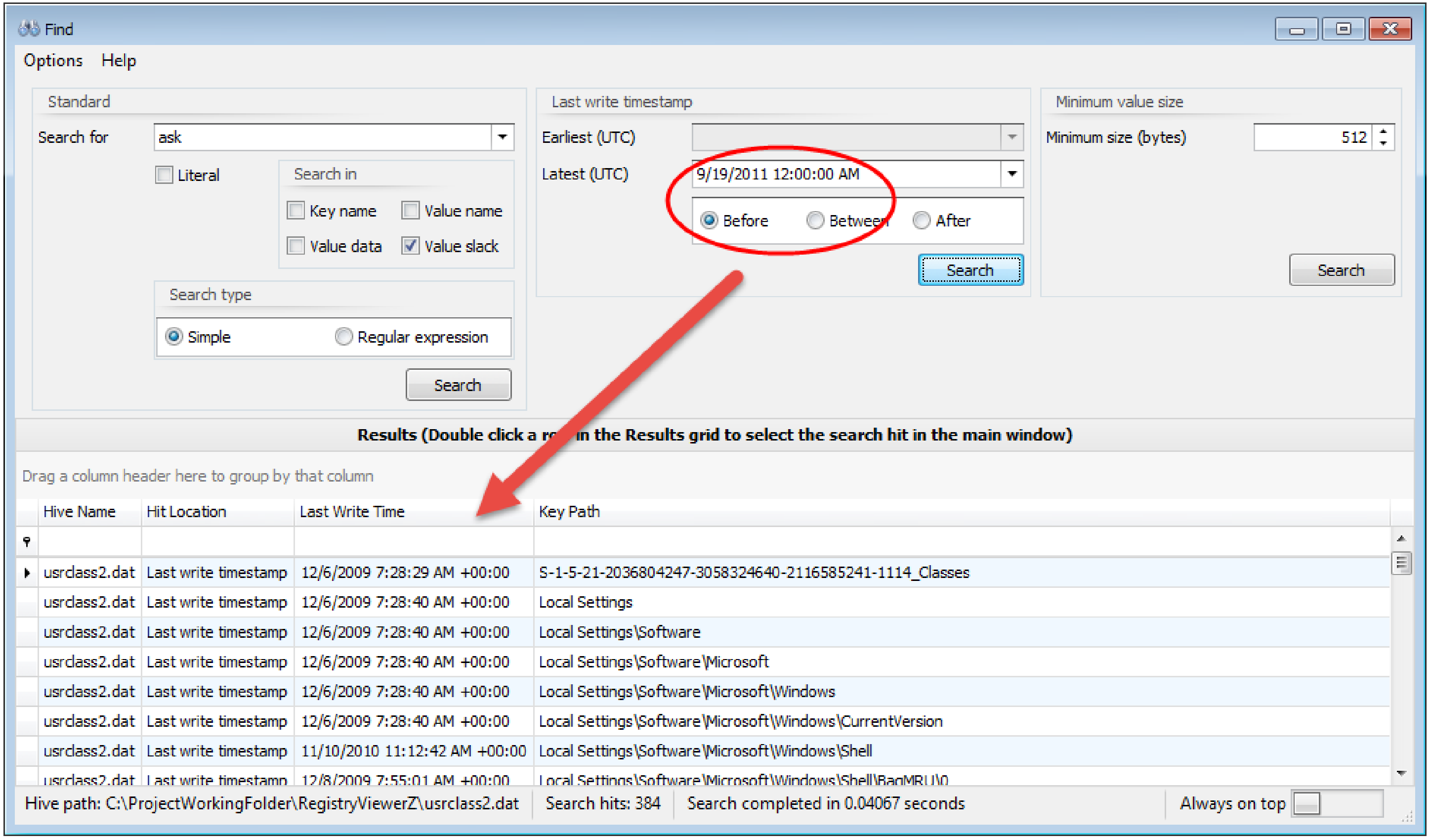
Hive path: C:\ProjectWorkingFolder\RegistryViewerZ\UsrClassDeletedBags.dat | Search hits: 13 | Search completed in 0.01868 seconds ac | Always on top □

The Literal checkbox controls whether the term searched for is looked for in binary data when searching in value data and/or slack. This is explained in more detail below.

If you entered a regular expression, change the radio button to 'Regular expression' so Registry Explorer knows to use RegEx when searching. The Help menu can be used to get additional help on building .net regular expressions. For additional resources on regular expressions, click here to view the regular expression searching section for RECmd.

Last write timestamp search

To conduct a last write timestamp search, choose the date range to search for via the radio buttons and enter the required time stamp values, and then click Search (or press Enter).



Search for: ask

☐ Literal

☐ Key name ☐ Value name

☐ Value data ☑ Value slack

● Simple ○ Regular expression

Earliest (UTC):

Latest (UTC): 9/19/2011 12:00:00 AM

● Before ○ Between ○ After

Minimum size (bytes): 512

Results (Double click a row in the Results grid to select the search hit in the main window)

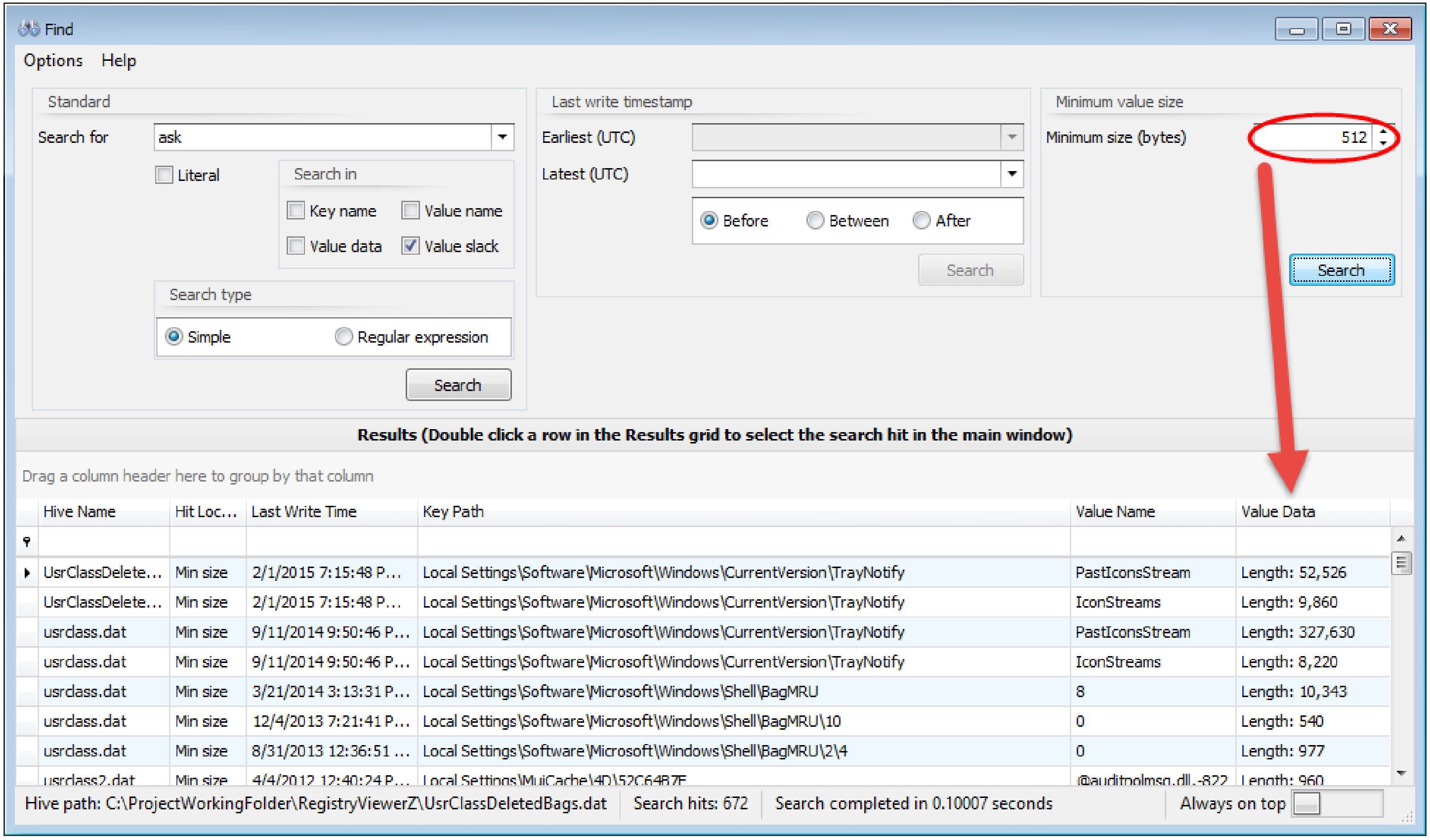
Drag a column header here to group by that column

Hive Name	Hit Location	Last Write Time	Key Path
usrclass2.dat	Last write timestamp	12/6/2009 7:28:29 AM +00:00	S-1-5-21-2036804247-3058324640-2116585241-1114_Classes
usrclass2.dat	Last write timestamp	12/6/2009 7:28:40 AM +00:00	Local Settings
usrclass2.dat	Last write timestamp	12/6/2009 7:28:40 AM +00:00	Local Settings\Software
usrclass2.dat	Last write timestamp	12/6/2009 7:28:40 AM +00:00	Local Settings\Software\Microsoft
usrclass2.dat	Last write timestamp	12/6/2009 7:28:40 AM +00:00	Local Settings\Software\Microsoft\Windows
usrclass2.dat	Last write timestamp	12/6/2009 7:28:40 AM +00:00	Local Settings\Software\Microsoft\Windows\CurrentVersion
usrclass2.dat	Last write timestamp	11/10/2010 11:12:42 AM +00:00	Local Settings\Software\Microsoft\Windows\Shell
usrclass2.dat	Last write timestamp	12/8/2009 7:55:01 AM +00:00	Local Settings\Software\Microsoft\Windows\Shell\BagMRU\0

Hive path: C:\ProjectWorkingFolder\RegistryViewerZ\usrclass2.dat | Search hits: 384 | Search completed in 0.04067 seconds | Always on top ☐

Minimum value size search

When searching for values above a minimum size, the size of the value data’s length is shown in the Value Data column as seen below.



Search for: ask

□ Literal

Search in: □ Key name ☐ Value name □ Value data ☑ Value slack

Search type: ● Simple ○ Regular expression

Earliest (UTC):

Latest (UTC):

● Before ○ Between ○ After

Minimum size (bytes): 512

Results (Double click a row in the Results grid to select the search hit in the main window)

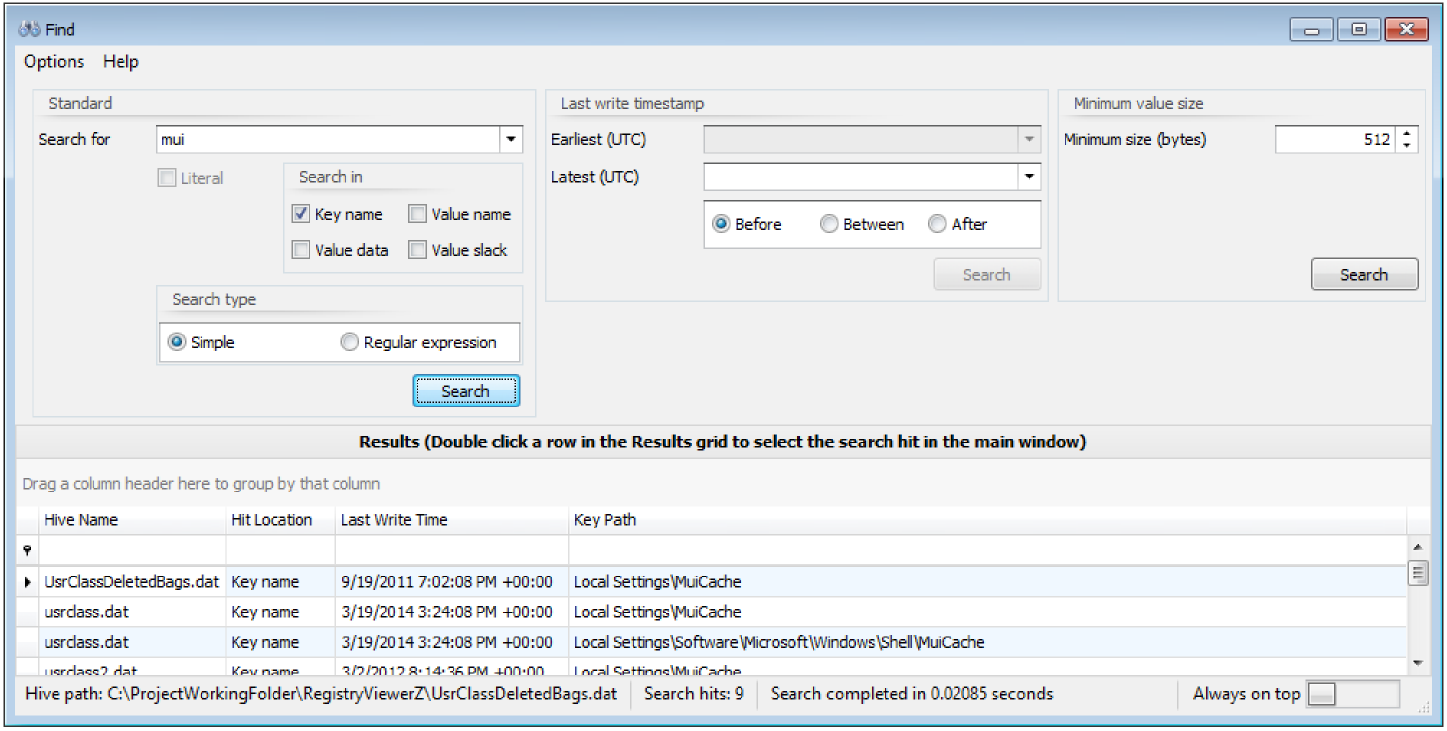
Drag a column header here to group by that column

Hive Name	Hit Loc…	Last Write Time	Key Path	Value Name	Value Data
UsrClassDelete…	Min size	2/1/2015 7:15:48 P…	Local Settings\Software\Microsoft\Windows\CurrentVersion\TrayNotify	PastIconsStream	Length: 52,526
UsrClassDelete…	Min size	2/1/2015 7:15:48 P…	Local Settings\Software\Microsoft\Windows\CurrentVersion\TrayNotify	IconStreams	Length: 9,860
usrclass.dat	Min size	9/11/2014 9:50:46 P…	Local Settings\Software\Microsoft\Windows\CurrentVersion\TrayNotify	PastIconsStream	Length: 327,630
usrclass.dat	Min size	9/11/2014 9:50:46 P…	Local Settings\Software\Microsoft\Windows\CurrentVersion\TrayNotify	IconStreams	Length: 8,220
usrclass.dat	Min size	3/21/2014 3:13:31 P…	Local Settings\Software\Microsoft\Windows\Shell\BagMRU	8	Length: 10,343
usrclass.dat	Min size	12/4/2013 7:21:41 P…	Local Settings\Software\Microsoft\Windows\Shell\BagMRU\10	0	Length: 540
usrclass.dat	Min size	8/31/2013 12:36:51 …	Local Settings\Software\Microsoft\Windows\Shell\BagMRU\2\4	0	Length: 977
usrclass2.dat	Min size	4/4/2012 12:40:24 P…	Local Settings\MuiCache\4D\52C64B7E	@auditpolmsg.dll,-822	Length: 960

Hive path: C:\ProjectWorkingFolder\RegistryViewerZ\UsrClassDeletedBags.dat | Search hits: 672 | Search completed in 0.10007 seconds | Always on top □

Interacting with search results

Once a search is underway, results will show up in the Results grid at the bottom of the Find window.



