X-Ways Software Technology AG

X-Ways Forensics/ WinHex

Integrated Computer Forensics Environment.

Data Recovery & IT Security Tool.

Hexadecimal Editor for Files, Disks & RAM.

Manual

Copyright © 1995-2026 Stefan Fleischmann, X-Ways Software Technology AG. All rights reserved.

Contents

1 Preface 1

1.1 About WinHex and X-Ways Forensics 1
1.2 Legalities 2
1.3 License Types 5
1.4 More differences between WinHex & X-Ways Forensics 6
1.5 Getting Started with X-Ways Forensics 6

2 Technical Background 8

2.1 Using a Hex Editor 8
2.2 Endian-ness 9
2.3 Integer Data Types 9
2.4 Floating-Point Data Types 9
2.5 Date Types 10
2.6 ANSI ASCII/IBM ASCII 11
2.7 Checksums, Hashes, Digests 12
2.8 Attribute Legend 13
2.9 Technical Hints and Limitations 14

3 User Interface 15

3.1 Overview 15
3.2 Start Center 16
3.3 Directory Browser 17
3.3.1 General Description 17
3.3.2 Virtual Objects 19
3.3.3 Partition Layout 20
3.3.4 Filtering 21
3.3.5 Columns and Filters 22
3.3.6 More about the Timestamp Columns 35
3.3.7 FlexFilters 37
3.4 Mode Buttons 38
3.5 Status Bar 43
3.6 Data Interpreter 43
3.7 Position Manager 44
3.8 Useful Hints 45
3.9 Special Dialog Window Functionality 46
3.10 Command Line Parameters 48
3.11 User-Defined Keyboard Shortcuts 51

4 Menu Reference 55

4.1 Directory Browser Context Menu 56
4.2 Case Data Window Context Menu 66
4.3 Data Window Context Menu 67
4.4 File Menu 68
4.5 Edit Menu 70
4.6 Search Menu 71
4.7 Navigation Menu 72
4.8 View Menu 73
4.9 Tools Menu 75
4.10 File Tools 78
4.11 Specialist Menu 80
4.12 Options Menu 82
4.13 Window Menu 83

4.14 Help Menu 83
4.15 Windows Context Menu 85

5 Forensic Features 85

5.1 Interpret Image File As Disk 85
5.2 Case Management 87
5.3 Multi-User Coordination For Large Cases 90
5.4 Evidence Objects 93
5.5 Case Log (Activity Log) 96
5.6 Case Report 97
5.7 Labels 100
5.8 Viewer Functionality 103
5.9 Registry Report 105
5.10 Simultaneous Search 108
5.11 Logical Search 109
5.12 Search Hit List 113
5.13 Search Term List 114
5.14 Hit Count in Search Term Lists 117
5.15 Event Lists 117
5.16 Mount As Drive Letter 119
5.17 File Type Categories.txt 121
5.18 Hash Database 122
5.19 Hash Comments 125
5.20 PhotoDNA 126
5.21 FuZzyDoc 129
5.22 Text Decoding and Spreadsheet Processing 130
5.23 OCR (Text Recognition in Pictures) 133
5.24 Picture Content Analysis with AI 135
5.25 Time Zone Concept 136
5.26 Evidence File Containers 137
5.27 Related Items 140
5.28 Generator Signatures 141
5.29 Picture Metadata Details 143
5.30 External Analysis Interface 148

6 Volume Snapshots and their Refinement 149

6.1 Introduction 149
6.2 Refinement at the Volume/Sector Level 150
6.2.1 Run X-Tensions 150
6.2.2 Particularly thorough file system data structure search 150
6.2.3 File Header Signature Search 152
6.2.4 Block-wise Hashing and Matching 153
6.3 Refinement at the File Level 154
6.3.1 Hash Value Computation and Matching 155
6.3.2 File Type Verification 156
6.3.3 Extraction of Internal Metadata 157
6.3.4 Archive Exploration 160
6.3.5 E-mail Extraction 162
6.3.6 Uncovering Embedded Data 163
6.3.7 Capture Still Images from Videos 166
6.3.8 Pictures Analysis and Processing 167
6.3.9 Detection of Encryption 170
6.3.10 Indexing 170
6.4 More Information about Volume Snapshot Refinement 172
6.4.1 Interdependencies 174

6.4.2 Notes 174

7 Some Basic Concepts 176

7.1 Edit Modes 176
7.2 Scripts 177
7.3 X-Tensions API 177
7.4 Sector-wise Access to Storage Devices / Disk Editor 179
7.5 Memory Editor/Analysis 181
7.6 Template Editing 182

8 Data Recovery 183

8.1 File Recovery with the Directory Browser 183
8.2 File Recovery by Type/File Header Signature Search 183
8.3 File Type Definitions 186
8.4 Manual Data Recovery 190

9 Options 190

9.1 General Options 190
9.2 Directory Browser 198
9.3 Notation/Output 203
9.4 Volume Snapshot Options 206
9.5 Viewer Programs & Gallery Options 212
9.6 Undo Options 217
9.7 Security Options 217
9.8 Search Options 220
9.9 Replace Options 225

10 Miscellaneous 226

10.1 Block 226
10.2 Modify Data 226
10.3 Conversions 227
10.4 Sector Superimposition 229
10.5 Wiping and Initializing 230
10.6 Disk Cloning 231
10.7 Images and Backups 232
10.8 Dummy Image Segments 238
10.9 Hints on Disk Cloning, Imaging, Image Restoration 239
10.10 Skeleton Images 240
10.11 Backup Manager 244
10.12 Recover/Copy Command 245
10.13 Duplicate File Detection 248
10.14 Surrogate Patterns 250
10.15 Reconstructing RAID Systems 251
10.16 NSRL RDSv3 Format 254

Appendix A: Template Definition 256

1 Header 256
2 Body: Variable Declarations 257
3 Body: Advanced Commands 259
4 Body: Flexible Integer Variables 260

Appendix B: Script Commands 261

1 Preface

1.1 About WinHex and X-Ways Forensics

Copyright © 1995-2026 Stefan Fleischmann, X-Ways Software Technology AG. All rights reserved.

X-Ways Software Technology AG
Carl-Diem-Str. 32
32257 Bünde
Germany

Web: https://www.x-ways.net/
Retrieve quotes and purchase from:
https://www.x-ways.net/order.html
E-mail address: mail@x-ways.com

Registered in Bad Oeynhausen (HRB 7475). CEO: Stefan Fleischmann. Board of directors (chairwoman): Dr. M. Horstmeyer.

X-Ways Software Technology AG is a stock corporation incorporated under the laws of the Federal Republic of Germany. WinHex was first released in 1995. This manual was compiled from the online help of WinHex/X-Ways Forensics 21.9 and was last updated in September 2026.

The software can be run on Windows 7/Server 2008 R2, Windows 8/8.1/Server 2012, Windows 10/Server 2016/Server 2019/Server 2022, Windows 11/Server 2025; 32-bit and 64-bit; Standard, PE and FE, to different extents. It can potentially still be of some use on Windows XP, 2003 Server, Vista and Server 2008, with limitations. The last version that what reasonably usable under Windows XP and Windows 2003 Server, including support for file archives and picture viewing, was v20.1. Some functionality is also available when run under Linux+Wine. However, some copy protection methods (among them dongles) unfortunately do not work under Linux+Wine at all.

User interface translation: Chinese by Sprite Guo. Japanese by Takao Horiuchi and Ichiro Sugiyama (not generally available). French by Jérôme Broutin, revised by Bernard Leprêtre. Spanish by José María Tagarro Martí. Italian by Andrea Ghirardini. Brazilian Portuguese by Heyder Lino Ferreira. Polish by ProCertiv Sp. z o.o. (LLC).

We would like to thank the state law enforcement agency of Rhineland-Palatinate for extraordinarily numerous and essential suggestions on the development of X-Ways Forensics and X-Ways Investigator.

Thanks to Dr. A. Kuiper for his method to process videos with MPlayer.

Professional users around the world include... (this list is from ~18 years ago)
U.S. and German federal law enforcement agencies, ministries such as the Australian Department of Defence, U.S. national institutes (e.g. the Oak Ridge National Laboratory in Tennessee), the Technical University of Vienna, the Technical University of Munich (Institute of Computer Science), the German Aerospace Center, the German federal bureau of aviation accident investigation, Microsoft Corp., Hewlett Packard, Toshiba Europe, Siemens AG, Siemens

Business Services, Siemens VDO AG, Infineon Technologies Flash GmbH & Co. KG, Ontrack Data International Inc., Deloitte & Touche, KPMG Forensic, Ernst & Young, Ericsson, National Semiconductor, Lockheed Martin, BAE Systems, TDK Corporation, Seoul Mobile Telecom, Visa International, Analytik Jena AG, and many other companies and scientific institutes.

1.2 Legalities

Copyright © 1995-2026 Stefan Fleischmann, X-Ways Software Technology AG. No part of this publication may be reproduced, or stored in a database or retrieval system without the prior permission of the author. Any brand names and trademarks mentioned in the program or in this manual are properties of their respective holders and are generally protected by laws. FuzZyDoc™ is a trademark of X-Ways Software Technology AG.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. However, the author neither offers any warranties or representations nor does he accept any liability with respect to the program or the manual.

License Agreement

Acknowledgements

The MD5 message digest is copyright by RSA Data Security Inc.

X-Ways Forensics contains software by Igor Pavlov, www.7-zip.com, and an Adler32 implementation by Arnaud Bouchez.

Outside In® Technology Copyright © 1991, 2019, Oracle Corp. and/or its affiliates. All rights reserved.

FuzZyDoc™ is a trademark of X-Ways Software Technology AG.

X-Ways Forensics uses ResIL, a fork of DevIL. ResIL is governed by the LGPL (http://www.gnu.org/copyleft/lesser.html), version 2.1. The source code can be downloaded from http://sourceforge.net/projects/resil.

X-Ways Forensics contains an unofficial build of libPFF. libPFF is governed by the LGPL (http://www.gnu.org/copyleft/lesser.html), version 3.0. The original source code can be downloaded from http://libpff.sourceforge.net/.

X-Ways Forensics uses Dokan. Dokan is governed by the LGPL (http://www.gnu.org/copyleft/lesser.html), version 3.0. The source code can be found at https://dokan-dev.github.io/.

X-Ways Forensics uses WimLib. WimLib is governed by the LGPL (https://www.gnu.org/licenses/lgpl-3.0.html), version 3. The source code can be downloaded from https://wimlib.net.