Standard

Search for: mui

☐ Literal

Search in: ☑ Key name ☐ Value name ☐ Value data ☐ Value slack

Search type: ● Simple ○ Regular expression

Last write timestamp

Earliest (UTC):

Latest (UTC):

● Before ○ Between ○ After

Minimum value size

Minimum size (bytes): 512

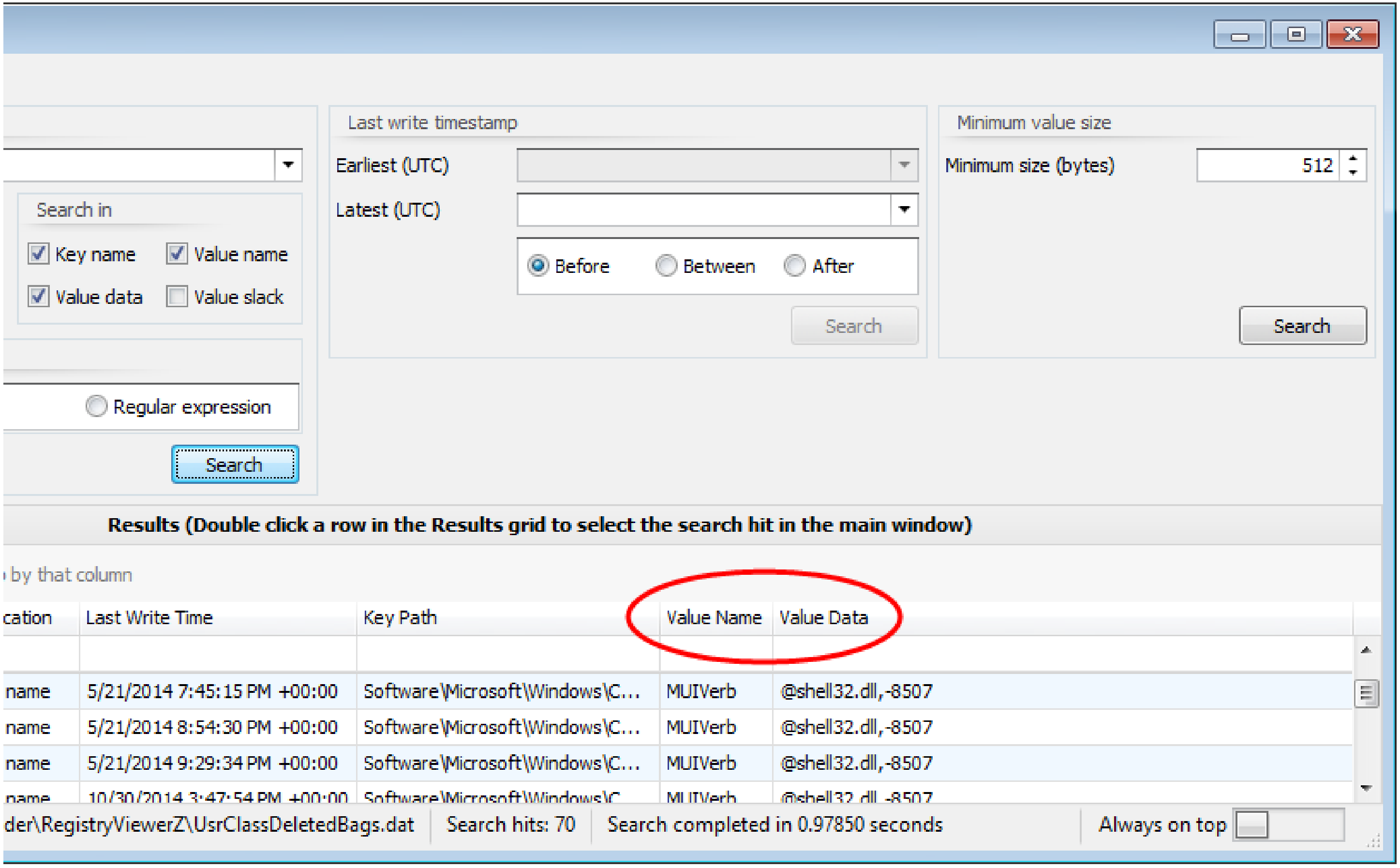
Results (Double click a row in the Results grid to select the search hit in the main window)

Hive Name	Hit Location	Last Write Time	Key Path
UsrClassDeletedBags.dat	Key name	9/19/2011 7:02:08 PM +00:00	Local Settings\MuiCache
usrclass.dat	Key name	3/19/2014 3:24:08 PM +00:00	Local Settings\MuiCache
usrclass.dat	Key name	3/19/2014 3:24:08 PM +00:00	Local Settings\Software\Microsoft\Windows\Shell\MuiCache
usrclass2.dat	Key name	3/2/2012 8:14:36 PM +00:00	Local Settings\MuiCache

Hive path: C:\ProjectWorkingFolder\RegistryViewerZ\UsrClassDeletedBags.dat | Search hits: 9 | Search completed in 0.02085 seconds

In the above example, a simple search was done for the string ‘mui’ which resulted in nine hits. The search results contains the hive the hit was found in, what type of hit it was (key name, value name, etc.), and other relevant information.

The columns shown in the Results grid will change depending on what kind of search was done. When searching in value name and/or value data, two additional columns will be shown as seen below.



Search in

☑ Key name ☑ Value name

☑ Value data □ Value slack

○ Regular expression

Last write timestamp

Earliest (UTC):

Latest (UTC):

● Before ○ Between ○ After

Minimum value size

Minimum size (bytes): 512

Results (Double click a row in the Results grid to select the search hit in the main window)

by that column

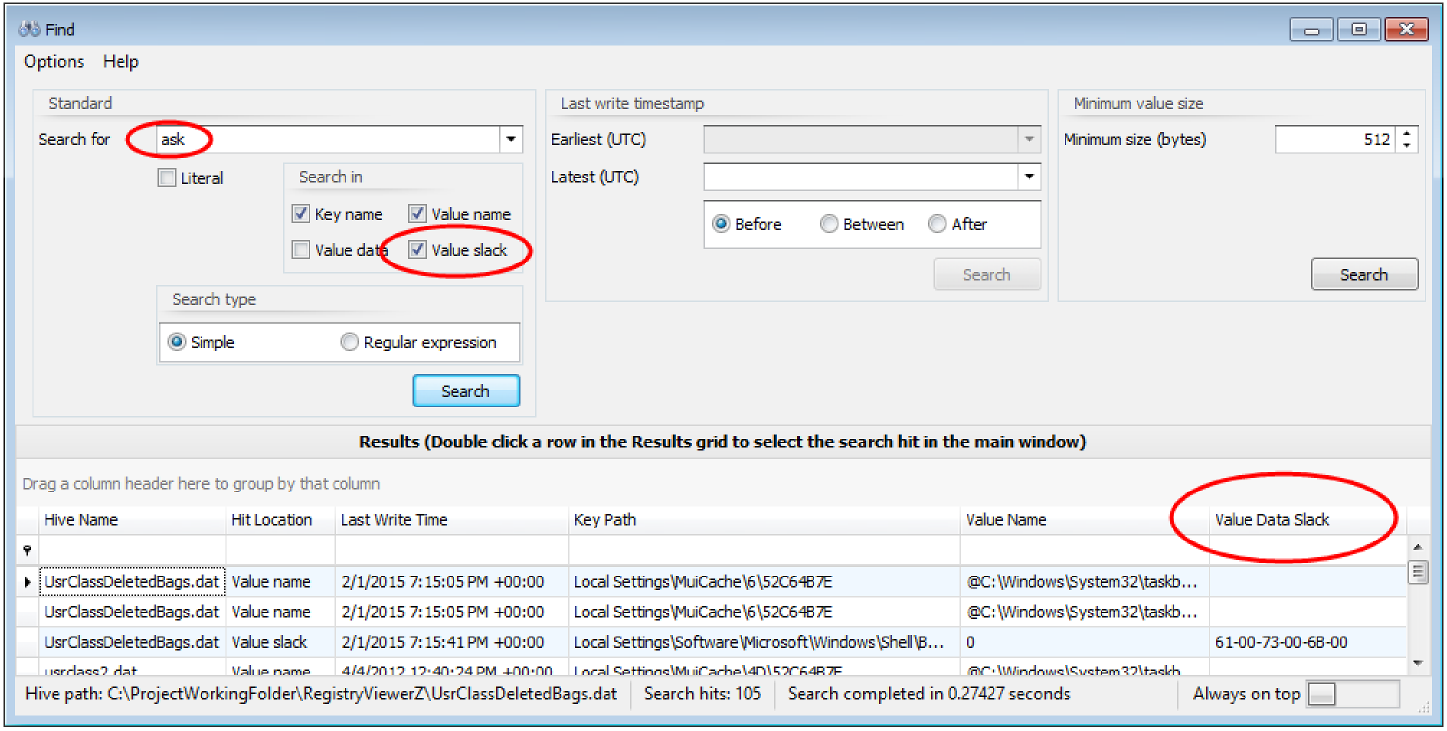
cation	Last Write Time	Key Path	Value Name	Value Data
name	5/21/2014 7:45:15 PM +00:00	Software\Microsoft\Windows\C...	MUIVerb	@shell32.dll,-8507
name	5/21/2014 8:54:30 PM +00:00	Software\Microsoft\Windows\C...	MUIVerb	@shell32.dll,-8507
name	5/21/2014 9:29:34 PM +00:00	Software\Microsoft\Windows\C...	MUIVerb	@shell32.dll,-8507
name	10/30/2014 3:47:54 PM +00:00	Software\Microsoft\Windows\C	MUIVerb	@shell32.dll,-8507

der\RegistryViewerZ\UsrClassDeletedBags.dat | Search hits: 70 | Search completed in 0.97850 seconds | Always on top □

When searching in value data and/or value slack, the Search for term will be found regardless of case or encoding (Western 1252 and/or Unicode to be exact). This makes it easy to find strings that have been encoded in binary data.

The way this works is to take the raw bytes that make up the value data and/or value slack and convert it to a string (again, in Western 1252 and Unicode), which is then searched using a regular expression. The regex will find the hit with exact capitalization, and the exact hit is then converted back to a byte string. This hit can then be reported back to the application and the data highlighted in context with the rest of the data, regardless of encoding or capitalization.

If the Literal checkbox is checked, the additional search against the converted data is not done behind the scenes. This allows you to look for specific byte patterns without Registry Explorer converting binary data to strings.


Find
Options
Help
Standard
Search in
Search type
Last write timestamp
Minimum value size
Search
Search
Search
Results (Double click a row in the Results grid to select the search hit in the main window)
Drag a column header here to group by that column

Search for: ask

□ Literal

Search in: ☑ Key name ☑ Value name □ Value data ☑ Value slack

Search type: ● Simple ○ Regular expression

Earliest (UTC):

Latest (UTC):

● Before ○ Between ○ After

Minimum size (bytes): 512

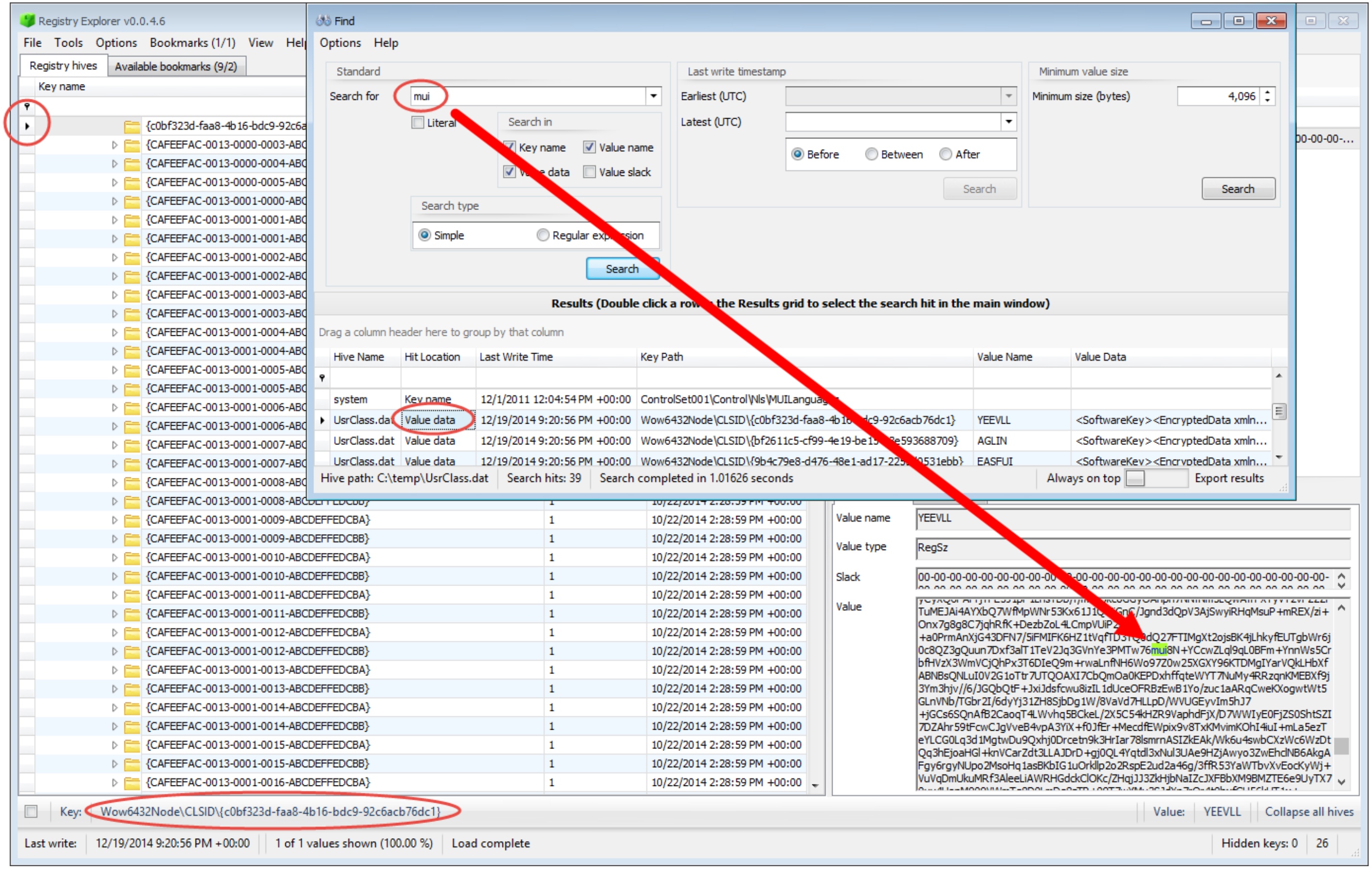
Hive Name	Hit Location	Last Write Time	Key Path	Value Name	Value Data Slack
UsrClassDeletedBags.dat	Value name	2/1/2015 7:15:05 PM +00:00	Local Settings\MuiCache\6\52C64B7E	@C:\Windows\System32\taskb…	
UsrClassDeletedBags.dat	Value name	2/1/2015 7:15:05 PM +00:00	Local Settings\MuiCache\6\52C64B7E	@C:\Windows\System32\taskb…	
UsrClassDeletedBags.dat	Value slack	2/1/2015 7:15:41 PM +00:00	Local Settings\Software\Microsoft\Windows\Shell\B…	0	61-00-73-00-6B-00
usrclass2.dat	Value name	4/4/2012 12:40:24 PM +00:00	Local Settings\MuiCache\4D\52C64B7E	@C:\Windows\System32\taskb	

Hive path: C:\ProjectWorkingFolder\RegistryViewerZ\UsrClassDeletedBags.dat | Search hits: 105 | Search completed in 0.27427 seconds | Always on top □

In the screen shot above, notice the hit in value slack was found Unicode encoded (61-00-73-6B-00)

Viewing search results

To view the search hit in the main Registry Explorer window, simply double click on the result you wish to view. The Registry hive containing the hit will be selected along with the key where the hit was found. If the Hit location is in a value name or in value data, the corresponding value will be selected under the key.



Search for: mui

☐ Literal

Search in: ☑ Key name ☑ Value name ☑ Value data ☐ Value slack

Search type: ● Simple ○ Regular expression

Earliest (UTC):

Latest (UTC):

● Before ○ Between ○ After

Minimum size (bytes): 4,096

Results (Double click a row in the Results grid to select the search hit in the main window)

Hive Name	Hit Location	Last Write Time	Key Path	Value Name	Value Data
system	Key name	12/1/2011 12:04:54 PM +00:00	ControlSet001\Control\Nls\MUILangua		
UsrClass.dat	Value data	12/19/2014 9:20:56 PM +00:00	Wow6432Node\CLSID\{c0bf323d-faa8-4b16 dc9-92c6acb76dc1}	YEEVLL	<SoftwareKey><EncryptedData xmln...
UsrClass.dat	Value data	12/19/2014 9:20:56 PM +00:00	Wow6432Node\CLSID\{bf2611c5-cf99-4e19-be1 e593688709}	AGLIN	<SoftwareKey><EncryptedData xmln...
UsrClass.dat	Value data	12/19/2014 9:20:56 PM +00:00	Wow6432Node\CLSID\{9b4c79e8-d476-48e1-ad17-225 0531ebb}	EASFUI	<SoftwareKey><EncryptedData xmln...

Hive path: C:\temp\UsrClass.dat | Search hits: 39 | Search completed in 1.01626 seconds

Key name		
{c0bf323d-faa8-4b16-bdc9-92c6a		
{CAFEFAC-0013-0000-0003-ABC		
{CAFEFAC-0013-0000-0004-ABC		
{CAFEFAC-0013-0000-0005-ABC		
{CAFEFAC-0013-0001-0000-ABC		
{CAFEFAC-0013-0001-0001-ABC		
{CAFEFAC-0013-0001-0001-ABC		
{CAFEFAC-0013-0001-0002-ABC		
{CAFEFAC-0013-0001-0002-ABC		
{CAFEFAC-0013-0001-0003-ABC		
{CAFEFAC-0013-0001-0003-ABC		
{CAFEFAC-0013-0001-0004-ABC		
{CAFEFAC-0013-0001-0004-ABC		
{CAFEFAC-0013-0001-0005-ABC		
{CAFEFAC-0013-0001-0005-ABC		
{CAFEFAC-0013-0001-0006-ABC		
{CAFEFAC-0013-0001-0006-ABC		
{CAFEFAC-0013-0001-0007-ABC		
{CAFEFAC-0013-0001-0007-ABC		
{CAFEFAC-0013-0001-0008-ABC		
{CAFEFAC-0013-0001-0008-ABCDEFFEDCBB}	1	10/22/2014 2:28:59 PM +00:00
{CAFEFAC-0013-0001-0009-ABCDEFFEDCBA}	1	10/22/2014 2:28:59 PM +00:00
{CAFEFAC-0013-0001-0009-ABCDEFFEDCBB}	1	10/22/2014 2:28:59 PM +00:00
{CAFEFAC-0013-0001-0010-ABCDEFFEDCBA}	1	10/22/2014 2:28:59 PM +00:00
{CAFEFAC-0013-0001-0010-ABCDEFFEDCBB}	1	10/22/2014 2:28:59 PM +00:00
{CAFEFAC-0013-0001-0011-ABCDEFFEDCBA}	1	10/22/2014 2:28:59 PM +00:00
{CAFEFAC-0013-0001-0011-ABCDEFFEDCBB}	1	10/22/2014 2:28:59 PM +00:00
{CAFEFAC-0013-0001-0012-ABCDEFFEDCBA}	1	10/22/2014 2:28:59 PM +00:00
{CAFEFAC-0013-0001-0012-ABCDEFFEDCBB}	1	10/22/2014 2:28:59 PM +00:00
{CAFEFAC-0013-0001-0013-ABCDEFFEDCBA}	1	10/22/2014 2:28:59 PM +00:00
{CAFEFAC-0013-0001-0013-ABCDEFFEDCBB}	1	10/22/2014 2:28:59 PM +00:00
{CAFEFAC-0013-0001-0014-ABCDEFFEDCBA}	1	10/22/2014 2:28:59 PM +00:00
{CAFEFAC-0013-0001-0014-ABCDEFFEDCBB}	1	10/22/2014 2:28:59 PM +00:00
{CAFEFAC-0013-0001-0015-ABCDEFFEDCBA}	1	10/22/2014 2:28:59 PM +00:00
{CAFEFAC-0013-0001-0015-ABCDEFFEDCBB}	1	10/22/2014 2:28:59 PM +00:00
{CAFEFAC-0013-0001-0016-ABCDEFFEDCBA}	1	10/22/2014 2:28:59 PM +00:00

Value name: YEEVLL

Value type: RegSz

Slack: 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-

Value:

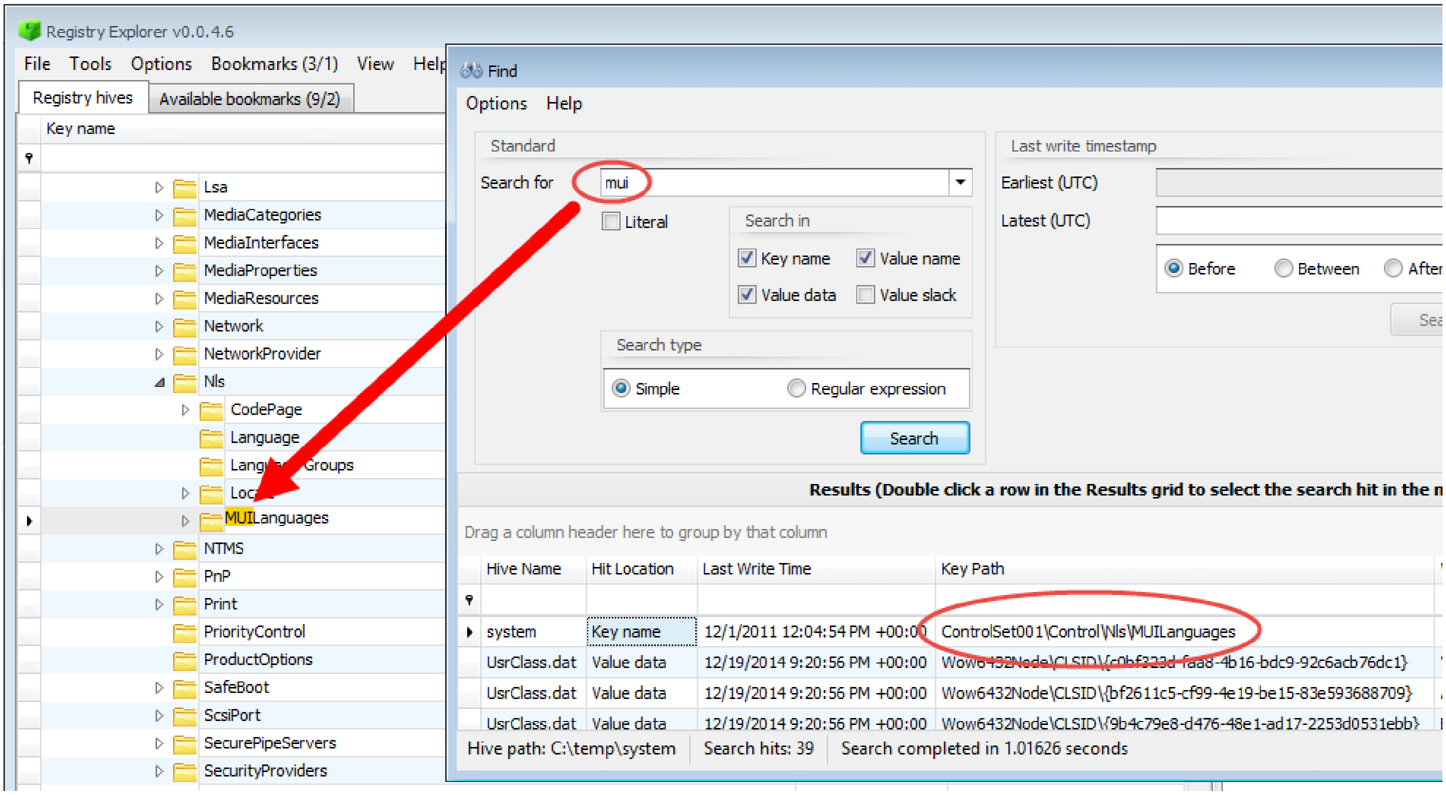
```
TuMEJAi4AYXbQ7WfMpWNr53Kx61J1Q GnC/Jgnd3dQpV3AjSwyiRHqMsuP+mREX/zi+
Onx7g8g8C7jqhRfK+DezbZoL4LCmpVUiP2
+a0PrmAnXjG43DFN7/5iFMIFK6HZ1tVqfTD3 dQ27FTIMgXt2ojsBK4jLhkyfEUTgbWr6j
0c8QZ3gQuun7Dxf3alT1TeV2Jq3GVnYe3PMTw76mui8N+YCcwZLql9qL0BFm+YnnWs5Cr
bfHVzX3WmVCjQhPx3T6DIeQ9m+rwaLnfNH6Wo97Z0w25XGXY96KTDMgIYarVQkLHbXf
ABNBsQNLuI0V2G1oTtr7UTQOAXI7CbQmOa0KEPDxhffqteWYT7NuMy4RRzqnKMEBXf9j
3Ym3hjv//6/JGQbQtF+JxiJdsfcwu8izIL1dUceOFRBzEwB1Yo/zuc1aARqCweKXogwtWt5
GLnVNb/TGbr2I/6dyYj31ZH8SjbDg1W/8VaVd7HLLpD/WVUGEyvIm5hJ7
+jGCs6SQnAfB2CaoqT4LWvhq5BCkeL/2X5C54kHZR9VaphdFjX/D7WWIyE0FjZS0ShtSZI
7DZAhr59tFcwCJgVveB4vpA3YiX+f0JfEr+MecdfEWpix9v8TxKMvimKOhI4iuI+mLa5ezT
eYLCG0Lq3d1MgtwDu9Qxhj0Drcetn9k3HrIar78lsmrnASIZkEAk/Wk6u4swbCXzWc6WzDt
Qq3hEjoaHGl+knVCarZdt3LLAJDrD+gj0QL4Yqtdl3xNul3UAe9HZjAwyo3ZwEhdNB6AkgA
Fgy6rgyNUpo2MsoHq1asBKbIG1uOrkllp2o2RspE2ud2a46g/3ffR53YaWTbvXvEocKyWj+
VuVqDmUkuMRf3AleeLiAWRHGdckClOKc/ZHqjJJ3ZkHjbNaIZcJXFBbXM9BMZTE6e9UyTX7
```

Key: Wow6432Node\CLSID\{c0bf323d-faa8-4b16-bdc9-92c6acb76dc1} | Value: YEEVLL

Last write: 12/19/2014 9:20:56 PM +00:00 | 1 of 1 values shown (100.00 %) | Load complete | Hidden keys: 0 | 26

For all simple searches, the search hit will be highlighted (or, in the case of a RegBinary hit, the bytes that make up the hit will be selected). A few more examples of this are shown below.

For key name hits, the matching part of the key name is highlighted.



Search for: mui

☐ Literal

Search in: ☑ Key name ☑ Value name ☑ Value data ☐ Value slack

Search type: ● Simple ○ Regular expression

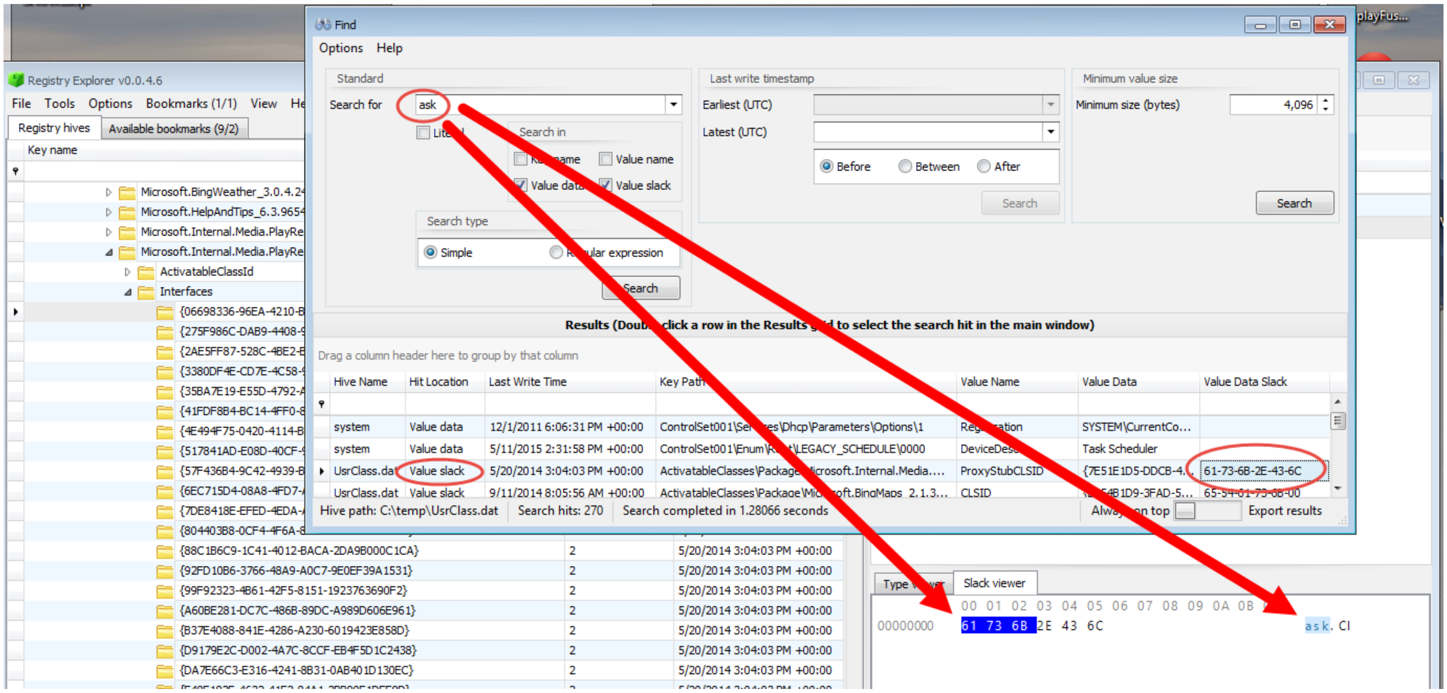
Results (Double click a row in the Results grid to select the search hit in the m

Drag a column header here to group by that column

Hive Name	Hit Location	Last Write Time	Key Path
system	Key name	12/1/2011 12:04:54 PM +00:00	ControlSet001\Control\Nls\MUILanguages
UsrClass.dat	Value data	12/19/2014 9:20:56 PM +00:00	Wow6432Node\CLSID\{c0bf323d-faa8-4b16-bdc9-92c6acb76dc1}
UsrClass.dat	Value data	12/19/2014 9:20:56 PM +00:00	Wow6432Node\CLSID\{bf2611c5-cf99-4e19-be15-83e593688709}
UsrClass.dat	Value data	12/19/2014 9:20:56 PM +00:00	Wow6432Node\CLSID\{9b4c79e8-d476-48e1-ad17-2253d0531ebb}

Hive path: C:\temp\system | Search hits: 39 | Search completed in 1.01626 seconds

For value data (when the value type is RegBinary) and value slack, the bytes that make up the search hit are selected in the hex viewer.


playFus...
Registry Explorer v0.0.4.6
File Tools Options Bookmarks (1/1) View He
Registry hives
Available bookmarks (9/2)
Key name
Microsoft.BingWeather_3.0.4.24
Microsoft.HelpAndTips_6.3.9654
Microsoft.Internal.Media.PlayRe
Microsoft.Internal.Media.PlayRe
ActivatableClassId
Interfaces
{06698336-96EA-4210-B
{275F986C-DAB9-4408-9
{2AE5FF87-528C-4BE2-B
{3380DF4E-CD7E-4C58-9
{35BA7E19-E55D-4792-A
{41FDF8B4-BC14-4FF0-8
{4E494F75-0420-4114-B
{517841AD-E08D-40CF-9
{57F436B4-9C42-4939-B
{6EC715D4-08A8-4FD7-A
{7DE8418E-EFED-4EDA-A
{804403B8-0CF4-4F6A-8
{88C1B6C9-1C41-4012-BACA-2DA9B000C1CA} 2 5/20/2014 3:04:03 PM +00:00
{92FD10B6-3766-48A9-A0C7-9E0EF39A1531} 2 5/20/2014 3:04:03 PM +00:00
{99F92323-4B61-42F5-8151-1923763690F2} 2 5/20/2014 3:04:03 PM +00:00
{A60BE281-DC7C-486B-89DC-A989D606E961} 2 5/20/2014 3:04:03 PM +00:00
{B37E4088-841E-4286-A230-6019423E858D} 2 5/20/2014 3:04:03 PM +00:00
{D9179E2C-D002-4A7C-8CCF-EB4F5D1C2438} 2 5/20/2014 3:04:03 PM +00:00
{DA7E66C3-E316-4241-8B31-0AB401D130EC} 2 5/20/2014 3:04:03 PM +00:00
Find
Options Help
Standard
Last write timestamp
Earliest (UTC)
Latest (UTC)
● Before ○ Between ○ After
Search
Minimum value size
Search
Always on top
Export results
Type viewer
Slack viewer

Search for: ask

☐ Literal

Search in: ☐ Key name ☐ Value name ☑ Value data ☑ Value slack

Search type: ● Simple ○ Regular expression

Minimum size (bytes): 4,096

Results (Double click a row in the Results grid to select the search hit in the main window)

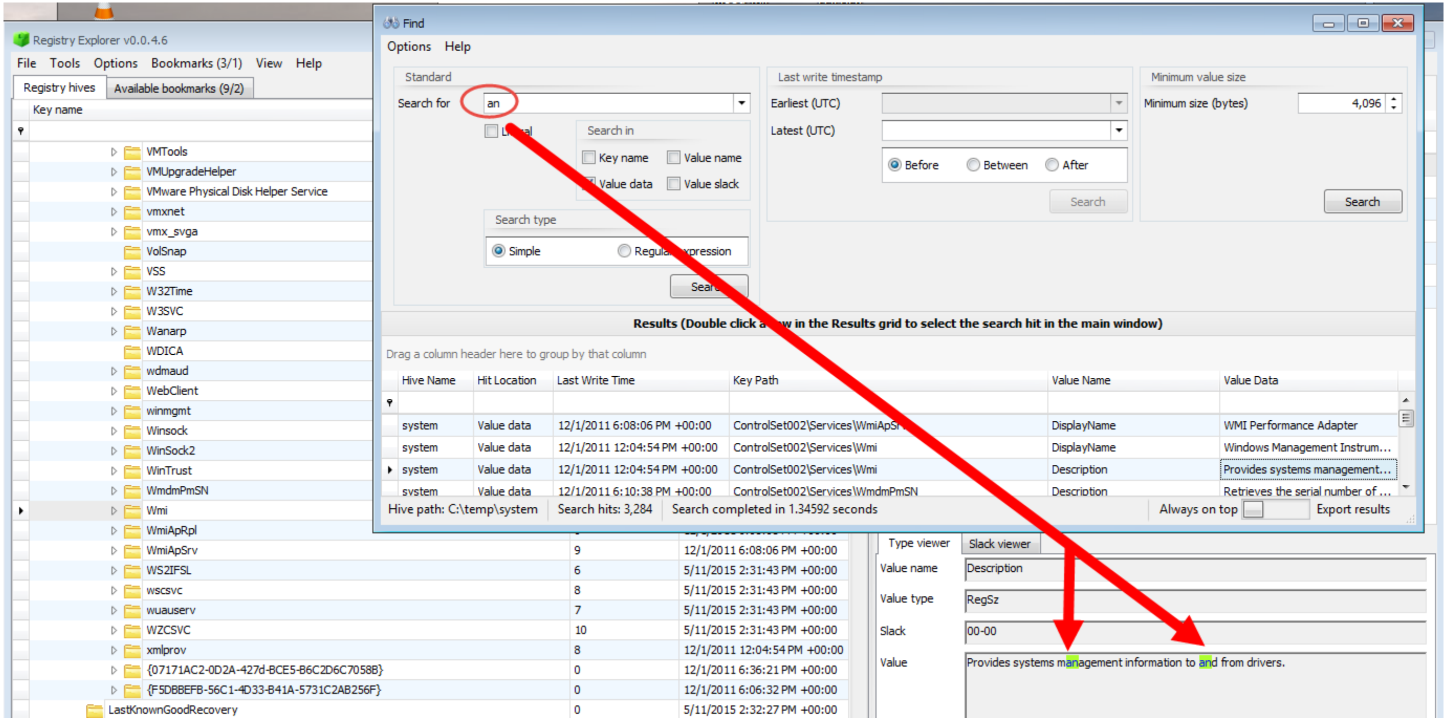
Drag a column header here to group by that column

Hive Name	Hit Location	Last Write Time	Key Path	Value Name	Value Data	Value Data Slack
system	Value data	12/1/2011 6:06:31 PM +00:00	ControlSet001\Services\Dhcp\Parameters\Options\1	Registration	SYSTEM\CurrentCo...	
system	Value data	5/11/2015 2:31:58 PM +00:00	ControlSet001\Enum\Root\LEGACY_SCHEDULE\0000	DeviceDesc	Task Scheduler	
UsrClass.dat	Value slack	5/20/2014 3:04:03 PM +00:00	ActivatableClasses\Package\Microsoft.Internal.Media....	ProxyStubCLSID	{7E51E1D5-DDCB-4...	61-73-6B-2E-43-6C
UsrClass.dat	Value slack	9/11/2014 8:05:56 AM +00:00	ActivatableClasses\Package\Microsoft.BingMaps_2.1.3...	CLSID	{[illegible]4B1D9-3FAD-5...	65-54-01-73-6B-00

Hive path: C:\temp\UsrClass.dat | Search hits: 270 | Search completed in 1.28066 seconds

```
          00 01 02 03 04 05 06 07 08 09 0A 0B
00000000  61 73 6B 2E 43 6C                      ask.Cl
```

For value name and non-RegBinary value data hits, all instances of the search term are highlighted.