X-Ways Forensics uses lzfse. Copyright (c) 2015-2016, Apple Inc. All rights reserved. The source code can be downloaded from https://github.com/lzfse/lzfse. Copyright (c) 2015-2016, Apple Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of the copyright holder(s) nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

X-Ways Forensics uses Zstandard software (zstd). Copyright (c) Meta Platforms, Inc. and affiliates. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name Facebook, nor Meta, nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Windows event log (.evtx) viewing capability based on works by Andreas Schuster.

MiniZ: The MIT License. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

TinyXML: Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved. This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscapes SSL. This library is free for commercial and non-commercial use as long as the following conditions are aheared to. The following conditions apply to all code found in this distribution, be it the

RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com). Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)" The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related :-). 4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)". THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

Unicode: Copyright 2001-2004 Unicode, Inc. Disclaimer This source code is provided as is by Unicode, Inc. No claims are made as to fitness for any particular purpose. No warranties of any kind are expressed or implied. The recipient agrees to determine applicability of information provided. If this file has been purchased on magnetic or optical media from Unicode, Inc., the sole remedy for any claim will be exchange of defective media within 90 days of receipt. Limitations on Rights to Redistribute This Code Unicode, Inc. hereby grants the right to freely use the information supplied in this file in the creation of products supporting the Unicode Standard, and to make copies of this file in any form for internal or external distribution as long as this notice remains attached.

ZLib from http://zlib.net/: This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software. Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions: 1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required. 2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software. 3. This notice may not be removed or altered from any source distribution.

FFmpeg from https://www.ffmpeg.org/: Governed by the LGPL (http://www.gnu.org/copyleft/lesser.html), version 3.0.

1.3 License Types

You may evaluate WinHex free of charge, for at most 45 days. For regular use and for use as a full version, you need at least one license. For multiple users at the same time or use on multiple machines by one user at the same time, you will need multiple licenses. License agreement. Unlike the evaluation version, the full version of WinHex will save files larger than 200 KB, write disk sectors, edit virtual memory and show no evaluation version reminders. It will reveal its licensing status on start-up and in the About box (the window that appears when you click the version number in the upper right corner).

- Personal licenses are available at a reduced price for non-commercial purposes only, in a non-business, non-institutional, and non-government environment.

- Professional licenses allow usage of the software in any environment (at home, in a company, in an organization, or in public administration). Professional licenses provide the ability to execute scripts.

- Specialist licenses in addition to that allow to use Specialist menu commands, read the file systems exFAT, Ext2, Ext3, Ext4, CDFS/ISO9660, UDF, can highlight free drive space and slack space, enable support for RAID reconstruction, Windows dynamic disks, Linux LVM2, some more columns in the directory browser, and reverse disk cloning/imaging. Particularly useful for IT security specialists.

- WinHex Lab Edition in addition to that understands the file systems HFS, HFS+/HFSJ/HFSX, ReiserFS, Reiser4, XFS, BtrFS (multiple disks via LVM2 or RAID setups are supported, but not BtrFS multi-device setups), UFS1, UFS2, APFS (unencrypted), QNX, allow to create evidence file containers, and allow to run regular X-Tensions.

- Licenses for X-Ways Forensics ("forensic licenses") in addition to the above allow to use the powerful case managing and report generating capabilities, the internal viewer and the separate viewer component, the gallery view, many more volume snapshot refinement operations, many more columns and filters in the directory browser (and the order of the columns can be changed), comments and labels. Furthermore, they allow to read and write evidence files (.e01), the exploration of SquashFS file system (those based on GZIP/zlib, LZMA, LZO or XZ) and can do **much, much more**! Particularly useful for computer forensic examiners.

X-Ways Investigator is a simplified version of X-Ways Forensics. It does not have all the functionality of X-Ways Forensics, not even all the functionality of WinHex. Users of X-Ways Forensics can temporarily reduce the user interface of X-Ways Forensics to that of X-Ways Investigator to see if additional licenses for X-Ways Investigator would benefit their organization to split up the investigative workload across multiple users, some of them non-technical. X-Ways Investigator is not really meant as a stand-alone product.

The maximum number of simultaneous character sets in the text display also depends on the license type (cf. View menu). A more complete license type comparison can be found online at

http://www.x-ways.net/winhex/comparison.html. Please see http://www.x-ways.net/order.html on how to order your licenses. Thank you.

1.4 More differences between WinHex & X-Ways Forensics

WinHex (main executable file is winhex.exe or winhex64.exe) always identifies itself as WinHex in the user interface, X-Ways Forensics (main executable file xwforensics.exe or xwforensics64.exe) as X-Ways Forensics. The shared program help and the shared manual, however, statically refer to the name "WinHex" in most cases, sometimes "X-Ways Forensics".

WinHex and X-Ways Forensics share the same code base. X-Ways Forensics offers numerous additional forensic features over WinHex with a specialist license, but does not allow to edit disk sectors or interpreted images and lacks various functions to wipe data known from WinHex. In X-Ways Forensics, disks, interpreted image files, virtual memory, and physical RAM are strictly opened in view mode (read-only) only, to enforce forensic procedures, where no evidence must be altered in the slightest. This strict write protection of X-Ways Forensics ensures that no original evidence can possibly be altered accidentally, which can be a crucial aspect in court proceedings.

Only when not bound by strict forensic procedures and/or when in need to work more aggressively on disks or images (e.g. you have to repair a boot sector or wipe classified or unrelated data), then a user of X-Ways Forensics would run WinHex instead. With WinHex you can edit disk sectors and wipe entire hard disks, free space, slack space, selected files, selected disk areas etc.

Users of X-Ways Forensics may simply copy their xwforensics.exe executable file and name the copy winhex.exe (or for the 64-bit edition copy their xwforensics64.exe executable file and name the copy winhex64.exe) to get WinHex. The setup program creates such copies automatically. Or you can create hardlinks instead of copies (higher coolness factor). If the program is executed as *winhex*.exe, it will identify itself as WinHex everywhere (in the user interface, case report, case log, image descriptions, and all screenshots) and act/behave like WinHex. That version is the best of both worlds, with the full forensics feature set of X-Ways Forensics plus the sector editing and data wiping capability of WinHex in one.

1.5 Getting Started with X-Ways Forensics

For the latest download instructions, if your update maintenance is current, you can check your license status here. For more information about the installation of WinHex and X-Ways Forensics please see this web page.

Extract the files in the X-Ways Forensics download to a directory of your choice. An installation with the setup program is not necessary. The program is portable and can also be started directly from a USB stick on other computers, e.g. live systems that you would like to examine. Also download the viewer component (which is not included in the standard download as it is updated much more rarely). Use the 64-bit edition of the viewer component for the 64-bit edition of X-

Ways Forensics. By default, the viewer component is expected in the subdirectory \viewer (32 bit) or \x64\viewer (64 bit). Please be advised that the viewer component creates files in the profiles of the user who is currently logged on, unlike X-Ways Forensics, so if you wish to avoid to create files on a live system that you examine, don't let X-Ways Forensics use the viewer component. You may also wish to download MPlayer if you intend to have X-Ways Forensics produce stills from videos to see them in the gallery. Newer releases can always be extracted into the existing directory of an earlier release. You may continue to use WinHex.cfg configuration files from earlier releases in later releases (but never the other way around).

To associate a portable installation of X-Ways Forensics or X-Ways Investigator and its icon with .xfc case files on a particular machine, you could consciously run the application at least once explicitly as administrator and end it while any of the customizable standard paths is located on the same drive letter as your Windows installation, to give the application a hint that you are the owner of that Windows system and feel comfortable that data is written to it. That's either the path from where you run the application, the path where to create and expect case files, the path where to create and expect image files, or the path where to create temporary files.

Here are some instructions to help you get started and find some important features: Create a case, add an evidence object (such as your own C: drive or hard disk 0, or an image file). In the directory tree, you may use a right click to list the contents of a directory in the directory browser including all its subdirectories. For example, if you right-click the root directory of a volume, you will get a listing of all files in the entire volume. At the same time you can use a dynamic filter to focus on files based with certain filenames, of a certain file type, size, or with certain timestamps, etc. via Options | Directory Browser.

The powerful logical search functionality can be found in Search | Simultaneous Search. More interesting functions in X-Ways Forensics can be found in the context menu of the directory browser (e.g. the ability to copy files off an image) and in the Specialist menu, in particular “Refine Volume Snapshot”). The latter allows you to further process files automatically, e.g. explore zip archives, extract e-mail messages and attachments, check pictures for the amount of skin tones, check documents for encryption, etc.

There are a thousand different purposes for which X-Ways Forensics can be used, so in our opinion step-by-step instructions (click here first, then there, then look here) are not the right way to explain the software. This program help/user manual is rather meant to accurately describe all the available functionality and let you creatively combine different commands to achieve a certain goal. It is still the user who has to do the thinking, know what he/she is doing and how to interpret findings.

The 64-bit edition is recommended especially in situations where the 32-bit memory address space may be insufficient, when dealing with disks or images that contain many millions of files, or when dealing with many millions of search hits, provided that you have plenty of physical RAM installed. Certain operations that are computationally intensive (e.g. hashing or encrypting) may also be faster in the 64-bit edition.

2 Technical Background

2.1 Using a Hex Editor

A hex editor is capable of completely displaying the contents of each file type. Unlike a text editor, a hex editor even displays control codes (e.g. linefeed and carriage-return characters) and executable code, using a two-digit number based on the hexadecimal system.

Consider one byte to be a sequence of 8 bits. Each bit is either 0 or 1, it assumes one of two possible states. Therefore one byte can have one of 2•2•2•2•2•2•2•2 = 2^8 = 256 different values. Since 256 is the square of 16, a byte value can be defined by a two-digit number based on the hexadecimal system, where each digit represents a tetrade or nibble of a byte, i.e. 4 bits. The sixteen digits used in the hexadecimal system are 0-9, A-F.

You can change the value of a byte by changing these digits in the hexadecimal mode. It is also possible to enter the character that is assigned to a certain byte value by a character set (cf. Entering Characters). All kinds of characters are allowed (e.g. letters and punctuation marks). Example: A byte whose decimal value is 65 is displayed as 41 in hexadecimal notation (4•16+1=65) and as the letter A in text mode. The ASCII character set defines the capital letter A to have the decimal value of 65.

You have the option to toggle between the usual high-low nibble order within one byte in the hex display and the low-high order, by clicking the corresponding cell in the status bar, which shows either "BEN" for big-endian nibbles or "LEN" for little-endian nibbles. Or you can make use of a check box in the top right corner of the general options. Though uncommon and awkward for human reading habits, showing the lower nibble first and the higher nibble second is a logical choice where multiple bytes belong to an integer value stored in little-endian byte order because that way all hex digits are arranged in ascending order of power. The difference that this setting makes is apparent in particular when looking at the 12-bit entries in a FAT12 file allocation table, whose nibbles usually appear intertwined, but in low-high order are straight-forward and consecutive.

When editing files of a certain type (for instance executable files), it is essential not to change the file *size*. Moving the addresses of executable code and included data results in severely damaging such files. Please note that changing the contents of a file generally may be the reason for the corresponding application to behave anomalously. It is quite safe to edit text passages in a file. At any rate, it is recommendable to create backup files before editing.

The command “Combined Search” was especially designed for editing files created by computer games to save the game state. If you know the value of a variable in two of such files, you can find out the offset, i.e. the position, at which this data is saved. Example: If two files hold the information that you have 5 resp. 7 points/lives/..., search simultaneously for the hex value 05 in the first and 07 in the second file.

2.2 Endian-ness

Microprocessors differ in the position of the least significant byte: Intel®, MIPS®, National Semiconductor, and VAX processors have the least significant byte first. A multi-byte value is stored in memory from the lowest byte (the "little end") to the highest byte. For example, the hexadecimal number 12345678 is stored as 78 56 34 12. This is called the *little-endian* format.

Motorola and Sparc processors have the least significant byte last. A multi-byte value is stored in memory from the highest byte (the "big end") to the lowest byte. For example, the hexadecimal number 12345678 is stored as 12 34 56 78. This is called the *big-endian* format.

2.3 Integer Data Types

Format/Type	Range	Example
signed 8 bit	-128...127	FF = -1
unsigned 8 bit	0...255	FF = 255
signed 16 bit	-32,768...32,767	00 80 = -32,768
unsigned 16 bit	0...65,535	00 80 = 32,768
signed 24 bit	-8,388,608...8,388,607	00 00 80 = -8,388,608
unsigned 24 bit	0...16,777,215	00 00 80 = 8,388,608
signed 32 bit	-2,147,483,648...2,147,483,647	00 00 00 80 = -2,147,483,648
unsigned 32 bit	0...4,294,967,295	00 00 00 80 = 2,147,483,648
signed 64 bit	-2^{63} ($\approx -9 \cdot 10^{18}$)...$2^{63}-1$ ($\approx 9 \cdot 10^{18}$)	00 00 00 00 00 00 00 80 = -2^{63}

Unless stated otherwise, multi-byte numbers are stored in little-endian format, meaning that the first byte of a number is the least significant and the last byte is the most significant. This is the common format for computers running Microsoft Windows. Following the little-endian paradigm, the hexadecimal values 10 27 can be interpreted as the hexadecimal number 2710 (decimal: 10,000).

The Data Interpreter is capable of interpreting data as all of the aforementioned integer types, plus unsigned 48-bit integers.

2.4 Floating-Point Data Types

Type	Range	Precision [Digits]	Bytes
Float (Single)	$\pm 1.5^{-45}...3.4^{38}$	7-8	4
Real	$\pm 2.9^{-39}...1.7^{38}$	11-12	6
Double (Double)	$\pm 5.0^{-324}...1.7^{308}$	15-16	8
Long Double (Extended)	$\pm 3.4^{-4932}...1.1^{4932}$	19-20	10

The type names originate from the C programming language. The corresponding Pascal names

are specified in brackets. The Real type exists only in Pascal. The Data Interpreter is capable of translating hex values in an editor window into floating-point numbers of all four types and vice-versa.

In the computer, a floating-point number F is represented by a mantissa M and an exponent E, where $M \times 2^E = F$. Both M and E are signed integer values themselves. The four data types differ in their value ranges (i.e. the number of bits reserved for the exponent) and in their precision (i.e. the number of bits reserved for the mantissa).

On Intel®-based systems, calculations upon floating-point numbers are carried out by a math coprocessor while the main processor waits. The Intel® 80x87 uses 80-bit precision for calculations, whereas RISC processors often use 64-bit precision.

2.5 Date Types

The following date formats are supported by the Data Interpreter:

- **MS-DOS Date & Time (4 bytes)**

The lower word determines the time, the upper word the date. Used by several DOS function calls, by the FAT file systems and many system utilities such as file archivers.

Bits	Contents
0-4	Second divided by 2
5-10	Minute (0-59)
11-15	Hour (0-23 on a 24-hour clock)
16-20	Day of the month (1-31)
21-24	Month (1 = January, 2 = February, etc.)
25-31	Year offset from 1980

- **Win32 FILETIME (8 bytes)**

The FILETIME structure is a 64-bit integer value representing the number of 100-nanosecond intervals since January 1, 1601. Used by the Win32 API.

- **OLE 2.0 Date & Time (8 bytes)**

A floating-point value (more exactly: a double) whose integral part determines the number of days passed since December 30, 1899. The fractional part is interpreted as the day time (e.g. 1/4 = 6:00 a.m.). This is the OLE 2.0 standard date type, e.g. it is used by MS Excel. ICQ 7.0 uses big-endian OLE 2.0 timestamps in chat messages

- **ANSI SQL Date & Time (8 bytes)**

Two consecutive 32-bit integer values. The first one determines the number of days since

November 17, 1858. The second one is the number of 100-microsecond intervals since midnight. This is the ANSI SQL standard and used in many databases (e.g. InterBase 6.0).

- **UNIX, C, FORTRAN Date & Time (4 bytes)**

A 32-bit integer value that determines the number of seconds since January 1, 1970. This data type was used in UNIX, by C and C++ ("time_t"), and by FORTRAN programs since the 80's. Sporadically defined as the number of *minutes* since January 1, 1970. The Data Interpreter options let you switch between both sub-types.

- **Mac HFS+ Date & Time (4 bytes)**

A 32-bit integer value that determines the number of seconds since January 1, 1904 GMT (HFS: local time). The maximum representable date is February 6, 2040 at 06:28:15 GMT. The date values do not account for leap seconds. They do include a leap day in every year that is evenly divisible by 4.

- **APFS Date & Time (8 bytes)**

- **Java Date & Time (8 bytes)**

A 64-bit integer value that specifies the number of milliseconds since January 1, 1970. Usually stored in big endian, which is the typical byte order in Java, but in little endian in BlackBerry memory.

- **Mac Absolute Time, a.k.a. Mac epoch time (4 bytes)**

A 32-bit integer value that determines the number of seconds since January 1, 2001.

2.6 ANSI ASCII/IBM ASCII

ANSI ASCII is the name utilized in WinHex for an extension of the ASCII character set as used in non-Unicode Windows applications. It was named ANSI by Microsoft after the American National Standards Institute, but not defined by that institute. Several different regional variants exist, one of which is active in Windows, typically code page 1252 in countries where a Western European language is spoken. MS-DOS and Windows command prompt windows use what is called the IBM ASCII character set in WinHex (also called OEM or DOS character set elsewhere). All of these 8-bit extensions of the 7-bit ASCII character sets differ in the characters with values greater than 127. If for example if you store plain text file with Windows Notepad in ANSI encoding and later view it with the type command in a command prompt window, special characters such as German umlauts will not be displayed correctly. Some of the regional ANSI code pages are double-byte code pages, i.e. use even 2 bytes for some characters instead of just 1 per character.

Select the character set for the text column in the View menu, or click the top of the text column, where the name of the active code page/character set is displayed to change settings. Use the

"Convert" command of the Edit menu to convert text files from one character set to the other.

The first 32 ASCII values do not define printable characters, but control codes:

Hex	Control Code	Hex	Control Code
00	Null	10	Data Link Escape
01	Start of Header	11	Device Control 1
02	Start of Text	12	Device Control 2
03	End of Text	13	Device Control 3
04	End of Transmission	14	Device Control 4
05	Enquiry	15	Negative Acknowledge
06	Acknowledge	16	Synchronous Idle
07	Bell	17	End of Transmission Block
08	Backspace	18	Cancel
09	Horizontal Tab	19	End of Medium
0A	Line Feed	1A	Substitute
0B	Vertical Tab	1B	Escape
0C	Form Feed	1C	File Separator
0D	Carriage Return	1D	Group Separator
0E	Shift Out	1E	Record Separator
0F	Shift In	1F	Unit Separator

2.7 Checksums, Hashes, Digests

A checksum is a characteristic number used for verification of data authenticity. Two files with equal checksums are highly likely to be equal themselves (byte by byte). Calculating and comparing the checksums of a file *before* and *after* a possibly inaccurate transmission may reveal transmission errors. An unaffected checksum indicates that the files are (in all likelihood) still identical. However, a file can be manipulated on purpose in such a way that its checksum remains unaffected. Digests are used instead of checksums in such a case, where malicious (i.e. not mere random) modifications to the original data are to be detected.

In WinHex, checksums can be calculated for example with a command in the Tools Menu.

The standard checksum is computed as a sum by interpreting data as a sequence of integer numbers, calculated on an 8-bit, 16-bit, 32-bit, or 64-bit accumulator. Alternative checksums use a multi-byte accumulator, but strictly 8-bit integer additions only. The CRC (cyclic redundancy code) is based on more sophisticated algorithms, which are safer.

Example: If a transmission alters two bytes of a file in such a way that the modifications are countervailing (for instance byte one +1, byte two -1), the standard checksum remains unaffected, whereas the CRC changes.

A so-called hash is, similar to a checksum, a characteristic number used for verification of data authenticity. But hashes are more than that: digests are *strong one-way hash codes*.

It is computationally feasible to manipulate any data in such a way that its checksum remains unaffected. Verifying the checksum in such a case would lead to the assumption that the data has

not been changed, although it has. Therefore, hashes are used instead of checksums if malicious (i.e. not mere random) modifications to the original data are to be detected. It is computationally infeasible to find any data that corresponds to a given hash. It is even computationally infeasible to find two pieces of data that correspond to the same hash.

Of course, random modifications, e.g. caused by an inaccurate transmission, can also be detected when using hashes, but checksums are sufficient and serve better for this purpose, because they can be calculated much faster.

WinHex can run the following hash algorithms: MD4, MD5, SHA-1, SHA-256, SHA-512, RipeMD-128, RipeMD-160, Tiger128, Tiger160, Tiger192 as well as TTH (Tiger Tree Hash) and ed2k (specialist and forensic licenses only).