Search for: an

Search in: ☐ Key name ☐ Value name ☑ Value data ☐ Value slack

Search type: ● Simple ○ Regular expression

○ Before ○ Between ○ After

Minimum size (bytes): 4,096

Hive Name	Hit Location	Last Write Time	Key Path	Value Name	Value Data
system	Value data	12/1/2011 6:08:06 PM +00:00	ControlSet002\Services\WmiApSrv	DisplayName	WMI Performance Adapter
system	Value data	12/1/2011 12:04:54 PM +00:00	ControlSet002\Services\Wmi	DisplayName	Windows Management Instrum...
system	Value data	12/1/2011 12:04:54 PM +00:00	ControlSet002\Services\Wmi	Description	Provides systems management...
system	Value data	12/1/2011 6:10:38 PM +00:00	ControlSet002\Services\WmdmPmSN	Description	Retrieves the serial number of ...

Hive path: C:\temp\system | Search hits: 3,284 | Search completed in 1.34592 seconds

VMTools
VMUpgradeHelper
VMware Physical Disk Helper Service
vmxnet
vmx_svga
VolSnap
VSS
W32Time
W3SVC
Wanarp
WDICA
wdmaud
WebClient
winmgmt
Winsock
WinSock2
WinTrust
WmdmPmSN
Wmi

WmiApRpl		
WmiApSrv	9	12/1/2011 6:08:06 PM +00:00
WS2IFSL	6	5/11/2015 2:31:43 PM +00:00
wscsvc	8	5/11/2015 2:31:43 PM +00:00
wuauserv	7	5/11/2015 2:31:43 PM +00:00
WZCSVC	10	5/11/2015 2:31:43 PM +00:00
xmlprov	8	12/1/2011 12:04:54 PM +00:00
{07171AC2-0D2A-427d-BCE5-B6C2D6C7058B}	0	12/1/2011 6:36:21 PM +00:00
{F5DBBEFB-56C1-4D33-B41A-5731C2AB256F}	0	12/1/2011 6:06:32 PM +00:00
LastKnownGoodRecovery	0	5/11/2015 2:32:27 PM +00:00

Value name: Description
Value type: RegSz
Slack: 00-00
Value: Provides systems management information to and from drivers.

Search tips

The fastest searches are against key names. Searching against value names will be slower than key names only. Searching value data/value slack is slower still. This is because every value of every key has to be looked at in order to search for value names or value data/slack across all loaded hives.

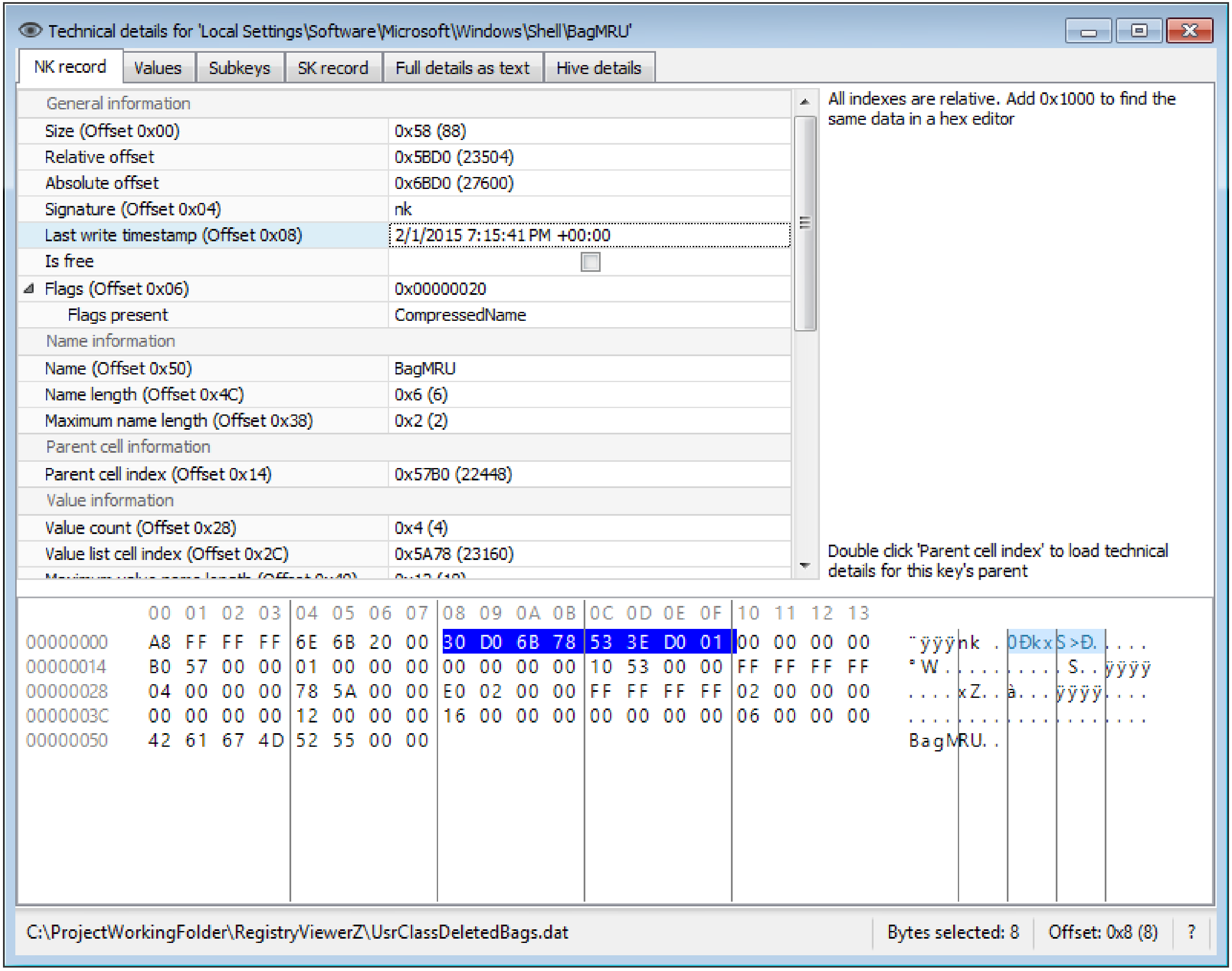
Do not let this stop you from searching against value names and data however. Even with these options selected, Registry Explorer can still search multiple hives very quickly (often under a second), but this depends on the number of keys and values in the loaded hives.

You can export the search results to Excel via the button in the lower right.

Technical details in depth

One unique feature of Registry Explorer is the ability to view the technical details of any key, its values, security information, etc. This feature bridges the gap between a hex editor and other viewers in that Registry Explorer can be used to validate itself as to its interpretation of Registry data.

To view the technical details of a key, select the key you are interested in, then right click and select ‘Technical details’ from the context menu. **F5** can also be used as a shortcut.



Technical details for 'Local Settings\Software\Microsoft\Windows\Shell\BagMRU'
NK record
Values
Subkeys
SK record
Full details as text
Hive details

General information	
Size (Offset 0x00)	0x58 (88)
Relative offset	0x5BD0 (23504)
Absolute offset	0x6BD0 (27600)
Signature (Offset 0x04)	nk
Last write timestamp (Offset 0x08)	2/1/2015 7:15:41 PM +00:00
Is free	□
Flags (Offset 0x06)	0x00000020
Flags present	CompressedName
Name information	
Name (Offset 0x50)	BagMRU
Name length (Offset 0x4C)	0x6 (6)
Maximum name length (Offset 0x38)	0x2 (2)
Parent cell information	
Parent cell index (Offset 0x14)	0x57B0 (22448)
Value information	
Value count (Offset 0x28)	0x4 (4)
Value list cell index (Offset 0x2C)	0x5A78 (23160)
Maximum value name length (Offset 0x40)	0x12 (18)

All indexes are relative. Add 0x1000 to find the same data in a hex editor

Double click 'Parent cell index' to load technical details for this key's parent

```
          00 01 02 03|04 05 06 07|08 09 0A 0B|0C 0D 0E 0F|10 11 12 13
00000000  A8 FF FF FF|6E 6B 20 00|30 D0 6B 78|53 3E D0 01|00 00 00 00   ¨ÿÿÿnk . 0ÐkxS>Ð. . . . .
00000014  B0 57 00 00|01 00 00 00|00 00 00 00|10 53 00 00|FF FF FF FF   °W. . . . . . . . . . . S. . ÿÿÿÿ
00000028  04 00 00 00|78 5A 00 00|E0 02 00 00|FF FF FF FF|02 00 00 00   . . . . xZ. . à. . . ÿÿÿÿ. . . .
0000003C  00 00 00 00|12 00 00 00|16 00 00 00|00 00 00 00|06 00 00 00   . . . . . . . . . . . . . . . . . . . .
00000050  42 61 67 4D|52 55 00 00                                         BagMRU. .
```

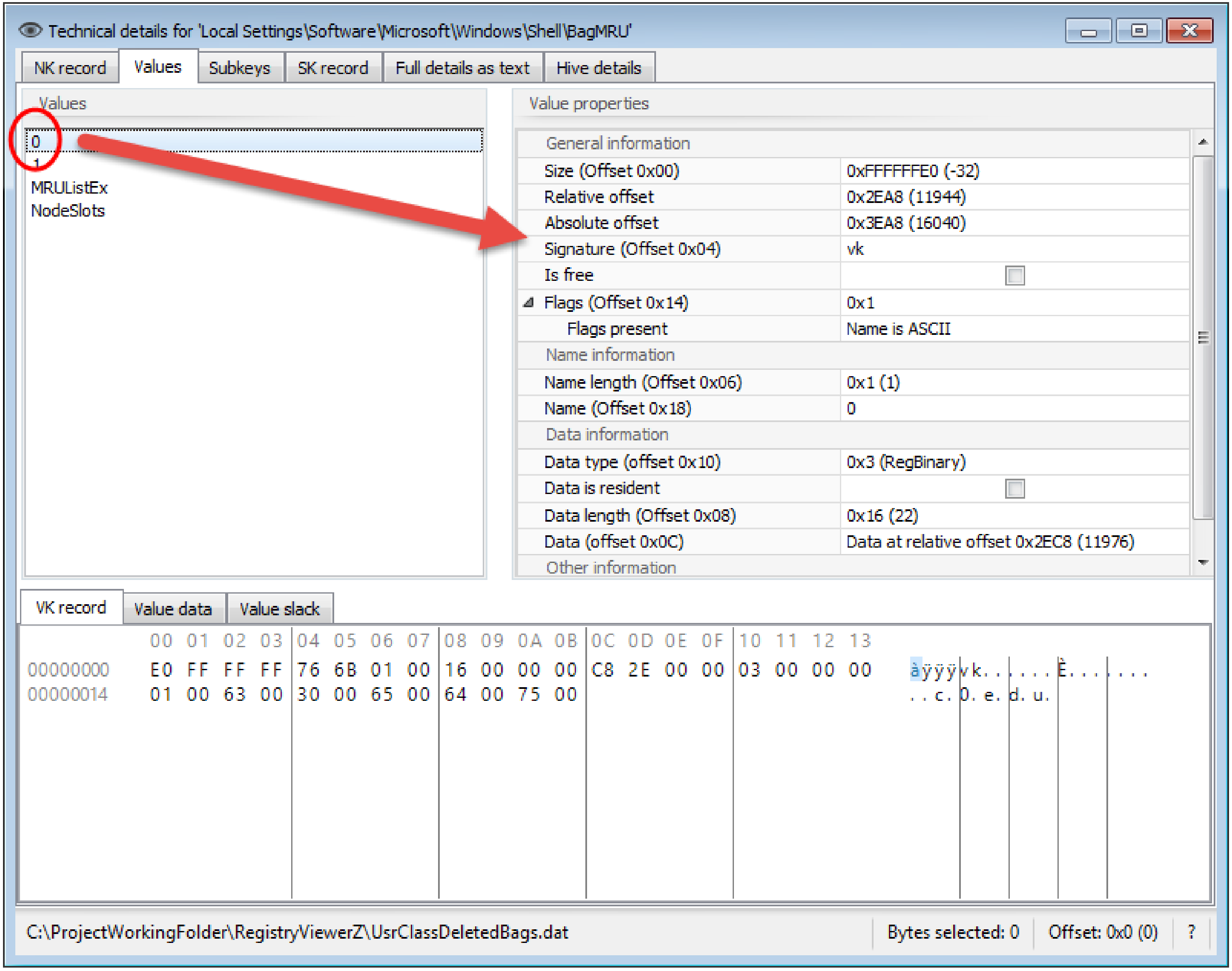
C:\ProjectWorkingFolder\RegistryViewerZ\UsrClassDeletedBags.dat | Bytes selected: 8 | Offset: 0x8 (8) | ?

In the example above, the Technical details for the 'Local Settings\Software\Microsoft\Windows\Shell\BagMRU' key are shown. The bytes at the bottom of the details form are the bytes for the NK record as they are found in the Registry hive as viewed in a hex editor.

As different properties are selected, the highlighted bytes change to reflect the location in the raw data where that property lives. The Last write timestamp property is selected, as are the bytes that this property is derived from.

The selected bytes can be copied via **Ctrl+C.** Hold **Ctrl+Alt+C** to copy both the property name and the value to the clipboard.

If a key contains one or more values, the Values tab is visible and contains a list of all the key's values. Selecting a value will display the VK record's properties and raw data as we saw with the NK record above.

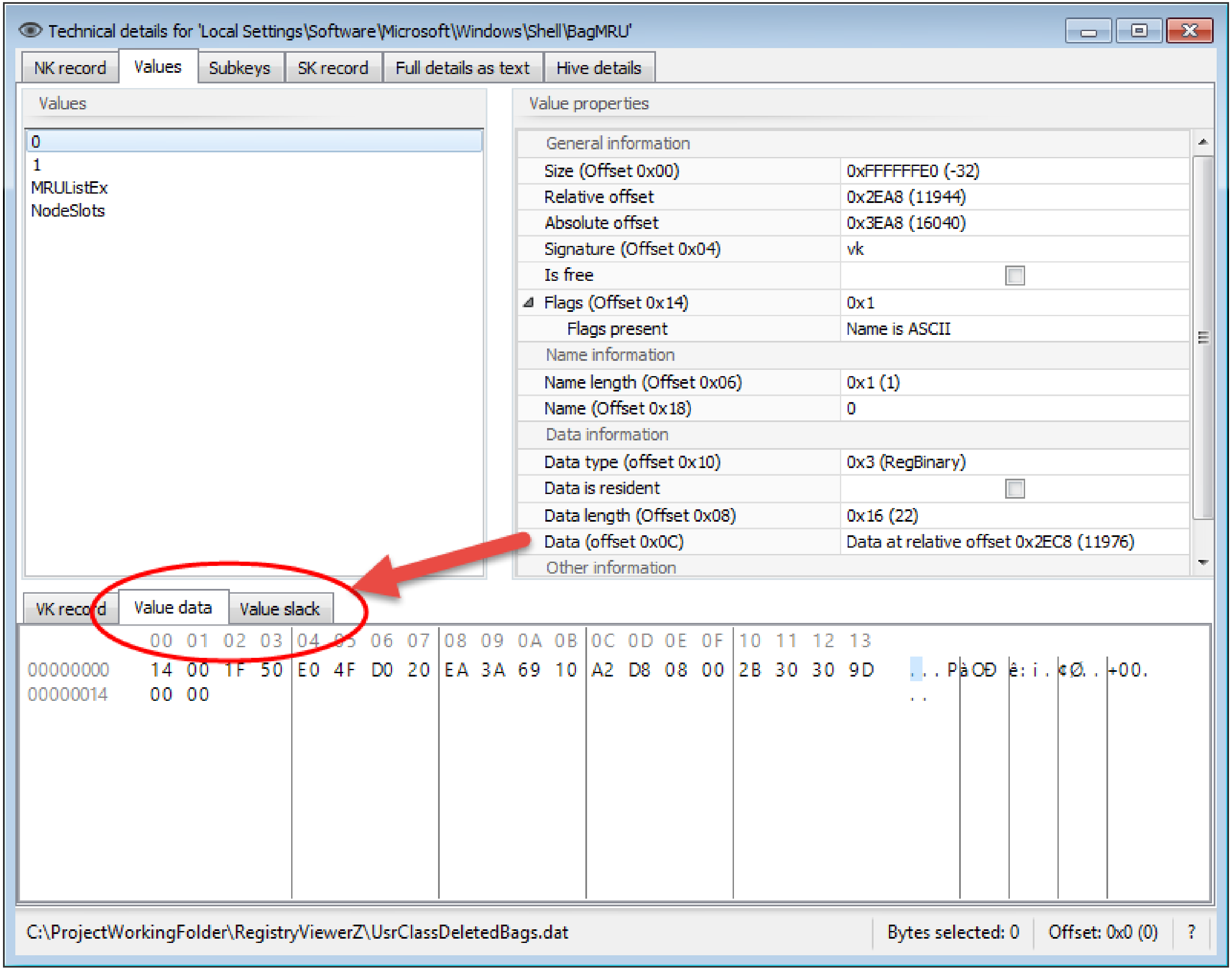


General information	
Size (Offset 0x00)	0xFFFFFFE0 (-32)
Relative offset	0x2EA8 (11944)
Absolute offset	0x3EA8 (16040)
Signature (Offset 0x04)	vk
Is free	□
Flags (Offset 0x14)	0x1
Flags present	Name is ASCII
Name information	
Name length (Offset 0x06)	0x1 (1)
Name (Offset 0x18)	0
Data information	
Data type (offset 0x10)	0x3 (RegBinary)
Data is resident	□
Data length (Offset 0x08)	0x16 (22)
Data (offset 0x0C)	Data at relative offset 0x2EC8 (11976)
Other information	

```
          00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13
00000000  E0 FF FF FF 76 6B 01 00 16 00 00 00 C8 2E 00 00 03 00 00 00   àÿÿÿvk......È.......
00000014  01 00 63 00 30 00 65 00 64 00 75 00                            ..c.0.e.d.u.
```

C:\ProjectWorkingFolder\RegistryViewerZ\UsrClassDeletedBags.dat | Bytes selected: 0 | Offset: 0x0 (0) | ?