2.8 Attribute Legend

A: to be archived
R: read-only
H: hidden
S: system
X: not indexed
P: NTFS reparse point
O: offline
T: temporary
I: has object ID
C: compressed at filesystem level
c: compressed in archive
E: encrypted at filesystem level
e: encrypted in archive
e!: file type specific encryption/DRM
e?: high entropy, possibly fully encrypted
(Res): HFS+ resouce
($EFS): NTFS encryption metadata
(INDX): NTFS non-directory index attribute
(ADS): NTFS alternate data stream
(SC): found in a volume shadow copy
(SUID): Set User ID
(SGID): Set Group ID

File mode:
l=symbolic link
c=character device
b= block device
s=socket
p=pipe

Permissions:

owner read/write/execute
group read/write/execute
other read/write/execute

2.9 Technical Hints and Limitations

- Technical specifications

Supported disk size:	at least 131 TB
Supported partition and volume size:	131 TB-1 byte
Supported file size in volume snapshots:	at least 120 TB-1 byte
Maximum number of sectors generally:	2^{40}-1
Maximum number of clusters generally:	2^{32}-1
Maximum number of hash values per hash database:	2^{31}-1
Maximum number of values in the PhotoDNA database (64 bit):	~58.8 million
File system supported for volumes > 2^{32} sectors:	NTFS, Ext*, XFS, Reiser*
File system supported for volumes > 2^{32} clusters:	NTFS, Ext4, XFS
Addressable physical storage devices as numbered in Windows:	0-127
Maximum number of simultaneously open interpreted disk images	100
Maximum number of simultaneously open partitions and interpreted volume images	256
Supported number of items in a volume snapshot (64 bit)	>536 million
Maximum number of search terms in a case	8191
Maximum number of search hits per evidence object	211,969,638
Maximum number of data windows:	1000
Maximum number of simultaneous program instances:	99
Maximum number of reversible keyboard inputs:	65535
Encryption depth:	128-256 bit
Offset presentation:	hexadecimal/decimal

- In most cases, the progress display shows the completed percentage of an operation. However, during search and replace operations it indicates the relative position in the current file or disk.

- Keys you specify for encryption/decryption are not saved on the hard disk. Provided that the corresponding security option is enabled, the key is stored in an encrypted state within the RAM, as long as WinHex is running.

- Search and replace operations generally run fastest with case sensitivity switched on and without wildcards enabled.

- When searching with the option “count occurrences” activated or when replacing without prompting, for a search algorithm there are generally two ways to behave when an occurrence has been found, which in some cases may have different results. This is explained by the following example:

  The letters *ana* are searched in the word “banana”. The first occurrence has already been found at the second character.

1st alternative: The algorithm continues the search at the third character. So *ana* is found again at the fourth character.
2nd alternative: The three letters *ana* found in the word “banana” are skipped. The remaining letters *na* do not contain *ana* any more.

WinHex is programmed in the second manner, because this delivers the more reasonable results when counting or replacing occurrences. However, if you continue a search using the **F3** key or you choose the replace option “prompt when found”, the algorithm follows the first paradigm.

Special Performance Enhancements

File header signature searches, block-wise hash matching, FILE record searches, searches for lost partitions, and physical simultaneous searches are sparse-aware operations when dealing with certain compressed and sparse .e01 evidence files. That means that areas that on the original hard disk were never written and thus still zeroed out or areas that had been wiped on the original hard disk or consciously omitted areas in cleansed images are skipped and almost require no time, because their data neither has to be read nor decompressed nor further processed (searched/hashed/matched against the block hash database).

Sparse-awareness is active for .e01 evidence files that were created by X-Ways Forensics and X-Ways Imager with a chunk size of 32 KB, 128 KB or 512 KB. Also possibly for images created by 3rd party software, depending on the settings and the internal layout. Operations are not sparse-aware on images of Windows dynamic disks, images of LVM2 disks, and on reconstructed RAIDs based on .e01 evidence files.

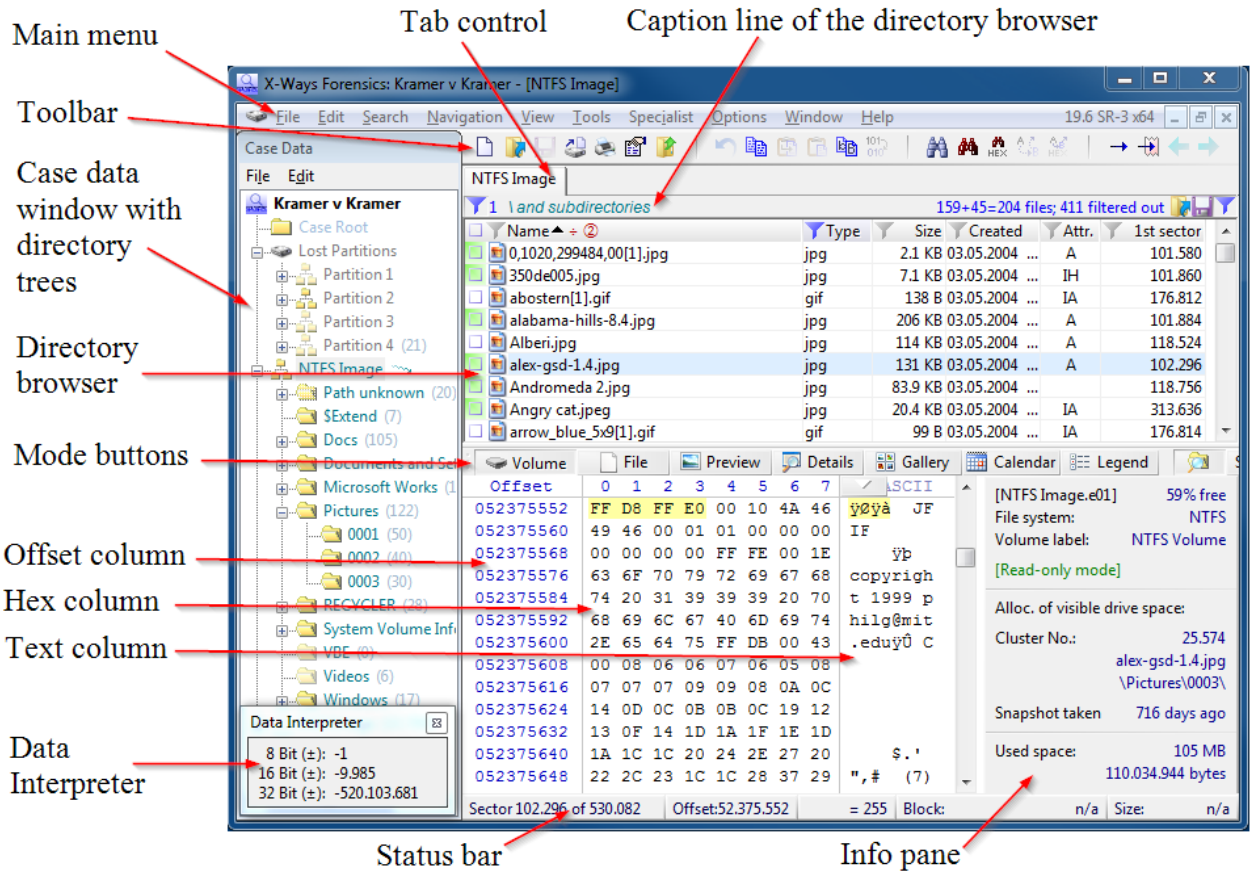
Logical searches and indexing in files stored in an NTFS file system are also sparse-aware at the .e01 evidence file level, and generally logical searches in virtual "Free space" files.

Logical searches and indexing in NTFS, Ext*, XFS and UFS file systems are sparse-aware at the file system level. That means no time is wasted on large sparse areas within sparse files. Those areas are ignored, regardless of whether the evidence object is an .e01 evidence file, raw image, RAID, or actual disk.

3 User Interface

3.1 Overview

To familiarize yourself with the names of the various elements of the user interface, please refer to this screenshot:



3.2 Start Center

The so-called Start Center is a dialog window that is optionally displayed at startup and is meant as a simplified control panel for beginning your work. It allows to quickly open files, disks, memory modules, and folders as well as up to 255 recently edited documents (16 by default, left-hand list). These may be files, folders, logical drives or physical disks. When opened again, WinHex restores the last cursor position, the scrolling position, and the block (if defined) of each document, unless the corresponding option is disabled.

From the Start Center you are also able to access *projects* and *cases* (right-hand top list). A project consists of one or more documents to edit (files or disks). It remembers the editing positions, the window sizes and positions and some display options. By saving a window arrangement as a project you can continue to work in several documents right where you left them, with a single click only. This is especially useful for recurring tasks. When you load a project, all currently opened windows are automatically closed first.

Besides, WinHex automatically saves the window arrangement from the end of a WinHex session as a project, and can re-create it next time at startup. Each project is stored in a .prj file. It can be deleted or renamed right within the Start Center (context menu or DELETE/F2 key).

Last not least, the Start Center is the place where to manage *scripts*. You may check, edit, create,

rename, and delete scripts using the context menu. To execute a script, double-click it or single-click it and click the OK button.

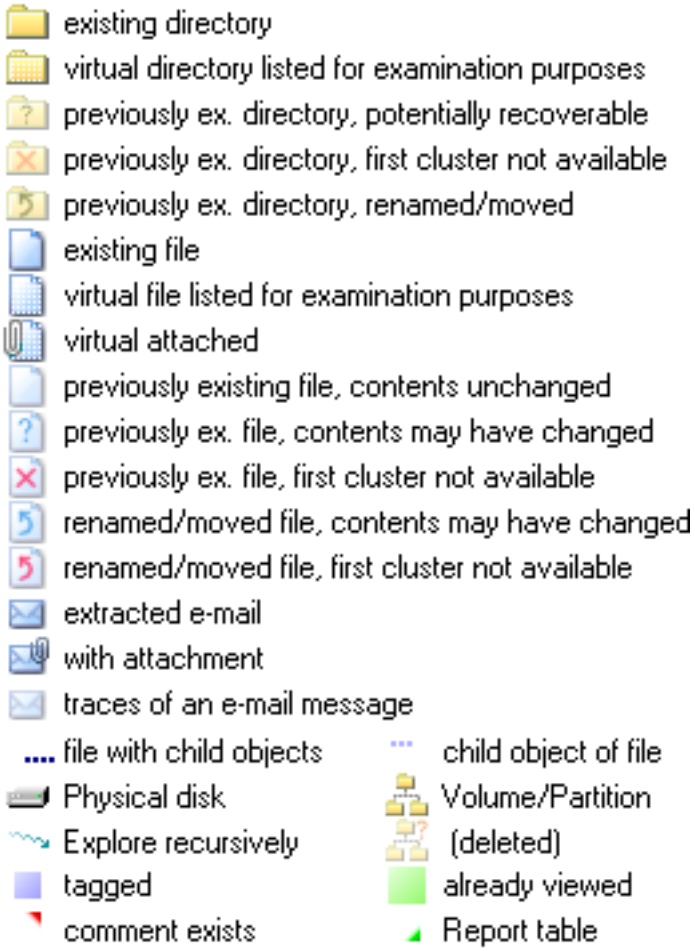
3.3 Directory Browser

3.3.1 General Description

The perhaps most essential user interface element in WinHex and X-Ways Forensics is the so-called *directory browser*, which resembles the Windows Explorer's right-hand list. Its main task is to display (and interact with) the volume snapshot. Complete functionality is available only with a forensic license. By default, the directory browser lists directories first, then files. Compressed files are displayed in blue, encrypted files in green. Right-clicking any item in the directory browser brings up a context menu with commands for opening a file or directory, exploring a directory, locating the beginning of a file or directory on the disk, locating the corresponding directory entry (FAT) or file record (NTFS), listing the allocated clusters in a separate window, etc.

When navigating from one directory to another, exploring files with child objects (e.g. e-mail messages that have attachments), navigating to the parent of a child object, activating or deactivating filters, trying different sort criteria etc., please note that you can easily return to a previous view using the Back command in the Position menu or the Back button in the toolbar.

The **icons** are explained in the legend directly in the program (forensic license only). Deleted files and directories are represented in the directory browser with lighter icons. Icons with a blue question mark indicate that the original file or directory contents may be still available. Deleted objects that WinHex knows are no longer accessible (either because their first cluster has been reallocated, because it is unknown, or because they have a size of 0 bytes) have icons crossed out in red. Icons with an arrow on FAT volumes (only with a specialist or forensic license) and (after refining the volume snapshot) NTFS volumes show renamed and moved files with their original name/in their former directory. On Reiser4 these are moved files with their current name in their former directory. A blue arrow indicates that contents for a file are available (though these are not specifically the contents from before the file was renamed or moved). A red arrow indicates that no contents are available.



In the caption line of the directory browser you see on the left the explored path (in case of recursive exploration in italics and turquoise color). When clicking any component of the current path, this will now navigate directly to that directory (or file with child object) whose name you clicked. On the right you see the number of listed files and directories (typically separate figures for existing objects + previously existing objects + virtual objects). Also, the number of listed tagged files is indicated, if any are tagged. The number of active filters is displayed as well, next to the blue filter symbol on the left. Column-based and column-independent active filters are counted separately. Useful because there might be column-based filters active for columns that are not currently visible in the directory browser, and that column-independent filters are active may be otherwise apparent only when checking in the directory browser options dialog.

The directory browser can **sort** files and directories in ascending or descending order, and still reveals the two previous sort criteria with a lighter arrow. For example, if you first click the filename column and then the filename extension column, files with the same extension will internally still be sorted by name. In order to undefine the secondary and tertiary sort criteria, hold the Shift key when clicking on the column header to determine the primary sort criterion. Internally, this selects the internal ID as the secondary sort criterion. This is to ensure that the order of items with identical data for the primary sort criterion is still well defined and reproducible after having sorted by other sort criteria in the meantime. No matter how you sort, the virtual files at the volume or disk level (files that cover free space, volume slack, unpartitioned space etc.) are always listed at the bottom because they are better deal with

separately from ordinary files.

The column that functions as the primary sort criterion is also the target of “jump as you type”. That is, you can type the first character or first few characters of the entry that you are looking for when the directory browser has the focus to automatically navigate and select the first or next matching item in the list, starting from the current position. For example, if the directory browser is sorted by the Type column, type “z” if you wish to find the first zip file in the list. If however there is another file listed with a type starting with “z”, one that precedes “zip” alphabetically, for example “zac”, then type the next character (before the feature times out and forgets the “z” that you have already entered), in this case “i”, until you find what you are looking for or nothing happens any more (if there is no matching item). Matching occurs in a cycle. That means even if the current position shows a zip file, you can type any preceding letter to jump to the first matching item from the top again, for example “d” for .docx. If you are looking for .docx files, but find a large group of .doc files, then you need to type all four characters of docx, because only the “x” distinguishes docx from doc.

By clicking the floppy disk icon on the right-hand side of the directory browser caption line you can save directory browser settings to a .settings file. You have a great amount of control over what is actually remembered and what will be restored when loading that file. For example you can have .settings files with regularly used filter settings, or other files with just your preferred column order and widths that do not activate or deactivate any column-based filters and do not make you lose your current filter settings. Or .settings files with just your favorite conditional cell coloring that do not override column widths. Or all settings *except* conditional cell coloring etc. Your favorite .settings files can be loaded with a single mouse click. That is possible if you name them with a single-digit number or letter or any exotic single Unicode symbol, followed optionally by a space and more descriptive text, e.g. "1.settings" or "E Most essential columns only.settings" or "Ⓟ Default filters for pictures.settings" etc. That first character will be displayed in the caption line of the directory browser and will be clickable. If multiple .settings files exist whose naming matches the requirements, their respective first characters will be listed in ascending order of Unicode values. Other .settings files can be selected if you simply click the icon for that, next to the floppy disk.

3.3.2 Virtual Objects

When orphaned objects are found, e.g. files that have been deleted and whose original path cannot be determined by the program, or generally when the original path is unknown to the program, such objects are listed in a special virtual directory “Path unknown”.

With a specialist or forensic license, there are virtual files in the root directory that allow you to conveniently address special areas in a volume. These are always grouped at the bottom of the list:

File system areas: Reserved sectors and/or clusters that are claimed by the file system itself for internal purposes.

Free space: Clusters marked by the file system as not in use. Depends on the volume snapshot

options.

Idle space: Areas in a volume of which WinHex does not know what they are used for, including clusters marked by the file system as in use, whose exact allocation however could not be determined. This can be the case if the file system lost track of them, i.e. forgot that these cluster are actually available for re-allocation. Usually there is no idle space. The size of idle space and the number of the first idle cluster are only determined when needed (e.g. when you click the "Idle space" file for the first time), as depending on the number of cluster this is a potentially time-consuming operation.

Volume slack: Sectors at the end of the partition that are unused by the file system because they do not add to another cluster.

Indirect blocks (Ext2, Ext3, UFS): Special blocks that contain block numbers. Not part of "File system areas".

Unnoted attribute clusters (NTFS): Clusters that contain non-resident attributes that have not been individually processed by X-Ways Forensics. Not part of "File system areas".

.journal (ReiserFS): Blocks that form the fixed journalling area. On Ext3 and HFS+, this is not considered a virtual file because it is defined by the file system itself in dedicated records.

3.3.3 Partition Layout

The partition layout of physical storage devices is depicted graphically below the list of partitions in the directory browser. The horizontal locations and widths of all partitions are directly proportional to the capacity of the entire storage device. It is not guaranteed that every partition will be visible because tiny partitions on a very large storage device might turn out just a few pixels wide or even rounded down to a width of 0 pixels because the representation is truly proportional and unbiased. If a suspect has set aside a dedicated partition for unlawful or suspicious content, the capacity chosen is not inflated or minimized in the depiction just for the sake of easy clickability of all partitions.

Partitions/volumes that are not referenced in any active partition table (usually deleted partitions) are presented in a lighter color. Partitions manually defined by the user (of WinHex/X-Ways Forensics) are depicted in a different color to make them stand out more. Areas that are not occupied by any partition are shown as hollow, with dotted outlines.

Thanks to simple 3D rendering and the angle, you can still see the full width (i.e. true size) of partitions even if they partially intersect with other partitions because those are set apart. The overlapping of partitions is problematic because the question may arise which data in the affected disk area logically belongs to which partition. The depiction is intended to alert users of this issue. On Windows LDM disks, for dynamic volumes that consist of multiple discontiguous storage space fragments (on potentially more than one physical storage devices), only the start locations are hinted at, where their names appear, along with the word "spanned". The other fragments of such volumes are not shown.

The partition layout depiction responds to mouse-over events, left-clicks, right-clicks and double-clicks. Large rectangles are more convenient to target with the mouse than narrow rows in the directory browser, so this feature addition may naturally change the way you explore partitions.

3.3.4 Filtering

You may activate filters based on criteria (columns) such as filename, description, file type category, attributes, or hash set. Whenever an active filter actually filters out files or directories in the directory browser, this is flagged with a blue filter icon in the directory browser's header line, and you will be informed of how many items exactly have been omitted from the list. You also have the option, by clicking the icons for "open file"/"save file" on the right-hand side of the caption line of the directory browser, to store filter and sort settings in a separate file and load them again at any time. Such files are given the extension ".settings". Note that it is not guaranteed that different versions of the software can load each other's settings.

You have the option to load multiple .settings files at the same time, which each can target different files using different filters (internally combined with AND or OR), and all resulting files will get a particular label. This allows for complex nested filter conditions like this: Files of type A only if contained in path X plus files of type B if not deleted plus files whose names contain the word Y or Z and who have the System attribute etc. etc. A filter for the resulting label is automatically activated.

Whenever one or more filters are active that actually filter out items in the currently displayed directory browser, there are two blue filter symbols in the directory browser's caption line. They point out that your current view is incomplete because of active files, and they also allow you to deactivate *all* filters with a single mouse click, to ensure you are not missing any file when you no longer want the filter. You can activate or deactivate column-based filters individually with a single mouse click on the column header's filter symbol when holding the Shift key. The options of the respective filter remain unchanged in this case.

The filters have been given some "intelligence" when navigating from a parent file to a child file or vice-versa, so that the filters "know" when it's a good time to be turned off.
For example:
- If you are using a filter to focus on all extracted e-mail messages recursively, and then you double-click an individual e-mail message to have a look at its attachments in the directory browser, the filter is automatically deactivated, so that you can actually see these attachments. A simple click on the Back button returns to the previous point of exploration and restores the previous filter settings and the last selection, so that you can easily continue reviewing the next e-mail message!
- If you are using a filter to focus on videos or documents, and then you double-click a video or a document to see the video stills exported for that video or the embedded pictures in that document, respectively, the filter is automatically deactivated, too.
- When you are viewing video stills only, in a gallery, and you use the Backspace key or "Find parent object" menu command to navigate to the video that this still belongs to (e.g. in order to play that video), then any active filters will be turned off so that the video can actually be listed.