At the bottom of the Values tab, the raw VK record is shown. A hex viewer for the value data and value slack (if the value has slack) is also shown. This allows you to see both the VK records and the data in one place. The value data/slack is the data that is available at the Data offset.



Values

- 0
- 1
- MRUListEx
- NodeSlots

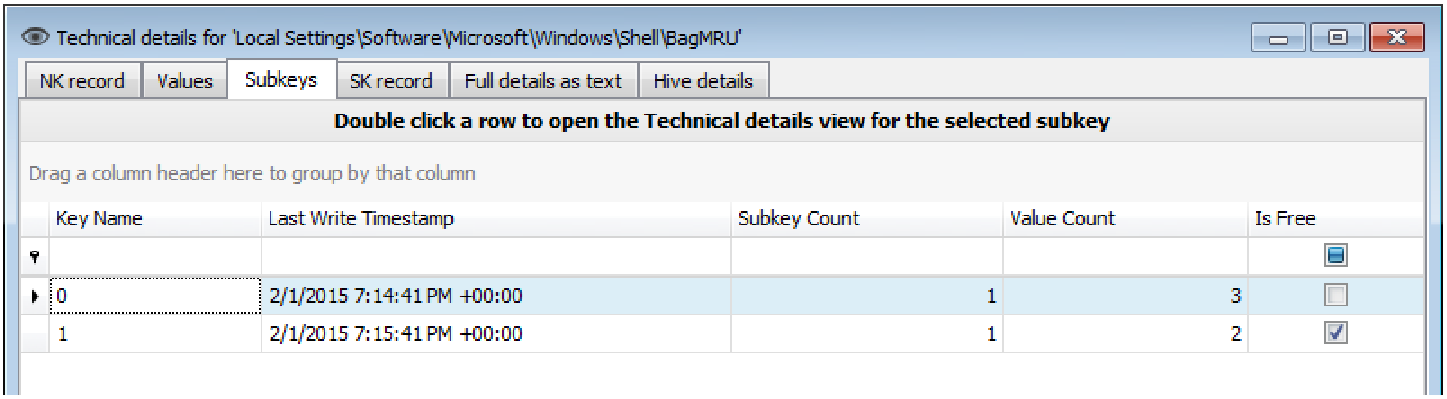
Value properties

General information	
Size (Offset 0x00)	0xFFFFFFE0 (-32)
Relative offset	0x2EA8 (11944)
Absolute offset	0x3EA8 (16040)
Signature (Offset 0x04)	vk
Is free	☐
Flags (Offset 0x14)	0x1
Flags present	Name is ASCII
Name information	
Name length (Offset 0x06)	0x1 (1)
Name (Offset 0x18)	0
Data information	
Data type (offset 0x10)	0x3 (RegBinary)
Data is resident	☐
Data length (Offset 0x08)	0x16 (22)
Data (offset 0x0C)	Data at relative offset 0x2EC8 (11976)
Other information	

```
          00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13
00000000  14 00 1F 50 E0 4F D0 20 EA 3A 69 10 A2 D8 08 00 2B 30 30 9D   ...PàOÐ ê:i.¢Ø..+00.
00000014  00 00                                                          ..
```

C:\ProjectWorkingFolder\RegistryViewerZ\UsrClassDeletedBags.dat   Bytes selected: 0   Offset: 0x0 (0)   ?

If a key contains subkeys, the Subkeys tab is visible and contains a list of the current key's subkeys. Double clicking on a subkey will open the Technical details report for that key in its own window.

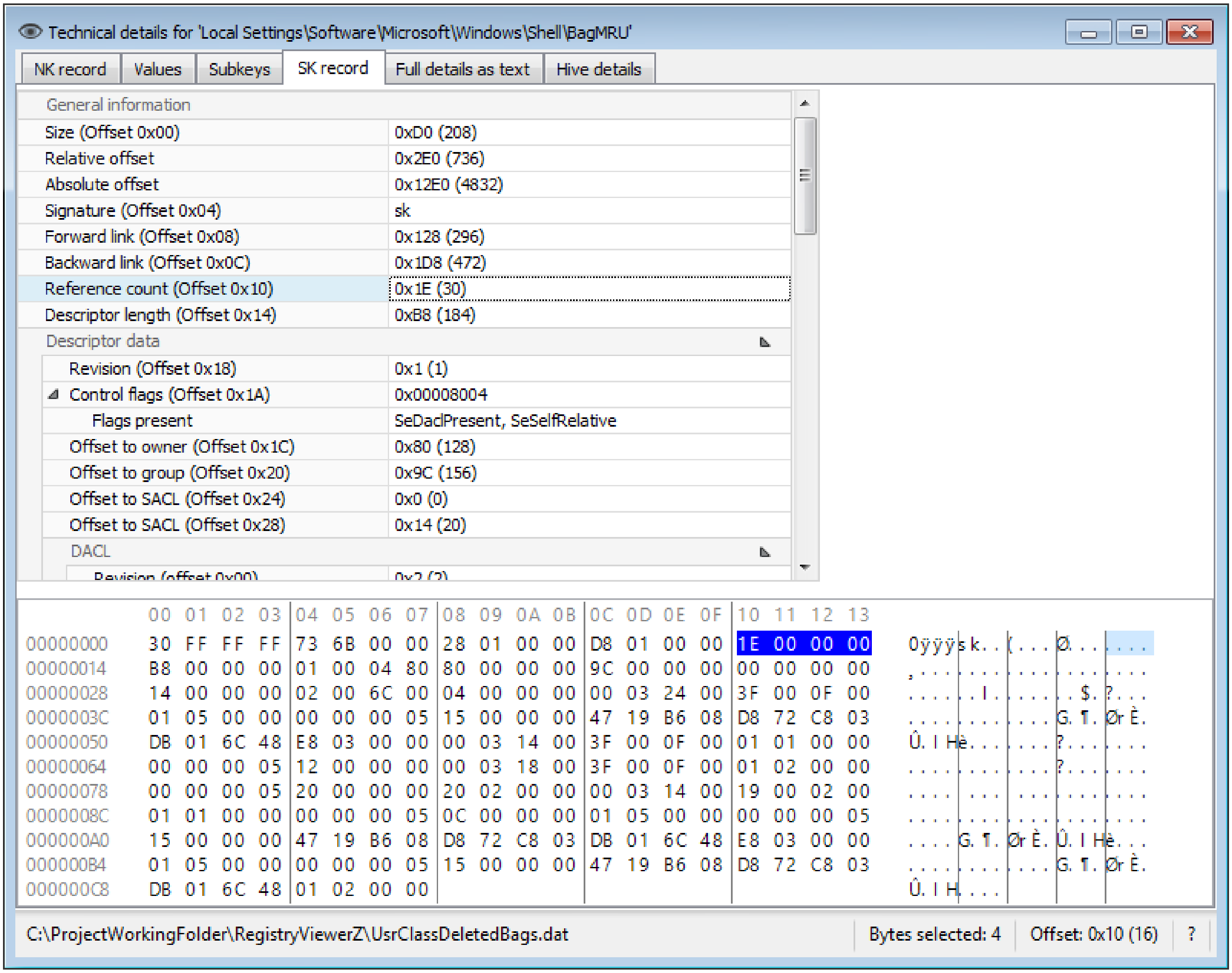

Technical details for 'Local Settings\Software\Microsoft\Windows\Shell\BagMRU'
NK record
Values
Subkeys
SK record
Full details as text
Hive details

Double click a row to open the Technical details view for the selected subkey

Drag a column header here to group by that column

Key Name	Last Write Timestamp	Subkey Count	Value Count	Is Free
0	2/1/2015 7:14:41 PM +00:00	1	3	☐
1	2/1/2015 7:15:41 PM +00:00	1	2	☑

Keys found in the active Registry (in other words, not deleted) will have an SK tab that contains the security key information for the NK record. The SK tab works the same as the NK and VK tab in that the hex editor updates when properties are selected, etc.


Technical details for 'Local Settings\Software\Microsoft\Windows\Shell\BagMRU'
NK record
Values
Subkeys
SK record
Full details as text
Hive details

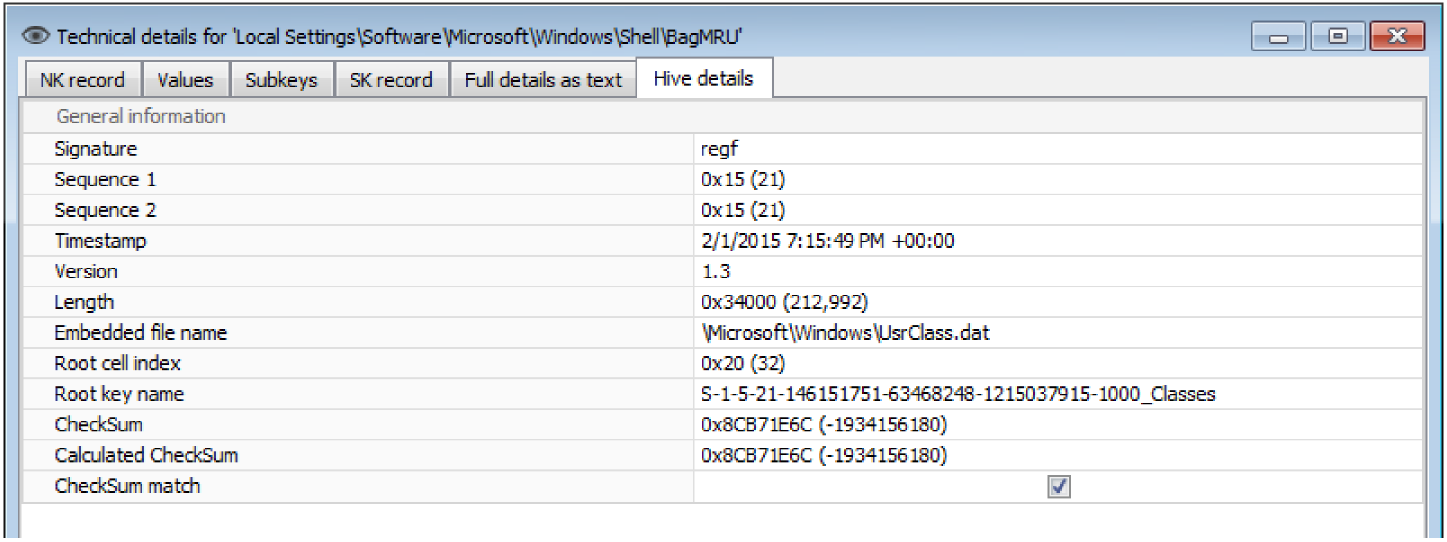
General information	
Size (Offset 0x00)	0xD0 (208)
Relative offset	0x2E0 (736)
Absolute offset	0x12E0 (4832)
Signature (Offset 0x04)	sk
Forward link (Offset 0x08)	0x128 (296)
Backward link (Offset 0x0C)	0x1D8 (472)
Reference count (Offset 0x10)	0x1E (30)
Descriptor length (Offset 0x14)	0xB8 (184)
Descriptor data	
Revision (Offset 0x18)	0x1 (1)
Control flags (Offset 0x1A)	0x00008004
Flags present	SeDaclPresent, SeSelfRelative
Offset to owner (Offset 0x1C)	0x80 (128)
Offset to group (Offset 0x20)	0x9C (156)
Offset to SACL (Offset 0x24)	0x0 (0)
Offset to SACL (Offset 0x28)	0x14 (20)
DACL	
Revision (offset 0x00)	0x2 (2)

```
          00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13
00000000  30 FF FF FF 73 6B 00 00 28 01 00 00 D8 01 00 00 1E 00 00 00   0ÿÿÿsk..(...Ø.......
00000014  B8 00 00 00 01 00 04 80 80 00 00 00 9C 00 00 00 00 00 00 00   ¸...................
00000028  14 00 00 00 02 00 6C 00 04 00 00 00 00 03 24 00 3F 00 0F 00   ......l.......$.?...
0000003C  01 05 00 00 00 00 00 05 15 00 00 00 47 19 B6 08 D8 72 C8 03   ............G.¶.ØrÈ.
00000050  DB 01 6C 48 E8 03 00 00 00 03 14 00 3F 00 0F 00 01 01 00 00   Û.lHè.......?.......
00000064  00 00 00 05 12 00 00 00 00 03 18 00 3F 00 0F 00 01 02 00 00   ............?.......
00000078  00 00 00 05 20 00 00 00 20 02 00 00 00 03 14 00 19 00 02 00   .... ... ...........
0000008C  01 01 00 00 00 00 00 05 0C 00 00 00 01 05 00 00 00 00 00 05   ....................
000000A0  15 00 00 00 47 19 B6 08 D8 72 C8 03 DB 01 6C 48 E8 03 00 00   ....G.¶.ØrÈ.Û.lHè...
000000B4  01 05 00 00 00 00 00 05 15 00 00 00 47 19 B6 08 D8 72 C8 03   ............G.¶.ØrÈ.
000000C8  DB 01 6C 48 01 02 00 00                                        Û.lH....
```

C:\ProjectWorkingFolder\RegistryViewerZ\UsrClassDeletedBags.dat | Bytes selected: 4 | Offset: 0x10 (16) | ?

The 'Full details as text' tab contains a textual representation of the selected key including the NK record, all VK records, and the SK record. This can be copied and pasted into reports as needed.

Finally, the Hive details tab contains information about the hive where the key was found. This includes the sequence numbers, timestamp, length, root key name, checksum, and so on.


Technical details for 'Local Settings\Software\Microsoft\Windows\Shell\BagMRU'
NK record
Values
Subkeys
SK record
Full details as text
Hive details

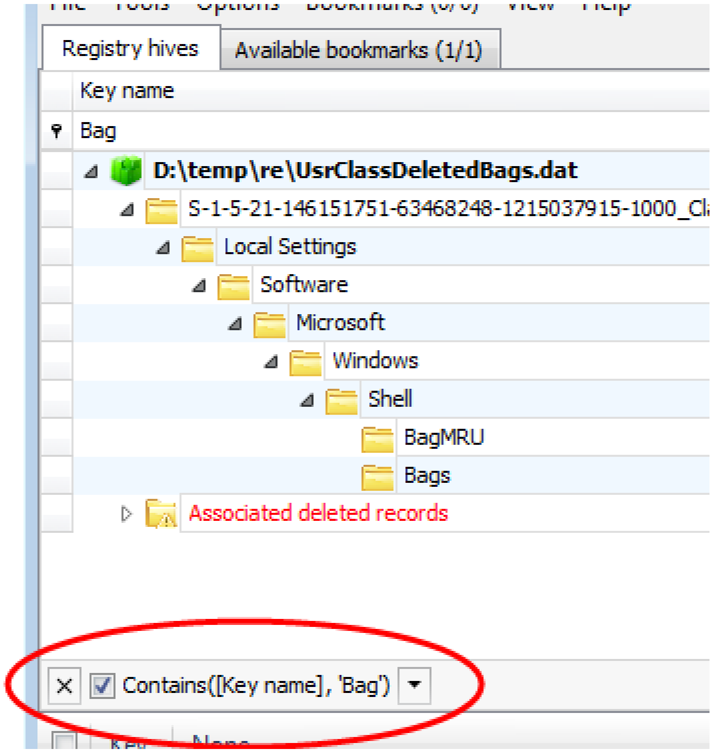
General information	
Signature	regf
Sequence 1	0x15 (21)
Sequence 2	0x15 (21)
Timestamp	2/1/2015 7:15:49 PM +00:00
Version	1.3
Length	0x34000 (212,992)
Embedded file name	\Microsoft\Windows\UsrClass.dat
Root cell index	0x20 (32)
Root key name	S-1-5-21-146151751-63468248-1215037915-1000_Classes
CheckSum	0x8CB71E6C (-1934156180)
Calculated CheckSum	0x8CB71E6C (-1934156180)
CheckSum match	☑

Other options

On all tree and grid controls, right clicking on a column header will show a menu that allows for such things as sorting, showing and hiding columns, showing the filter editor, and submenu related to conditional formatting.

Filters

When filters are in place (by entering text in the areas below the column name), information about the active filter will be shown at the bottom of the tree or grid as shown below.

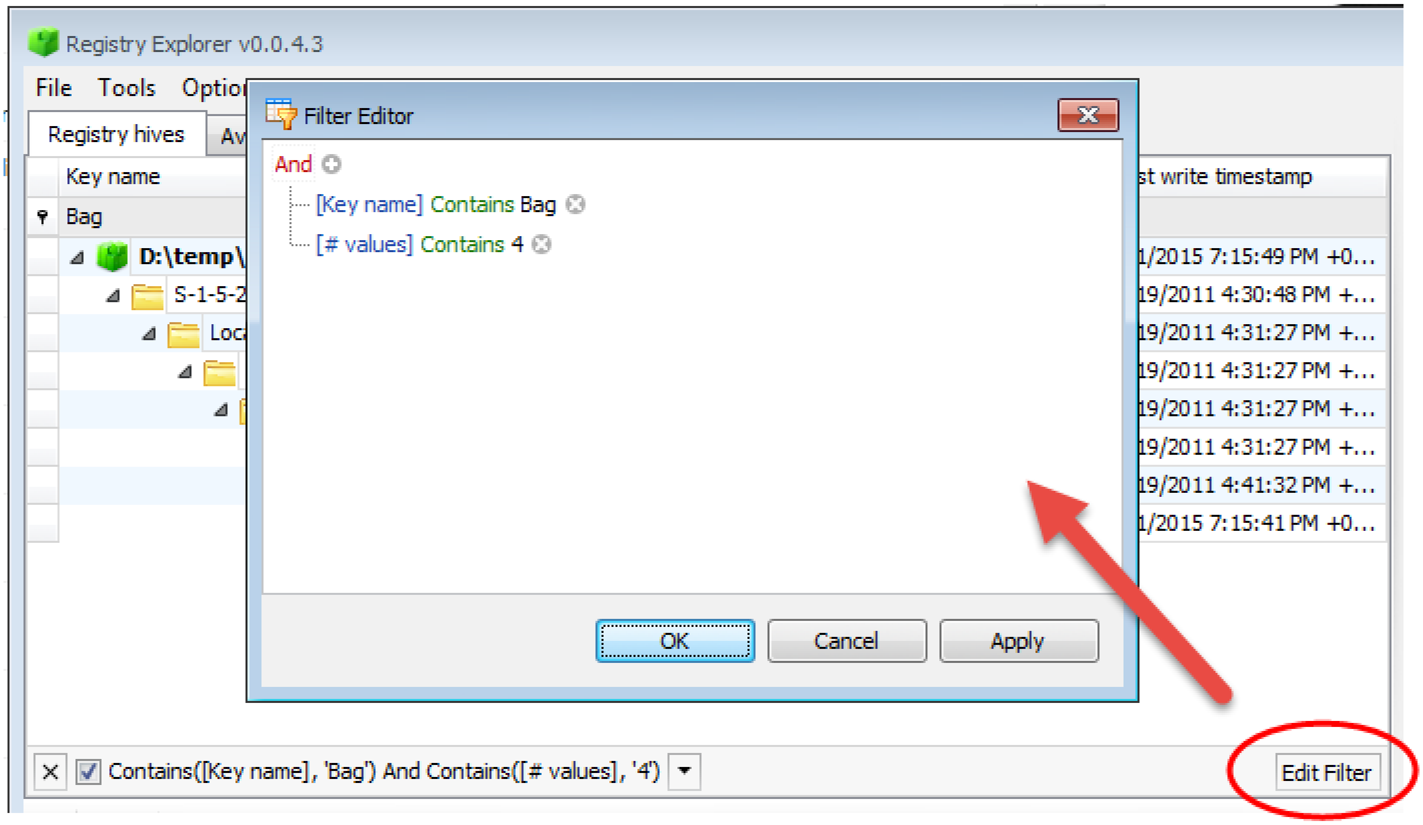

File Tools Options Bookmarks (0/0) View Help
Registry hives
Available bookmarks (1/1)
Key name
Bag
Key
Name

- D:\temp\re\UsrClassDeletedBags.dat
  - S-1-5-21-146151751-63468248-1215037915-1000_Cl
    - Local Settings
      - Software
        - Microsoft
          - Windows
            - Shell
              - BagMRU
              - Bags
  - Associated deleted records

☒ ☑ Contains([Key name], 'Bag') ▾

The leftmost X can be used to clear the active filter, the checkbox can be used to disable the filter without clearing it, and the down arrow on the right side contains a history of the different filters that have been recently used.

The ‘Edit filter’ button on the far right allows you to edit the current filter as needed.



And

[Key name] Contains Bag

[# values] Contains 4

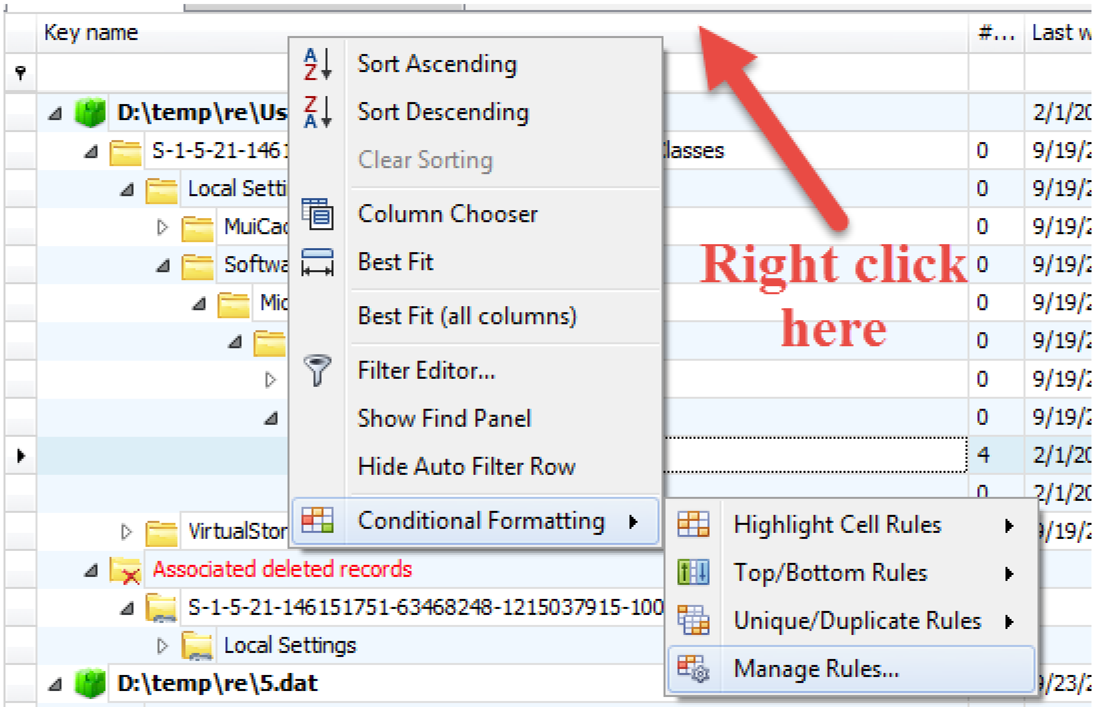
Contains([Key name], 'Bag') And Contains([# values], '4')

Using these options, very detailed filters can be created.

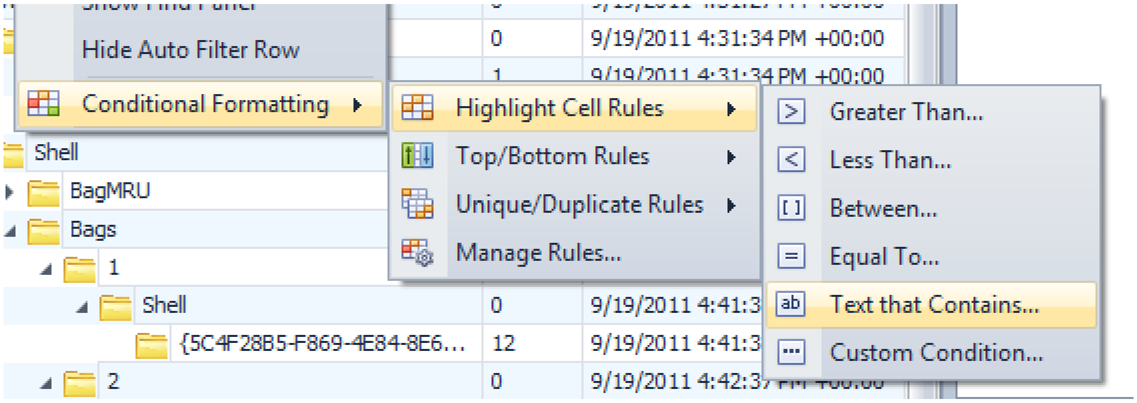
Conditional formatting

The Registry hives tree, Available bookmarks tree, values grid, and Find results grid all support creating rules to format any column contained therein.

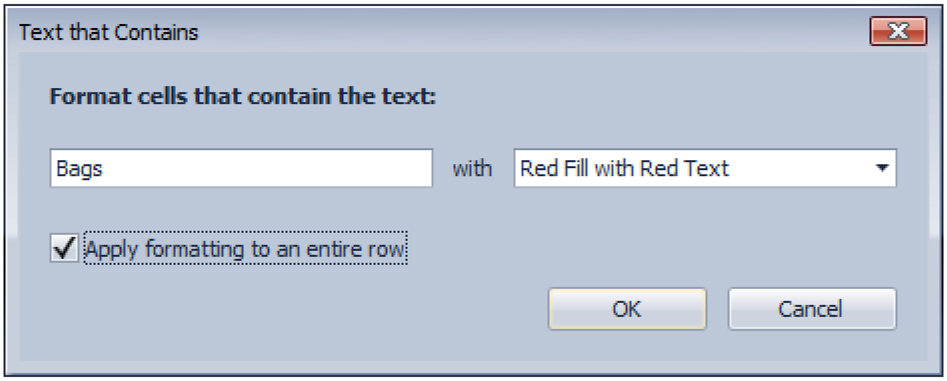
For example, by right clicking on the Key name column in the Registry hives tree, the following menu is shown.



There are options to format things on a variety of conditions, as seen below.



If we select the ‘Text that contains…’ option and enter ‘Bags’ along with how we want any matching rows to be formatted, the tree will reflect these changes. For example, if we entered the following conditions:

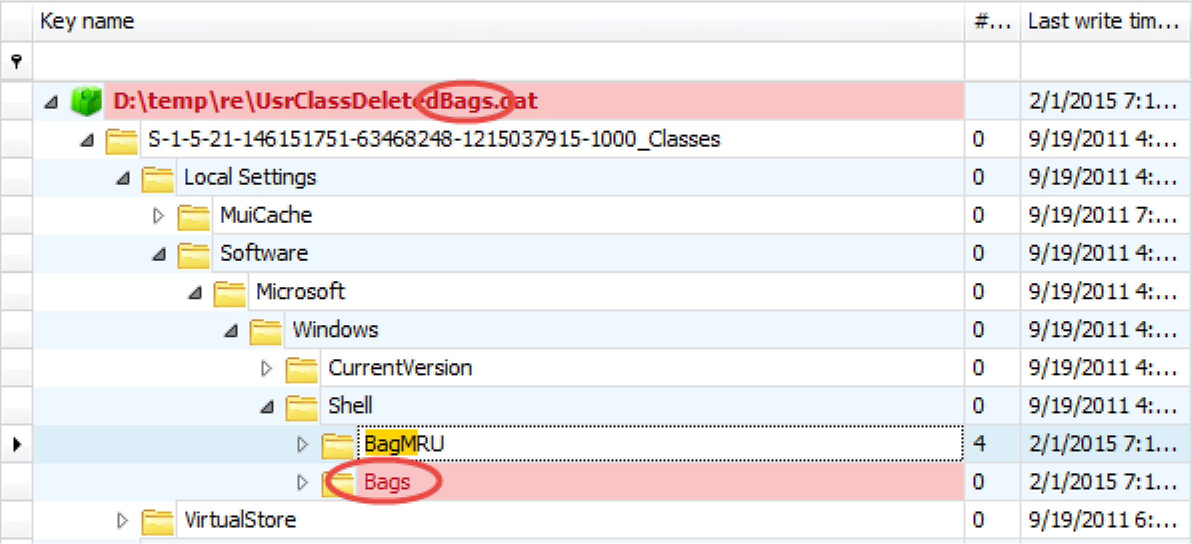

Text that Contains

Format cells that contain the text:

Bags with Red Fill with Red Text

☑ Apply formatting to an entire row

The Registry hives tree would then look like the screen shot below. The instances of ‘Bags’ in the highlighted rows have been circled for emphasis.



Key name	#...	Last write tim...
D:\temp\re\UsrClassDeletedBags.dat		2/1/2015 7:1...
S-1-5-21-146151751-63468248-1215037915-1000_Classes	0	9/19/2011 4:...
Local Settings	0	9/19/2011 4:...
MuiCache	0	9/19/2011 7:...
Software	0	9/19/2011 4:...
Microsoft	0	9/19/2011 4:...
Windows	0	9/19/2011 4:...
CurrentVersion	0	9/19/2011 4:...
Shell	0	9/19/2011 4:...
BagMRU	4	2/1/2015 7:1...
Bags	0	2/1/2015 7:1...
VirtualStore	0	9/19/2011 6:...

These formatting options allow you to create powerful visual indicators when data that is relevant to you is present in the hives you are looking at. Of course, all formatting options are remembered.

Use the same conditional formatting menu to edit rules.

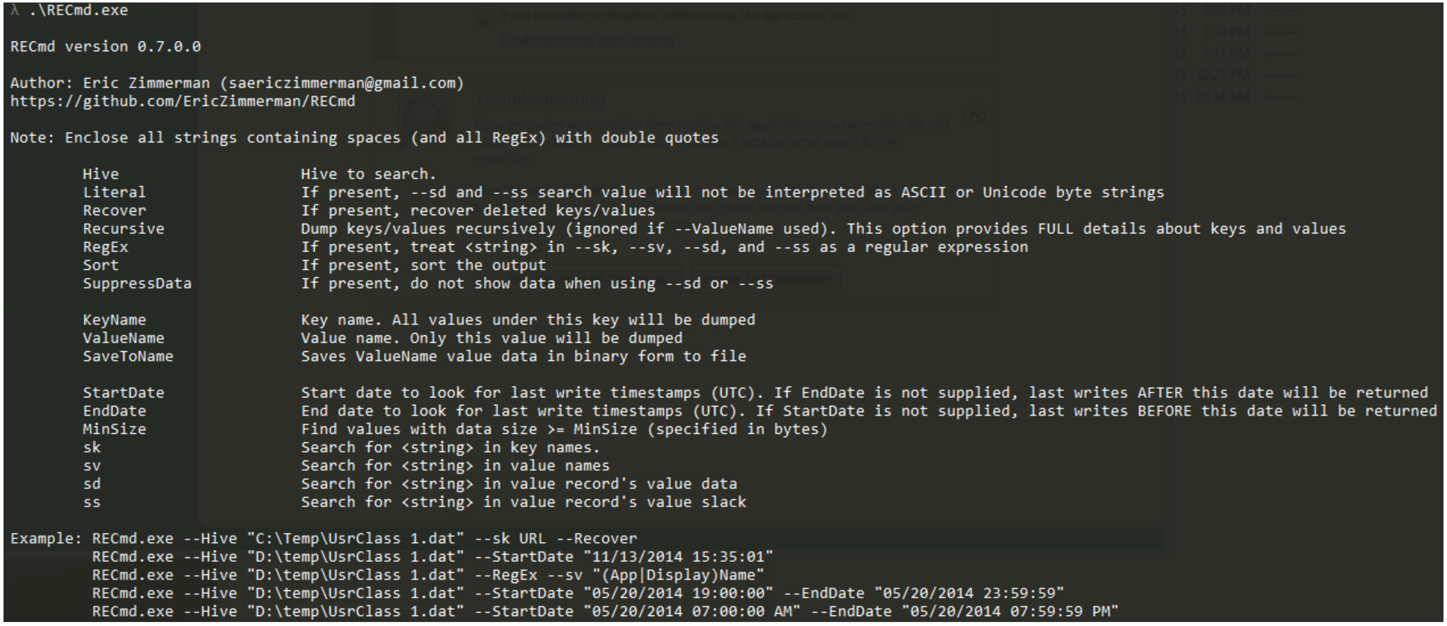
RECmd

RECmd is a command line tool used to access offline Registry hives. It includes many of the same features as Registry Explorer including searching, looking at keys and values, and exporting data.

RECmd uses the same back end as Registry Explorer to process Registry hives. RECmd is open source and the source code is available here.

Getting started

Running RECmd.exe without any arguments displays a list of command line options as shown below.



```
λ .\RECmd.exe

RECmd version 0.7.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/RECmd

Note: Enclose all strings containing spaces (and all RegEx) with double quotes

        Hive                    Hive to search.
        Literal                 If present, --sd and --ss search value will not be interpreted as ASCII or Unicode byte strings
        Recover                 If present, recover deleted keys/values
        Recursive               Dump keys/values recursively (ignored if --ValueName used). This option provides FULL details about keys and values
        RegEx                   If present, treat <string> in --sk, --sv, --sd, and --ss as a regular expression
        Sort                    If present, sort the output
        SuppressData            If present, do not show data when using --sd or --ss

        KeyName                 Key name. All values under this key will be dumped
        ValueName               Value name. Only this value will be dumped
        SaveToName              Saves ValueName value data in binary form to file

        StartDate               Start date to look for last write timestamps (UTC). If EndDate is not supplied, last writes AFTER this date will be returned
        EndDate                 End date to look for last write timestamps (UTC). If StartDate is not supplied, last writes BEFORE this date will be returned
        MinSize                 Find values with data size >= MinSize (specified in bytes)
        sk                      Search for <string> in key names.
        sv                      Search for <string> in value names
        sd                      Search for <string> in value record's value data
        ss                      Search for <string> in value record's value slack

Example: RECmd.exe --Hive "C:\Temp\UsrClass 1.dat" --sk URL --Recover
         RECmd.exe --Hive "D:\temp\UsrClass 1.dat" --StartDate "11/13/2014 15:35:01"
         RECmd.exe --Hive "D:\temp\UsrClass 1.dat" --RegEx --sv "(App|Display)Name"
         RECmd.exe --Hive "D:\temp\UsrClass 1.dat" --StartDate "05/20/2014 19:00:00" --EndDate "05/20/2014 23:59:59"
         RECmd.exe --Hive "D:\temp\UsrClass 1.dat" --StartDate "05/20/2014 07:00:00 AM" --EndDate "05/20/2014 07:59:59 PM"
```

There are three groups of command line options for RECmd.