A simple click on the Back button returns to the previous overview of stills, enables the previous filters again, and restores the last selected item, so that you can easily continue with the next still! - This works analogously when systematically looking at e-mail attachments, if occasionally for relevant attachments you would like to view the containing e-mail message (and e.g. print it or include it in a report) and then return to the list of attachments.

3.3.5 Columns and Filters

Most filters and many columns are available with higher license types only, marked with e.g. [FOR].

Name

Name of the listed file or directory and (only with a forensic license, only for directories and files with child objects) in parentheses in a different color optionally the total number of contained files in the volume snapshot.

If an original name is found for a file in the Windows recycle bin or in an iPhone backup or certain other files during metadata extraction, that name is displayed in the Name cell with the alternative/current unique name in square brackets (or vice versa, depends no your settings). The alternative name is now also shown in square brackets in the case report. Both names are targeted by the Name filter. If you may find some small triangles in the bottom right corner of a Name column cell. Those remind you that labels exist for the respective file. Whether the triangles are displayed or not depends on the Notation settings. The colors of the triangles depend on whether labels were automatically generated or user-created. In the top right corner of a Name cell yellow post-it icons indicate whether a comment was entered about a file. Hovering the mouse cursor over that icon immediately reveals the comment. If a file is destined to appear in the case report because it was assigned to a label that is includable in the report as a report table, that file is marked with a special icon in its name cell. That icon for the report is displayed in a fainter color if the label is not currently selected for output in the report options.

The header of the Name column allows to quickly tag or untag all listed items with a single mouse click. It also indicates whether among the listed items are any tagged or untagged items.

Allows to filter based on one or multiple filename masks, one per line. This filter is useful if you have a list of relevant filenames or keywords and want to find out quickly whether files with such names are present. There are two different ways how to use the Name filter. The first way is to match certain expressions against the full name. The expressions may contain asterisks (wildcards), like "*.jpg". Up to two asterisks are allowed per mask if they are located at the beginning and the end of it. You may exclude files using file masks that start with a colon (:). Example: All files with names that start with the letter "A", but do not contain the word "garden": "A*" in one line and ":*garden*" in another. When multiple positive file mask expressions are used, they are combined with a logical OR, negative expressions (:) with a logical AND. If the "Substring search in filename" option is

active, then all the rules above do not apply. Instead, a search is run within the filenames for the specified characters or optionally regular expressions. For example, just type "invoice" to find files whose filename contains the word invoice, not "*invoice*". For an explanation of regular expressions please see Search Options. The anchor $ does not work in this context. The amount of text that can be pasted into the Name filter is 2 million characters. That doesn't mean that X-Ways Forensics can efficiently use a filter with many ten thousands of characters or more. When in doubt, use the "Match against full name" option, not the substring search, and "Match case", for better performance.

Existent — Shows whether a file is an existing file or a child object of an existing file or not (existing based on its point of reference, e.g. file system), either with a check mark or a mathematical symbol or in natural language, depending on the Notation options. A third state is "virtual". To filter for the existence status, please use the Description filter. Remember you can group files by existence status using the directory browser options, or you can sort by this column.

Description — Textual description of the item. Reveals similar properties as the icon in the Name column, such as whether the item is a file or directory or extracted e-mail or video still etc., the existence/deletion/virtual/carved status, and the status in the volume snapshot (e.g. tagged, already viewed). What text is included in the column can be customized in the Notation options (via General Options). That the settings of the Description column are part of the Notation Options means that you can have two different settings, one generally for the directory browser and the other one specifically for the the Export List command. This might be useful because in the exported list no icon can help you to tell certain object types and their deletion status apart, unlike in the directory browser.

This column also allows to filter or sort by the properties covered, which makes the Description filter one of most important filters. For example you can filter out:

- existing files (useful if you are merely interested in previously existing files [which could reside in existing directories])
- previously existing files and directories.
- tagged files and directories.
- half tagged files and directories (that contain at least 1 tagged and at least 1 untagged file).
- untagged files and directories.
- files that are marked as already viewed.
- files that are *not* marked as already viewed.
- excluded files and directories (marked as excluded in the volume snapshot).
- files and directories that are *not* excluded.

There is a shortcut to get to the filter dialog very quickly, by right-clicking the caption line of the directory browser. This works even if the Description column is not visible. (You may not need the Description column in the directory browser if you rely on the icon to tell apart different kinds of items.) The funnel symbol that represents the filter of the Description column has four possible colors: 1) Gray when inactive, as usually. 2) Gray with a very, very light tendency to blue, almost

indistinguishable from gray, when the filter is on theoretically, but only excluded files would be filtered out, but no excluded files are actually getting filtered out currently. 3) Blue-gray when only excluded files are filtered out by the filter, and such files have actually been filtered out. 4) Ordinary blue to attract attention if the Description filter is active and does not only focus on excluded files, but filters out files based on other properties. This subdued color scheme was introduced because many user consider it rather "normal" that excluded files are filtered out because they exclude them for the very purpose of not seeing them any more, so they may prefer not to be reminded of that by a glaring blue color.

The filter for still images from videos has a special option that allows to also list the corresponding video, directly preceding its stills. That way it is easy to see which still images belong to which video, and you can comment on the video or attach a label to the video without navigating back and forth and without using the slightly less intuitive way to apply labels to an item that you cannot see (with the "for parent file" option). The tiles that represent the videos may act as visual delimiters in the gallery if you disable auxiliary thumbnails in the gallery options, so that you can easily see where still images of the next video begin.

The filter also allows to focus on carved files in general, and in particular carved files that are either aligned at sector boundaries or not, for example after having run a file header signature search at the byte level, to remove garbage files, which are more frequent among files that are not aligned. You can also focus on files from which text was extracted for logical searches (by OCR or decoding), with a certain minimum number of characters (like 5 or 10, 255 at most), for example to avoid pictures in which a few characters have been recognized merely erroneously, i.e. pictures that not actually do contain text.

A special filter setting is available that allows you to focus on files whose creation date is later than the modification date, i.e. which apparently were copied and that way got a new creation date. The Notation options allow to mark all such files with the word "copied". The presence of that word can be used for conditional cell coloring, so that you quickly see which files are likely original files and which files were copied. Note that a search for the word "copied" is language-specific (in case you share your conditional cell coloring settings with users in other countries).

Ext. Filename extension. The part of the filename that follows the last dot, if any, except if the last dot is the very first character (not uncommon in the Unix/Linux world).

Type [INV, FOR] File type. If the header signature of a file was not specifically checked (see Refine Volume Snaphot), this is merely a repetition of the filename extension and displayed in gray. Otherwise, if the file signature verification revealed the true nature of the file, a typical extension of that type will be output. That extension will be displayed in black if it is still the same as the actual extension of the file, or in blue if the actual extension does not match the type of the file. A convenient filter can be activated based on this column. In the filter dialog you can select individual file types or entire categories. You can load and save your selection. There are buttons that allow to expand or collapse all categories at once. Expanding all

categories can be useful if you would like to quickly find a certain file type by typing its letters while the tree view window has the input focus.

Please note that collisions among file type designations become apparent when selections for the file type filter are loaded from .settings files or cases. For example if you had originally selected "mmf" = "MailMessage File" (category e-mail), then you will find that "mmf" is also selected as "Yamaha SMAF" (category Sound/Music). This is normal and does not change what the Type filter does. When in doubt, the Type filter also includes other types with the same designation, to avoid that anything is overlooked.

Type status [INV, FOR]

The **status** of the Type column. Initially “not verified”. After verifying file types based on signatures (as part of refining the volume snapshot or viewing files in preview or gallery mode): If a file is very small (less than 8 bytes), the status is “irrelevant”. If neither the extension nor the signature of a given file is known to the file type signature database or an internal algorithm, the status is “not in list”. If the signature matches the extension according to the database or if an internal algorithm agrees that the extension fits the file type, the status is “confirmed”. If the filename extension is known from the database, yet neither the signature definition nor an internal algorithm can identify the file type, the status is “not confirmed”. If the file type is identified and the filename has no extension or a non-meaningful extension like .dat or .tmp, or if a more standardized type designation is set, then the status is “newly identified”. If the signature matches a certain file type in the database or an internal algorithm identifies a certain file types, however the extension is that of a *different* file type, then the status is “mismatch detected”. Filter available.

Additionally, this column may contain a hint about the **consistency** of the format of files of various supported types as either "OK", "irregular" or "corrupt", for carved files perhaps immediately, for other files perhaps after file type verification or metadata extraction have taken place. "Irregular" can mean incomplete, inconsistent, unexpected, not viewable, ... anything out of the ordinary. For example in the case of JPEG irregular could mean that no footer signature was found at the end of the file.

For an explanation of file type ranks and groups please see the description of File Type Categories.txt.

Type description [INV, FOR]

Displays the name of the application that a file type belongs to, what the filename extension stands for, etc. as specified in File Type Categories.txt. If the same extension occurs multiple times in the definition file, all its meanings are listed. For example, .pm could be a Perl module, a PageMaker document, or Pegasus file, or an X11 Pixmap file.

Category [INV, FOR]

File type category corresponding to the file type, according to the definition in “File Type Categories.txt” (see below). Filter available. You have the option to select multiple file type categories for filtering instead of just one, in a dialog window instead of a (quicker to use) pop-up menu. If the same file type/extension is defined multiple times, belonging to different categories, only one category for this file type will be displayed. The category filter works nonetheless. The category filter can be

	activated using a popup menu. In that popup menu you can also see statistics about how many files of each category are currently listed in the directory browser (or would be listed if the category filter was turned off).
Evidence object [INV, FOR]	The name of the evidence object that the file or directory is part of. Useful in a recursive case root listing, i.e. when the directory browser shows all files of all evidence objects. Sorting by this column sorts by the evidence object number, which you can see in the evidence object properties. That number usually depends on the position of the evidence object in the case tree.
Path	Path of the file or directory, starting with a backward slash, based on a volume's root. Filter available. The filter expressions are interpreted as substrings that can match any part of the path, so no wildcards are needed or supported.
Full path [SPE, LAB, FOR]	The path including the name of the file or directory itself. Sorting by full path can yield a convenient order because child objects directly follow their respective parents, even if some parent files or directories or e-mail messages have the exact same name. Filter available.
Parent name, Child objects [INV, FOR]	Both columns come with filters. The filter for child object allows you for example to quickly find all e-mails that have an attachment with a certain name. The filter for parent name for example allows you to quickly find all attachments that were attached to e-mail with a subject that contains certain words. Note that filters for the columns Name, Parent name, and Child objects share the same settings and are mutually exclusive (cannot be active at the same time, one will deactivate the other).
Size	Logical size of the file (i.e. size without slack) or physical size of a directory. Physical file size and valid data length (for files stored in an NTFS file system) can be seen in the Info Pane in File mode instead. If recursive selection statistics are enabled, with a forensic license the size of a directory is the total size of all the files directly or indirectly contained in that directory, otherwise the size of the data structures of the directory. Filter available. To focus specifically on files with an unknown size, use the filter condition ≤ -1. You can use the modulo option to filter out files that are not a multiple of the sector size, when looking for raw disk images or TrueCrypt/VeraCrypt container files etc.
Created	The date and time the file or directory was created on the volume it resides on. Not available in most Linux filesystems.
Modified	The date and time the file or directory was last modified. On FAT, time precision is 2-second intervals only. On CDFS, the only available date and time stamp is listed in this column although it does not necessarily indicate last modification. Filter available.
Accessed	The date and time the file or directory was last read or otherwise accessed. NTFS last access timestamps are displayed in gray if identical to the creation timestamp, as that on most systems likely means that these timestamps are simply not maintained, for performance reasons, and thus not very significant. On FAT, only the date is recorded. Filter available.
Record changed	The date and time the file's or directory's FILE record (on NTFS) or inode (Linux filesystems) was last modified. These are filesystem data structures that contain the

file's meta data. Filter available.

Deleted — The date and time the file or directory was deleted. Available generally on Linux filesystems and possibly on NTFS (after a particular thorough file system data structure search and viewing/previewing the $UsnJrnl:$J file on the volume, if there is any). Not to be confused with so-called deletion timestamps that other forensic tools may show you on NTFS volumes, for files that have not even been deleted from the file system. Filter available.

Content created [INV, FOR] — Creation timestamp that can be extracted from the internally stored metadata in various file types (see context menu command), as put there by the program that created the file. Internal timestamps are usually less volatile and can be more difficult to manipulate than file system level timestamps. They are useful for example for corroboration. If an official creation timestamp is found in the internal metadata, that timestamp will be presented in this column. If not, various other plausible timestamps may be used as a substitute, even a timestamp derived from the filename if necessary. That way around 60% of all JPEG files can be presented with a Content created value. Filter available.

For more about the timestamp columns please see the next chapter.

Attr. — DOS/Windows attributes on FAT/NTFS filesystems, Unix/Linux permissions and filemode on Unix/Linux/Mac filesystems, plus some proprietary symbols that are explained in the legend (forensic license only) and in topic 2.9.
"Partial initialization" means that according to the file system (NTFS or exFAT) the so-called valid data length is smaller than the logical file size, i.e. the data at the end of the file is undefined, similar to file slack has nothing to do with the file, and was stored on the disk at that location before. You can see the valid data length of the file in File mode in the Info Pane, and the undefined area is highlighted in a different color.
Some GUID partition table partition attributes are shown in the Attr. column: system (=required by operating system), hidden (=not mounted as drive letter), read-only, shadow copy.
When sorting by the Attr. column, files with "more interesting" attributes are listed first, e.g. attributes that indicate encryption, and files without any attributes set or whose attributes are unknown are listed last.
Filter available.

1st sector [not INV] — The number of the sector that contains the beginning file the file's or directory's data. Sorting by 1st sectors means to sort by physical location on the disk and will show files next to each other, that are physically stored near to each other. This column is specially populated for files in Zip archives, with the sector that contains the local zip record of such a file. Clicking a file in a zip archive in Volume/Partition mode automatically jumps directly to its local zip record, which is followed by the (usually) compressed file data. This does not apply to files in nested zip archives. A filter is available, which allows to focus on files whose contents start in certain sector ranges, for example to identify files that are definitely affected by known bad sectors or to identify files whose contents are stored past the end of a known incomplete image. Remember that optionally you

can see physical sector numbers here (disk-based) instead of logical sector numbers (partition-based) if so desired, see Directory Browser Options. With the modulo option you can target files that are either cluster-aligned or not.

In the dialog window with the directory browser options this column can be turned into an "Offset" column, displaying decimal or hexadecimal start offsets of the data of files instead of start sector numbers. This is more precise information and available for most files. The title of the column will be changed accordingly in most places of the user interface. The offset can optionally be made a physical offset (from the point of view of the physical disk/image if shown in a partition) just like the sector number can be made a physical sector number. The filter of that column expects numbers of the same meaning as shown in the directory browser (i.e. either offsets or sectors, either logical or physical), and in the same notation (decimal for sector numbers, decimal or hexadecimal for offsets). The directory browser context menu command "Find duplicates in list" can identify duplicates based on exact identical start offsets instead of just identical start sectors if the "First sector" column is populated with offsets.

FS offset [SPE, LAB, FOR]

Shows the offset of the defining data structure of a file or directory in the file system, i.e. the structure that is the basis for the inclusion of a file in the volume snapshot. That offset is where you can check details manually in case there are any doubts about where X-Ways Forensics got the file system level metadata from. This is also where you may apply a suitable template to get an alternative interpretation and where you can point disadvantaged users of other tools to as they may not be able to find such a crucial location otherwise or don't even get certain deleted files listed. Carved files and files that are embedded in other files for obvious reasons do not have such an offset in the file system (or in the case of carved files at least it is not known to X-Ways Forensics). The file system offset is also where you navigate to when you use the dedicated context menu command to locate a file's FILE record/inode/file entry/catalog key etc., as known from all versions. Clicking the FS offset cell of a file or directory automatically navigates to that offset instead of to the first data sector when in Disk/Partition/Volume mode.

For a partition the FS offset, if available, is the offset where a partition is defined, in a partition table usually.

ID

The identifier assigned to the file or directory by the file system or by WinHex. Not necessarily unique. A filter is available, which makes it more convenient to find other hard links of a given file.

Int. ID

The unique internal identifier of a file or directory in the volume snapshot. Items added to a volume snapshot last have the highest identifiers. Filter available. Useful for example and very easy to use if you would like to focus on the *x* files that were added to the volume snapshot last (after having refined it) or if you would like to resume a logical search with internal ID *y* (filtering out files that may have already been searched before).

For evidence objects that contain a huge number of files, the modulo option allows you to focus on a subset of files that is more or less representative of all files

(though less random than files listed first when sorting by hash value). Applying the modulo operation to the internal ID will pick files from any directory, with any name, creation date etc. To see only 1,000 out of 100,000 files, i.e. every 100th file, use the operation "internal ID modulo 100 = 0". Also useful for testing purposes: If you wish to compare the performance of different hard disks, RAID systems, processors, configurations for volume snapshot refinements, you don't have to process all files in an evidence object. You can get quicker, yet likely representative results for example in 1/10 of the time if you only process every 10th file, pseudo-randomly selected by internal ID.

Even for normal work, examiners may not be required by their bosses/their prosecutor to conduct a 100% complete examination, for example if after review of a reasonably sized and representative subset you can extrapolate that about 10% of several 10,000 photos is illegal material.

Int. parent [not INV]
: The unique internal identifier of the parent directory of a file or directory in the volume snapshot. Useful e.g. when exporting files and directories and there are multiple directories with the same name in the same path (e.g. one existing, one deleted), so that via the internal parent ID you can tell which file resided in which directory even if the path is ambigous.

Unique ID [INV, FOR]
: An internal identifier of a file or directory that is unique within the entire case, not just within the volume snapshot of one evidence object, and unique for the whole life time of the case. The unique ID is easily readable. It contains a delimiter, separating evidence object ID and int. ID.

Unique ID as GUID [INV, FOR]
: Unique ID formatted and extended as a GUID.

Owner [FOR]
: The ID of the owner of the file or directory, on file systems that record that information. On NTFS it's the SID, or, if X-Ways Forensics can resolve it to a username with the help of the SAM registry files already encountered while working with the case, the username. Filter available.

Group [FOR]
: Shows the ID of the assigned group of a file in Linux file systems.

Author [INV, FOR]
: Shows the names of the authors of documents of various types (MS Office, OpenOffice/LibreOffice, RTF, PDF, ...), after metadata extraction. Filter available.

Sender, Recipient [INV, FOR]
: These columns are populated for e-mail messages and attachments extracted by X-Ways Forensics from e-mail archives, plus for original .eml files if metadata has been extracted from them. They come with filters. that allow you to enter any part of an e-mail address or name to search for certain e-mail messages. The filter expression is interpreted as a substring, so no wildcards are needed or supported. You may chose which recipient types you wish to target with the filter: To:, Cc:, or Bcc: or combinations thereof. You can also see To:, Cc:, and Bcc: recipients separately in their own columns, if you prefer.

Link count [FOR]	The hard link count of the file or directory, i.e. how often it is referenced by a directory. A hard link that just provides a short filename (SFN) to satisfy the legacy 8.3 requirements of old Microsoft DOS/Windows versions is not counted as a hard link. Instead, such files get their hard link count marked with a ° in the Links column of the directory browser. That way, the hard link count more accurately reflects the hard links actually present in the volume snapshot of X-Ways Forensics, and normal files always have a count of 1, whereas 2 or more means something more special. If a hard link count of 1 is marked with an asterisk (*), that means that the file or directory is stored as hard-linked in the directory structure in HFS+ although it would not be necessary based on the hard link count. If the hard link count is grayed out, that designates files that will be optionally omitted during a logical search to avoid unnecessary duplicate search efforts and duplicate search hits.
File count [INV, FOR]	The total number of files contained in a directory or in a file with child objects, in the volume snapshot, recursively, i.e. inclusive of further subdirectories. This number can also be found in the name column in parenthesis (depending on the settings).
Hit count [INV, FOR]	The number of search hits that have been found in a file.
Term count [INV, FOR]	The number of search terms (not search hits) that have been found in a file. This takes into account all search terms ever used in simultaneous searches in a case, not for only the search terms that may have been selected in the search term list, unless you have deleted search hits. You can sort by this column to get files listed first that are likely more relevant (because they contain more of the search terms that you were looking for). This column is populated only for evidence objects of a case.
Search terms [INV, FOR]	Lists up to 25 of the search terms found in a file, those that are counted in the preceding column. Useful to get an idea of the search hits in a file even in the normal directory browser, without the need to switch to a search hit list. (forensic license only) Filter available, which is not limited to the 25 search terms displayed in this column.
Page count [INV, FOR]	The page count is extracted from PDF and some Office file types as part of metadata extraction and shown in this column.
Pixels [INV, FOR]	The roughly rounded dimensions of a picture in thousand pixels (KP) or million pixels (MP, megapixels), as the result of width times height, for efficiency reasons stored as a very low precision value. The dimensions are computed simultaneously with skin color percentages, plus when viewing pictures (full-screen mode, preview mode, or in the gallery). Allows to easily distinguish between e.g. small browser cache garbage graphics and high-quality digital photos, with the associated filter, which allows you to focus on pictures with less or equal to the number of pixels that you specify or more or equal or both at the same time. (Works only approximately because of the low precision storage of pixel numbers.) Once at least 1 video still has been exported from a video file, the approximate resolution of the video can

also be seen in this column. (forensic license only)