General

This category includes the Hive and Recover switches. Hive is always required.

- **Hive**: The full path to the hive to process. If the path contains spaces, include them in double quotes.
- **Literal**: If present, the --sd and --ss search value is not interpreted as ASCII and Unicode strings
- **Recover**: If present, recover deleted keys and values
- **Recursive**: If present, dump keys and values recursively from the key specified by KeyName. This option provides much more detail about keys and values.
- **RegEx**: If present, treat <string> in --sk, --sv, --sd, and –ss as a regular expression
- **Sort**: If present, sort the output in a meaningful way based on the type of data requested.
- **SuppressData**: If present, do not include the contents of value data when using the sd or nd switches. This is useful for seeing the key paths and value names without the potential noise of the value data itself.

Query

If either the key or value has spaces in them, be sure to enclose them in quotes.

When passing in key names, the root key name is optional. This is because most of the time you will not even know the root key name in order to be able to include it.

To get default values, use a value name of "(default)".

- **KeyName**: The key name to look for. If used without ValueName, displays all subkeys and values.
- **ValueName**: Display only the value specified
- **SaveToName**: Saves ValueName value data in binary form to a file

Search

This is a particularly useful feature to locate data across hives in key names, value names, and perhaps most importantly, in value data.

Searching is broken down into four types, by last write timestamp, value data minimum size, simple string searches (and regular expression (RegEx) based searches when –RegEx is present).

- **StartDate**: The earliest date to look for in Registry key last write timestamps. Timestamp should be in UTC.
- **EndDate**: The latest date to look for in Registry key last write timestamps. Timestamp should be in UTC.
- **MinSize**: Find values with value data size greater than or equal to the specified size (in bytes).
- **sk**: Search for <string> in key names
- **sv**: Search for <string> in value names
- **sd**: Search for <string> in value record's value data. The value data will be converted to its equivalent in ASCII and Unicode and also searched/compared to <string> unless the --Literal switch is used
- **ss**: Search for <string> in value record's value slack. The value slack will be converted to its equivalent in ASCII and Unicode and also searched/compared to <string> unless the --Literal switch is used

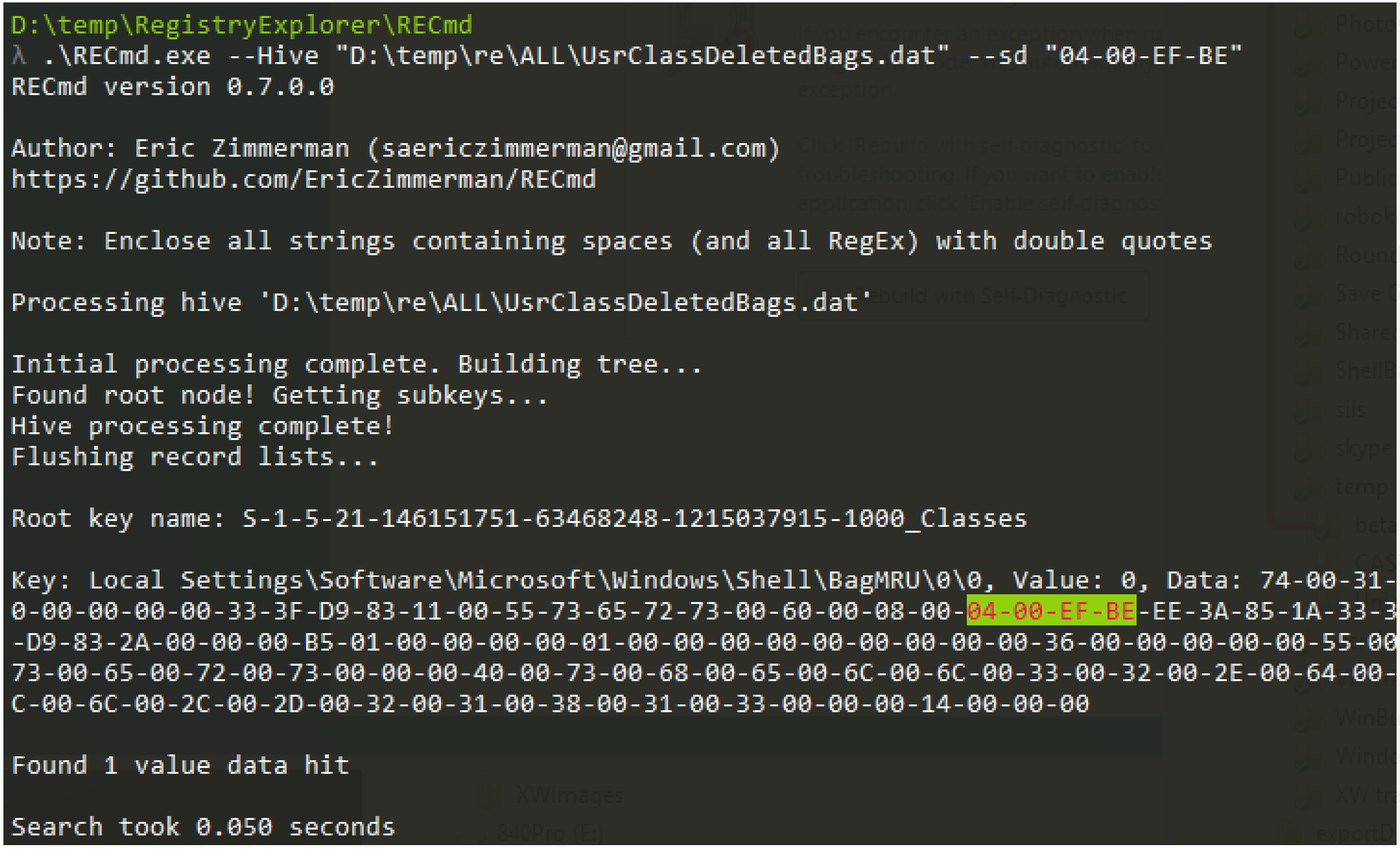
Simple searches

The two letter search options starting with 's' are string search options. These options look for matches via 'contains' logic rather than 'begins with' or similar. For example, if you search for 'cache', the following keys would match if they existed in the Registry hive:

- Muicache
- Cache items
- UnCAcHeD

Simple searches are not case sensitive.

To search for binary data in value data, simply string the hex characters you want to find together, separated by dashes (04-00-EF-BE for example).



```
D:\temp\RegistryExplorer\RECmd
λ .\RECmd.exe --Hive "D:\temp\re\ALL\UsrClassDeletedBags.dat" --sd "04-00-EF-BE"
RECmd version 0.7.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/RECmd

Note: Enclose all strings containing spaces (and all RegEx) with double quotes

Processing hive 'D:\temp\re\ALL\UsrClassDeletedBags.dat'

Initial processing complete. Building tree...
Found root node! Getting subkeys...
Hive processing complete!
Flushing record lists...

Root key name: S-1-5-21-146151751-63468248-1215037915-1000_Classes

Key: Local Settings\Software\Microsoft\Windows\Shell\BagMRU\0\0, Value: 0, Data: 74-00-31-
0-00-00-00-00-33-3F-D9-83-11-00-55-73-65-72-73-00-60-00-08-00-04-00-EF-BE-EE-3A-85-1A-33-3
-D9-83-2A-00-00-00-B5-01-00-00-00-00-01-00-00-00-00-00-00-00-00-00-36-00-00-00-00-00-55-00
73-00-65-00-72-00-73-00-00-00-40-00-73-00-68-00-65-00-6C-00-6C-00-33-00-32-00-2E-00-64-00-
C-00-6C-00-2C-00-2D-00-32-00-31-00-38-00-31-00-33-00-00-00-14-00-00-00

Found 1 value data hit

Search took 0.050 seconds
```

This allows you to find signatures for common data structures ANYWHERE in the Registry. The binary signature used above is that of a BEEF0004 extension block, commonly used in ShellBags. It contains information such as MAC dates/times, MFT info, etc.

When using the sd and ss switches, the value data or slack will be converted to its equivalent ASCII and Unicode representation from the raw bytes. For example, if you searched for Ask, three searches would actually happen:

1. For the Ask string itself
2. Raw data converted to ASCII string. A case insensitive search against this string is performed. If found, the position of the hit is used to extract the exact string that was hit on. This string is then converted back to bytes and reported as a hit.
3. Raw data converted to Unicode string. The rest happens as in step 2.

This allows string searches to find data regardless of encoding or case. If data is found in encoded form, the exact bytes making up the hit are highlighted. These bytes may differ from the searched for string if the capitalization was different.

If the --Literal switch is used with sd or ss, then only the first search is done behind the scenes. This allows you to look for specific byte patterns without RECmd interpreting raw data or slack to ASCII or Unicode.

Regular expression searches

When the --RegEx switch is present, the search term used is treated as a regular expression. Regular expression searches offer much more powerful capabilities to find things at the cost of having to follow a more complex set of rules when building search terms. Another tradeoff is that it can be slower depending on how complicated your RegEx is.

Enclose the RegEx in quotes to make sure the shell does not try to interpret anything in there.

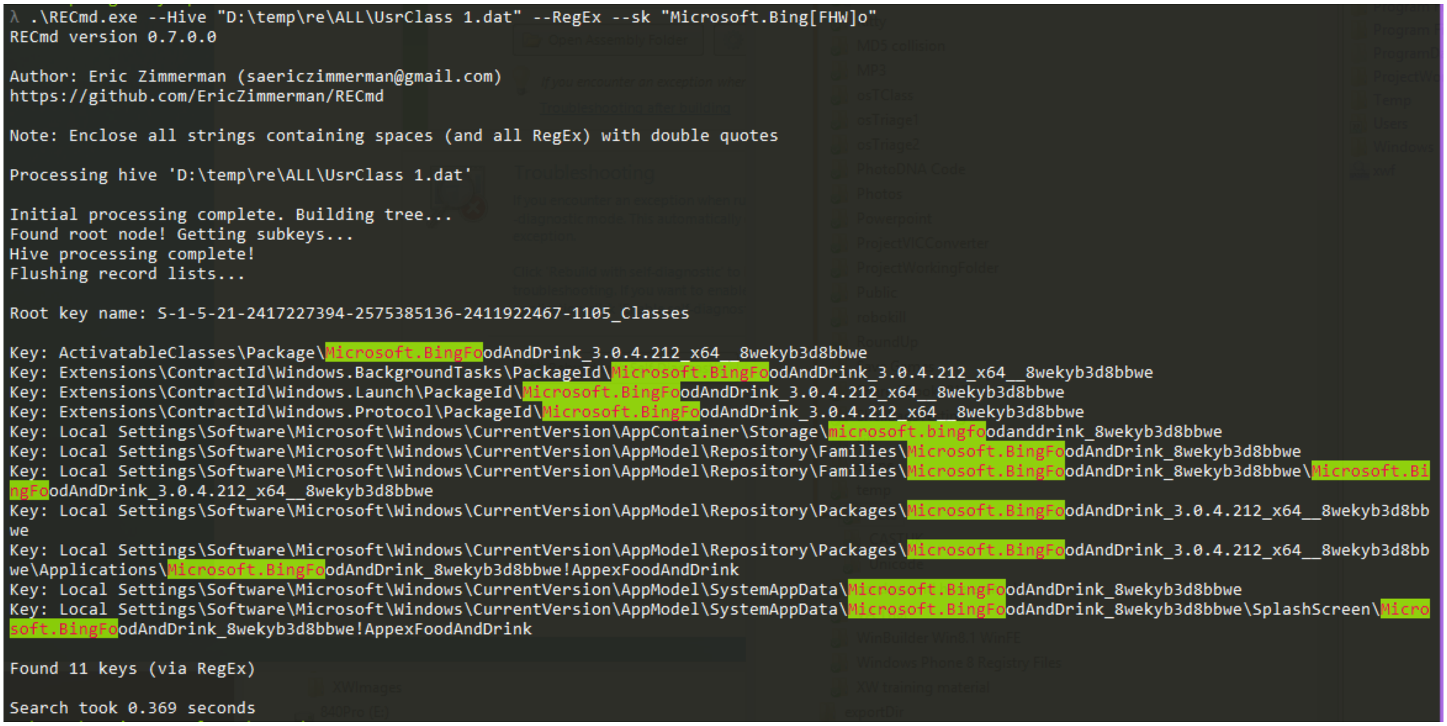
As with simple searches, regular expression based searches are case insensitive.

Regular Expression examples

Finding keys

To find all keys that contain ‘Microsoft.Bing’ followed by an F, H, or a W, then an o, use the following search:

RECmd.exe --Hive "D:\temp\re\UsrClass 1.dat" --RegEx --sk "Microsoft.Bing[FHW]o”



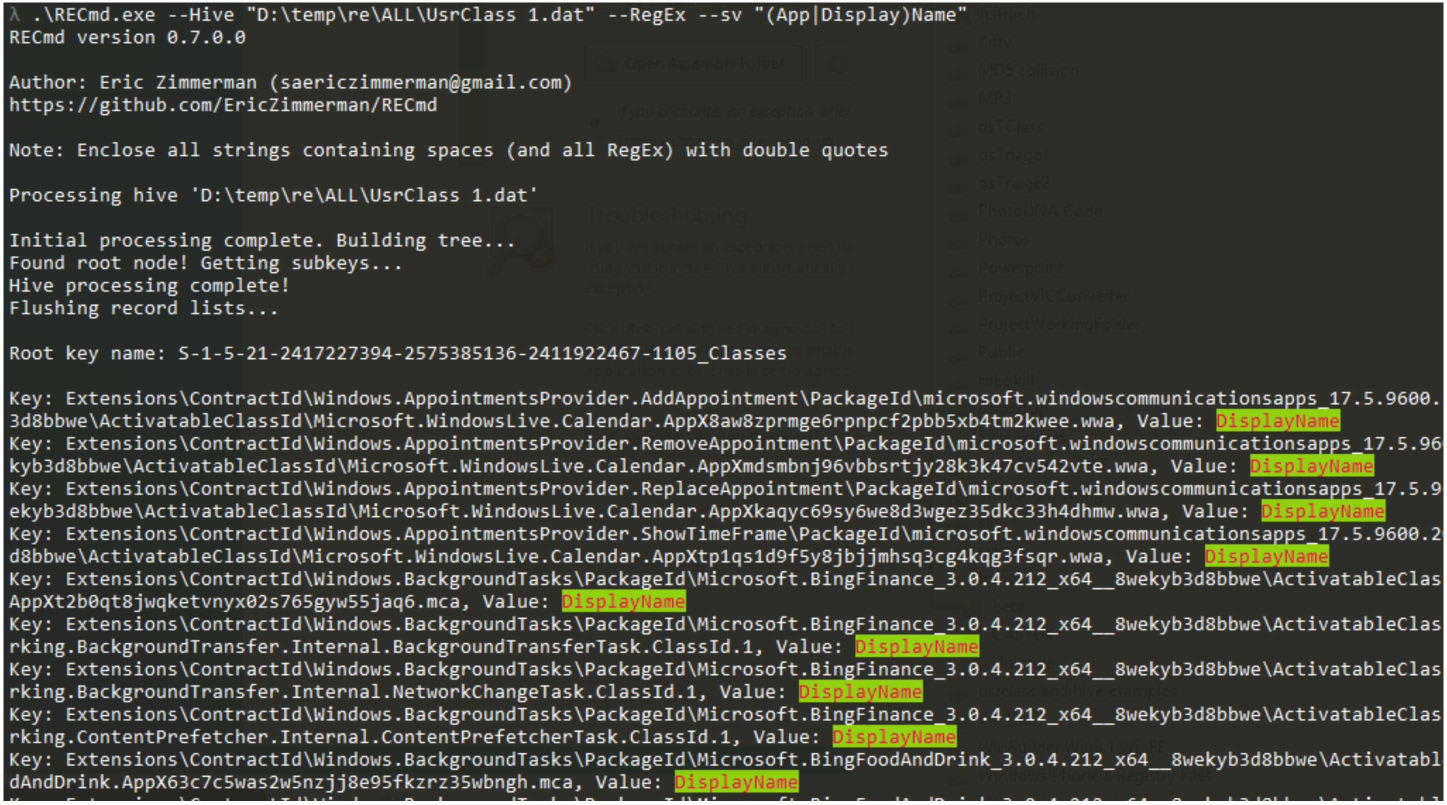
```
λ .\RECmd.exe --Hive "D:\temp\re\ALL\UsrClass 1.dat" --RegEx --sk "Microsoft.Bing[FHW]o"
RECmd version 0.7.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/RECmd

Note: Enclose all strings containing spaces (and all RegEx) with double quotes

Processing hive 'D:\temp\re\ALL\UsrClass 1.dat'

Initial processing complete. Building tree...
Found root node! Getting subkeys...
Hive processing complete!
Flushing record lists...

Root key name: S-1-5-21-2417227394-2575385136-2411922467-1105_Classes

Key: ActivatableClasses\Package\Microsoft.BingFoodAndDrink_3.0.4.212_x64__8wekyb3d8bbwe
Key: Extensions\ContractId\Windows.BackgroundTasks\PackageId\Microsoft.BingFoodAndDrink_3.0.4.212_x64__8wekyb3d8bbwe
Key: Extensions\ContractId\Windows.Launch\PackageId\Microsoft.BingFoodAndDrink_3.0.4.212_x64__8wekyb3d8bbwe
Key: Extensions\ContractId\Windows.Protocol\PackageId\Microsoft.BingFoodAndDrink_3.0.4.212_x64__8wekyb3d8bbwe
Key: Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage\microsoft.bingfoodanddrink_8wekyb3d8bbwe
Key: Local Settings\Software\Microsoft\Windows\CurrentVersion\AppModel\Repository\Families\Microsoft.BingFoodAndDrink_8wekyb3d8bbwe
Key: Local Settings\Software\Microsoft\Windows\CurrentVersion\AppModel\Repository\Families\Microsoft.BingFoodAndDrink_8wekyb3d8bbwe\Microsoft.Bi
ngFoodAndDrink_3.0.4.212_x64__8wekyb3d8bbwe
Key: Local Settings\Software\Microsoft\Windows\CurrentVersion\AppModel\Repository\Packages\Microsoft.BingFoodAndDrink_3.0.4.212_x64__8wekyb3d8bb
we
Key: Local Settings\Software\Microsoft\Windows\CurrentVersion\AppModel\Repository\Packages\Microsoft.BingFoodAndDrink_3.0.4.212_x64__8wekyb3d8bb
we\Applications\Microsoft.BingFoodAndDrink_8wekyb3d8bbwe!AppexFoodAndDrink
Key: Local Settings\Software\Microsoft\Windows\CurrentVersion\AppModel\SystemAppData\Microsoft.BingFoodAndDrink_8wekyb3d8bbwe
Key: Local Settings\Software\Microsoft\Windows\CurrentVersion\AppModel\SystemAppData\Microsoft.BingFoodAndDrink_8wekyb3d8bbwe\SplashScreen\Micro
soft.BingFoodAndDrink_8wekyb3d8bbwe!AppexFoodAndDrink

Found 11 keys (via RegEx)

Search took 0.369 seconds
```

Finding values

To find all values with names that contain either ‘AppName’ or ‘DisplayName’, use the following search:

RECmd.exe --Hive "D:\temp\re\UsrClass 1.dat" --RegEx --sv "(App|Display)Name"

320 results were found in 0.380 seconds, but due to the length of the output, only the first few are shown below.

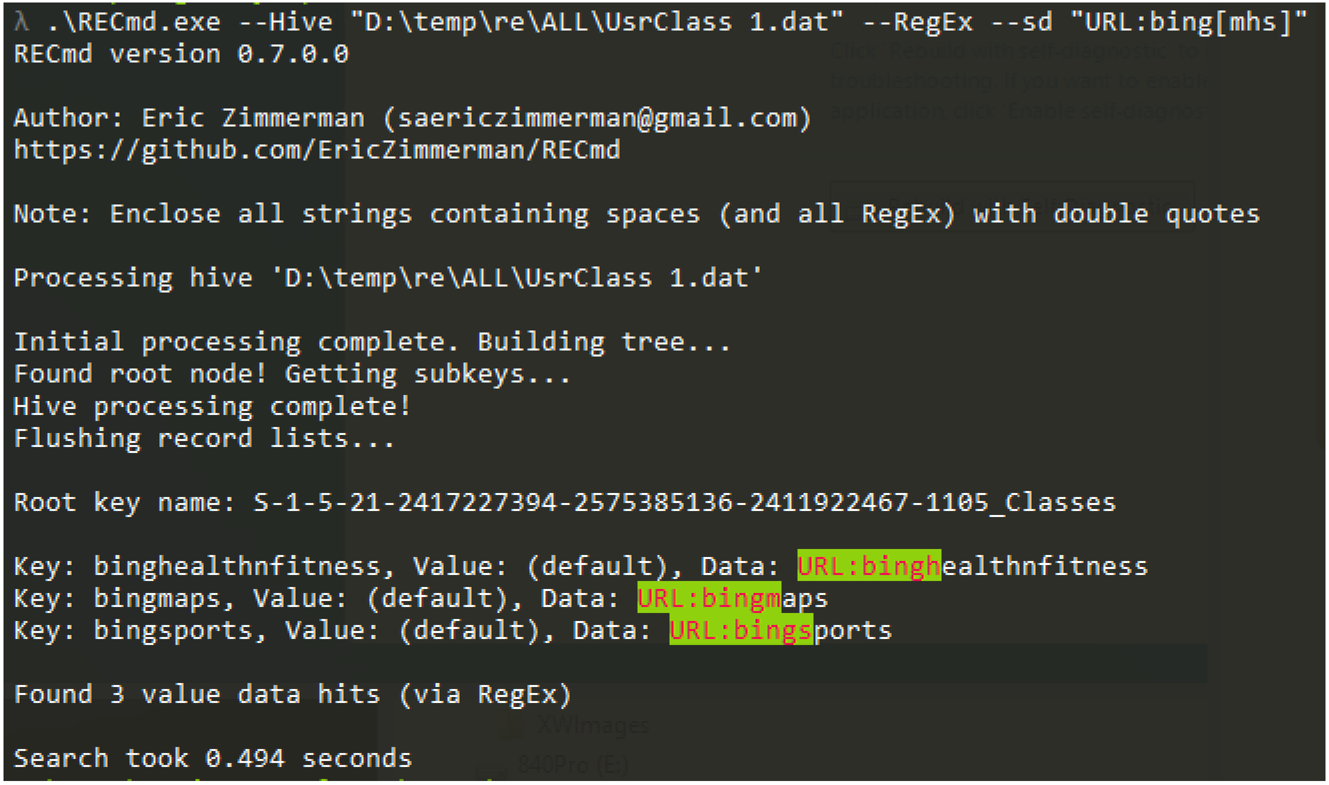


```
λ .\RECmd.exe --Hive "D:\temp\re\ALL\UsrClass 1.dat" --RegEx --sv "(App|Display)Name"
RECmd version 0.7.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/RECmd

Note: Enclose all strings containing spaces (and all RegEx) with double quotes

Processing hive 'D:\temp\re\ALL\UsrClass 1.dat'

Initial processing complete. Building tree...
Found root node! Getting subkeys...
Hive processing complete!
Flushing record lists...

Root key name: S-1-5-21-2417227394-2575385136-2411922467-1105_Classes

Key: Extensions\ContractId\Windows.AppointmentsProvider.AddAppointment\PackageId\microsoft.windowscommunicationsapps_17.5.9600.
3d8bbwe\ActivatableClassId\Microsoft.WindowsLive.Calendar.AppX8aw8zprmge6rpnpcf2pbb5xb4tm2kwee.wwa, Value: DisplayName
Key: Extensions\ContractId\Windows.AppointmentsProvider.RemoveAppointment\PackageId\microsoft.windowscommunicationsapps_17.5.96
kyb3d8bbwe\ActivatableClassId\Microsoft.WindowsLive.Calendar.AppXmdsmbnj96vbbsrtjy28k3k47cv542vte.wwa, Value: DisplayName
Key: Extensions\ContractId\Windows.AppointmentsProvider.ReplaceAppointment\PackageId\microsoft.windowscommunicationsapps_17.5.9
ekyb3d8bbwe\ActivatableClassId\Microsoft.WindowsLive.Calendar.AppXkaqyc69sy6we8d3wgez35dkc33h4dhmw.wwa, Value: DisplayName
Key: Extensions\ContractId\Windows.AppointmentsProvider.ShowTimeFrame\PackageId\microsoft.windowscommunicationsapps_17.5.9600.2
d8bbwe\ActivatableClassId\Microsoft.WindowsLive.Calendar.AppXtp1qs1d9f5y8jbjjmhsq3cg4kqg3fsqr.wwa, Value: DisplayName
Key: Extensions\ContractId\Windows.BackgroundTasks\PackageId\Microsoft.BingFinance_3.0.4.212_x64__8wekyb3d8bbwe\ActivatableClas
AppXt2b0qt8jwqketvnyx02s765gyw55jaq6.mca, Value: DisplayName
Key: Extensions\ContractId\Windows.BackgroundTasks\PackageId\Microsoft.BingFinance_3.0.4.212_x64__8wekyb3d8bbwe\ActivatableClas
rking.BackgroundTransfer.Internal.BackgroundTransferTask.ClassId.1, Value: DisplayName
Key: Extensions\ContractId\Windows.BackgroundTasks\PackageId\Microsoft.BingFinance_3.0.4.212_x64__8wekyb3d8bbwe\ActivatableClas
rking.BackgroundTransfer.Internal.NetworkChangeTask.ClassId.1, Value: DisplayName
Key: Extensions\ContractId\Windows.BackgroundTasks\PackageId\Microsoft.BingFinance_3.0.4.212_x64__8wekyb3d8bbwe\ActivatableClas
rking.ContentPrefetcher.Internal.ContentPrefetcherTask.ClassId.1, Value: DisplayName
Key: Extensions\ContractId\Windows.BackgroundTasks\PackageId\Microsoft.BingFoodAndDrink_3.0.4.212_x64__8wekyb3d8bbwe\Activatabl
dAndDrink.AppX63c7c5was2w5nzjj8e95fkzrz35wbngh.mca, Value: DisplayName
```

Finding data

To find all values whose data contains ‘URL:bing’ followed by either an m, h, or s, use the following search:

RECmd.exe --Hive "D:\temp\re\UsrClass 1.dat" --RegEx --sd "URL:bing[mhs]"



```
λ .\RECmd.exe --Hive "D:\temp\re\ALL\UsrClass 1.dat" --RegEx --sd "URL:bing[mhs]"
RECmd version 0.7.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/RECmd

Note: Enclose all strings containing spaces (and all RegEx) with double quotes

Processing hive 'D:\temp\re\ALL\UsrClass 1.dat'

Initial processing complete. Building tree...
Found root node! Getting subkeys...
Hive processing complete!
Flushing record lists...

Root key name: S-1-5-21-2417227394-2575385136-2411922467-1105_Classes

Key: binghealthnfitness, Value: (default), Data: URL:binghealthnfitness
Key: bingmaps, Value: (default), Data: URL:bingmaps
Key: bingsports, Value: (default), Data: URL:bingsports

Found 3 value data hits (via RegEx)

Search took 0.494 seconds
```

For more examples, run RECmd.exe without any command line arguments.

All regular expressions must of course be valid .net regular expressions. Different flavors of RegEx providers allow for different syntax, so be sure to use the proper syntax.

RegEx tutorials for .net.
https://msdn.microsoft.com/en-us/library/az24scfc%28v=vs.110%29.aspx

http://regexhero.net/reference/

https://msdn.microsoft.com/en-us/library/hs600312%28v=vs.110%29.aspx

http://www.codeproject.com/Articles/9099/The-Minute-Regex-Tutorial

http://www.systemtextregularexpressions.com/help

RegExBuddy is an awesome tool for building and testing RegEx against data sets.

Version changes

Version 0.7.1.0

RECmd changes

New: Added --Dir switch. This recursively searches for hives in a given directory and searches each of them

Registry Explorer changes

New: Registry Explorer can now function as a "default application" in that you can associate RE with *.dat and then double click hives. This also allows for setting up RE in other apps like X-Ways as an external viewer, dragging and dropping hives onto RE shortcut/executable, etc

New: Added Check for updates to About menu

Version 0.7.0.0

As of 0.7.0.0, Registry Explorer and RECmd are included together.

RECmd changes

NEW: Added –Literal switch. When present, --sd and --ss switches will not be interpreted
NEW: Added --ss switch for searching Value slack space
NEW: Search terms are now highlighted in search results. Edit nlog.config to adjust colors for foreground and background
NEW: Added --RegEx switch. When present, treat <string> in --sk, --sv, --sd, and --ss as a regular expression
NEW: If nlog.config is missing, add default config and warn user

CHANGE: Switches are NOT case sensitive any more
CHANGE: Remove RegEx specific switches (See --RegEx above)
CHANGE: Tweak command line option descriptions
CHANGE: Updated nlog

See here for changes in version 0.6.1.0.

Registry Explorer changes

This version is pretty much a complete rewrite under the hood. This was done to address performance issues due to initial (bad) design decisions.

Hive processing is fully asynchronous, but very large hives can take a few seconds to display once the hive is loaded. This is due to the need to load all

NEW: Full support for searching including key names, value names, and value data, both with simple searches and RegEx. Searching based on last write timestamps is supported as well
NEW: Fully asynchronous loading of hives which keeps the GUI responsive, even when loading 100+ MB hives (I am looking at you SOFTWARE hive)
NEW: Tech details hex editors now update with offset and selection length when bytes are selected
NEW: Added value context menu to copy value summary (a combination of name, type, and data), name, type, data, and slack to clipboard
NEW: Add value context menu to export data and slack to a file
NEW: Settings for things persist

NEW: Search strings are remembered and autopopulate when typing on the Find form. Use the Tools menu to clear
NEW: Added Convert | To hex ASCII and To hex Unicode to Find. This allows you to look for encoded strings in value data without having to manually convert strings to hex
NEW: Allow deleting of user created bookmarks in Bookmark Manager via Ctrl-Delete
NEW: Added context menu to Available bookmarks (Copy, expand/collapse, tech details) that work the same as the 'Registry hives' context menu options
NEW: Added 'Jump to key' context menu item on Available bookmarks tab that will select the hive's key on the 'Registry hives' tab
NEW: More hot keys added to main/context menus
NEW: Added 'Root key name' to Tech details | Hive details properties and strip root key from Tech details window title to save space
NEW: Added Export 'Registry hive' menu to File menu. This exports the tree exactly as it is shown to the selected format
NEW: Enable/disable expand/collapse subkey options depending on the expanded state of the selected key
NEW: Save positions of vertical and horizontal splitters
NEW: Trees and grids all save settings (sorting, filtering, conditional formatting rules) between sessions
NEW: Save size of main form
NEW: Improved hex editor control for RegBinary keys. Added offset, selection length, and data interpreter
NEW: Added 'Show associated deleted records' and 'Show unassociated deleted records' to Options menu
NEW: Added 'Slack viewer' tab for values that have slack space
NEW: Added 'Show parent keys when filtering' to options menu. Turning this OFF shows only the keys that match the filter. When ON, parent keys to keys matching the filter are also shown
NEW: Added a Total messages counter to lower status bar (far right) that indicates the total number of messages available on the Messages form
NEW: Added skinning support. Active skin can be changed from the Options menu
NEW: Added icon for Registry hive in the Registry hives tree to visually separate it from keys
NEW: Make hive name bold to make it stand out from keys
NEW: Tech details info can be copied via Ctrl+C (just the value) or Ctrl+Alt+C (Name: Value)
NEW: All hex viewers now support Ctrl+C to copy selected bytes to clipboard
NEW: Search for minimum value sizes added
NEW: Search in value slack added

CHANGE: Allow resizing of window below 800x600
CHANGE: Drag and dropping of hives supported on any of the 3 main sections of Registry Explorer
CHANGE: Status bars adjusted. Added options to hold Shift when double clicking in order to copy different parts of the key/value
CHANGE: Add vertical scroll bar to Technical details hex editors
CHANGE: Rename tree context menus from 'child nodes' to 'subkeys'
CHANGE: Hide Messages form by default since things load and process faster when its hidden
CHANGE: Icon for existing key placeholder in Associated deleted records updated
CHANGE: Icon for Associated deleted records updated
CHANGE: Made legend icons bigger
CHANGE: Bookmarks manager now allows editing/deleted both common and user created bookmarks

FIX: Bug fixes in Registry parser (yay unit tests)
FIX: Show SK record in Technical details form

Version 0.2.0.0

NEW: Added new tab in upper left, Available bookmarks, that shows all available bookmarks across all loaded Registry hives

NEW: Added 'Technical details' option to context menu. Use this to view all the down and dirty details about a key including its bytes, its security key, subkeys, values, etc. This provides an easy to use way to explore and validate Registry tools
NEW: Added several hotkeys for commonly used key context menu items
NEW: Allow exporting of keys either individually or recursively to .reg format via the context menu
NEW: Add 'Collapse all hives' button to status bar.
NEW: Added more bookmarks

CHANGE: Prevent illegal file name characters in category names (\, /, |, and so on). Any illegal characters will be replaced with an underscore
CHANGE: Nlog logging added
CHANGE: Registry parsing is now ~150% faster and memory usage reduced by 40-80%
CHANGE: Prevent the same hive from being loaded more than once
CHANGE: Expand the top level node after loading a hive
CHANGE: Hide or unhide all matching keys in all open hives vs only the active hive

FIX: When removing keys from auto hide list, remove any hidden keys in the tree that match as well
FIX: Actually export the timestamp when exporting Messages
FIX: GUI polish

Version 0.1.8.0

Initial release

Appendix A – Contributors

The following people have contributed in one way or another during the development and refinement of SBE

- SA/FE Devon P. Ackerman devon.ackerman@ic.fbi.gov, sadevonackerman@gmail.com
- David Cowen @HECFBlog
- Dan Pullega @4n6k
- Jerod Alexander @jerod
- Willi Ballenthin @williballenthin

Appendix B – Additional resources

- http://binaryforay.blogspot.com/: Contains detailed postings on the internal workings of the Registry
- https://github.com/EricZimmerman/Registry: Source code for the back end parser used in Registry Explorer
- https://github.com/EricZimmerman/RECmd: Source code for RECmd