Analysis [INV, FOR]

Combined column that shows FuzZyDoc matches for textual documents as well as PhotoDNA matches and the computed amount of skin tones in raster images (or the fact that a picture is a black & white or gray-scale picture or too small to contain any relevant graphical content). Available after refining the volume snapshot if the underlying technology is available. Sorting or filtering by this column is the most efficient way to discover traces of e.g. child pornography or search for scanned documents (gray scale or black & white pictures). Sorting by the Analysis column in descending order lists files with FuzZyDoc matches first (those files with the most confident matches for any hash set near the top, with lower percentages following), followed by PhotoDNA matches (showing the category names in an internal PhotoDNA hash database), followed by pictures with no PhotoDNA matches in descending order of their skin tone percentage. After that, irrelevant pictures are listed (picture with very small dimensions), and then files that are not pictures, and near the bottom black & white and gray scale pictures. Text color coding in that column makes it easier to distinguish between different kinds of categorizations. FuzZyDoc matches, PhotoDNA matches and color analysis results are mutually exclusive. That means that if a picture gets it colors analyzed and also a similarity with a PhotoDNA hash value is found, only the PhotoDNA category match is remembered in the Analysis column, not the skin tone percentage, because the PhotoDNA match is considered more helpful. A stylized P is displayed in the Analysis column for pictures for which at least one PhotoDNA hash value is stored in the volume snapshot. If that is the case, the hash value can be seen in Details mode.

Hash [SPE, LAB, FOR]

Up to two hash values can be computed for a file (e.g. MD5 and SHA-1) and then be presented in the two Hash columns. Filters available. The filters allow to focus on files that have a hash value, do not have a hash value, whose hash values start with certain hex values (if you specify only the beginning of a hash value) or have a certain value (if you specify a complete hash value). This filter can compare the hash values of files to up to 4 hash values that the user supplies as hex ASCII. Quicker alternative to creating a small hash set in the hash database if you just wish to quickly find a few files, e.g. duplicates of files with a known hash value that you can just copy from the hash column in the directory browser. The easiest way to use this filter when looking for duplicates of a file, which does not even require copy & paste of hash values, is to right-click a hash value of a given file in the directory browser in hex ASCII notation (not Base32) and invoke the "Filter by" command in the context menu.

The first Hash column displays pseudo-hash values in light gray color until real hash values have been computed [FOR]. Pseudo-hash values are based on the file metadata, not on the file contents. That's why they are available instantly even for very large files. They allow you to list files in a random order just like when you sort by real hash values, but without having to invest time to compute real hash values first. Useful for example for triage, if you have limited time and just wish to quickly look at some randomly selected files in a large evidence object first (e.g. pictures in a gallery) to determine how relevant an evidence object might be.

Looking at files in a *random* order might give you a more complete and accurate impression of what is stored in an evidence object, because the first *x*% of the files listed are more varied and more representative of the evidence object as a whole if they are in a truly random order. If you sort by name or path or size or timestamps on the other hand, many of the files you see will likely be somewhat similar (created by the same application or by the operating system, by the same user, for a similar purpose, created or copied or received around the same time, same file format, ...), so with some bad luck you will only see irrelevant files even if there is an equally large group of relevant files. Remember that if you don't sort in the directory browser at all, the view is skewed as well, because you will see the files in the order in which they are referenced by the volume snapshot, which is more or less the order in which they are referenced by the file system and thus not random.

Sorting by hash values can be combined with any filter, for example to see only pictures larger than 1 MB in a random order or only files of a certain user. Pseudo-hashes are not guaranteed to be unique or even remain the same when you close and re-open the evidence object.

Which hash value out of potentially two hash values stored in the volume snapshot is displayed in the Hash column can be changed in the Directory Browser Options dialog. Either the primary hash value or the secondary hash value or both at the same time (if the box is half checked). The Hash column filter is applied to the hash type(s) that is/are currently displayed. Which hash type(s) is/are displayed in the Hash column can be seen in the column header.

Hash set [INV, FOR]
: The names of the hash sets in the internal hash database in which the file's hash value was found. Up to 64 matches are returned. Filter available. The Hash Set column shows known matches for both internal hash databases simultaneously. The filter can be used to filter for selected hash sets of one of the databases at a time. The database to choose hash sets from can be selected in the filter dialog.

Categorization [INV, FOR]
: Can be set manually with the directory browser context menu, or automatically using X-Tensions, metadata in evidence file containers, matching against hash databases, or using other means. When using hash databases it depends on the category of the hash set in which a file's hash value was found. Initially all files are considered unknown. If known, they are either "irrelevant", "notable", or "uncategorized". Filter available. Note to users with two internal hash databases: The Categorization column shows only one category. If you assign the hash value of a certain file in one hash database to one category and the hash value of the same file in the other hash database to the other category, you will be warned once during matching and given exact information about which hash value in which hash sets in which hash databases are conflicting. The categorization as "notable" will prevail when in doubt.

Labels [INV, FOR]
: The name(s) of the label (s) that were assigned to a file or directory. Filter available. In the dialog window for label management you have the option to prevent certain labels from being displayed in the Labels column (hide them), for example because you don't need to see them and they just clutter up that column or

because you wish to share your screen with someone but don't want them to see those particular labels. You could also decide the certain *kinds* of labels should not be output (e.g. labels that are produced by the picture content analysis). That can be changed in the Notation settings instead. If the parent file of a file has been labeled in some way by the user, then this is pointed out in the "Labels" column for the child object as well, in light gray color and with an arrow, except if the child object has labels itself. Reminds the user that the parent was reviewed and marked as relevant already, which may spare him or her the extra step of navigating to the parent again.

The label filter can target labels in two different ways. It can require the presence of certain labels (selected labels with a check mark, at least one of multiple such labels = OR or all of those labels = AND) and require the absence of other selected labels (those with a red X, and again at least one of multiple such labels = OR or all of those labels = AND). That means for example you can focus on files that have (label A AND label B), but not label C. Or on files that have (label A OR label B), but not (label C AND D). The positive condition and the negative condition (if labels are selected for both) are always ANDed with one another. The buttons Force and Exclude assign labels to the positive and the negative camp, respectively. Whether labels are actually visible in the Label column or deliberated hidden has no effect on the filter.

Comments [INV, FOR]

The free text comment that may have been assigned to the file or directory by the examiner. Filter available. If not used for comments, you can choose to see decoded or OCR-derived text from a file in this column instead (depends on the Notation settings) if such text was stored in the volume snapshot. Such extracted text is displayed in a gray color to set it apart from actual user comments. To see more text, you can move the mouse cursor over the respective cell. The Comment *filter* still works only based on actual comments.

Metadata [INV, FOR]

Internal file metadata can be extracted from files of various types by refining the volume snapshot, and shown in this column. That is a subset of the more extensive metadata presented in Details mode, useful for filtering, export, and report purposes. It can be edited with a command in the directory browser context menu. Please note that the frequently occurring word "Generator signature" that can be seen in the Metadata column is not stored literally internally and thus cannot be found by a logical search in directory browser cells or with the filter.

There is an option to apply the Metadata column's filter to the more extensive metadata that are presented in Details mode, not the subset stored in the Metadata column. This is slower because the extensive metadata are not kept in the volume snapshot, but extracted from each file each time on demand (when needed). Thus the recommendation is to use this filter setting only in conjunction with other filters, e.g. the Type filter, if applied to very long file listings. Anyway if such an operation takes too long, you can abort it in the usual way. For example you can find files that way in whose extended metadata X-Ways Forensics lists a particular annotation like #250.

Generator

The generator signature, which is known from the Metadata column, is additionally

signature [FOR]	presented in its own separate column, for sorting purposes, which may allow to identify logical connections between files.
Device type [INV, FOR]	This column shows for about half of all JPEG, PDF, video, PNG and WEBP files by which class of input device they were produced or (out of necessity, e.g. after removal of Exif metadata) for what class of output device they were edited, if the confidence in that assessment exceeds a threshold that you can define in the metadata extraction settings. The results improve if the picture content analysis is also applied. A generating device type can be a scanner, DSLR (single-lens reflex), mirrorless camera, digital back, video camera (camcorder), point and shoot (compact) camera, smartphone, smartphone front (secondary) camera, mobile phone, webcam/IPCam, action cam, monitor camera, or tablet. "Printer" for a JPEG file means that the picture was meant to be printed. Screen could indicate a screenshot taken off a screen, or a wallpaper or picture that was meant to be displayed on a screen with tailored pixel dimenstions. Certain pictograms, AI-generated pictures and graphical elements are presented with the device type "no device" to show that they were not generated by any image capturing device. The device type is mostly derived from the generator signature. Filtering for the device type could be useful for example if you are looking for rather private photos like selfies taken with a smartphone's front camera or rather professional photos (e.g. DSLR or digital camera back).
Structure type [INV, FOR]	This column can be populated as part of metadata extraction. Filter available.
Relevance [INV, FOR]	Generic relevance of a file. Can be computed when extracting metadata. Filter available. This relevance is based on a variety of factors, such as the type of the file, its generator if known (for JPEG and PDF files), its currentness (last modification date), whether it is known from any hash database, the wealth of internal metadata that it contains, its size, the visual content of pictures, whether a PNG file is a smartphone screenshot, whether an HTML file has been locally saved by the user manually, whether there is something unusual about the file, etc. etc. The relevance is not merely content-based, but the result of a fundamental characterization. In particular the generator signature is a provenance-based criterion. The main idea is that if your time for examination is limited, you can start with the files that have the highest generic relevance, to maximize your chance to find what you are looking for, if it exists, and find it rather early. To sort listed files by relevance in descending order, i.e. prioritize them for review, you can also use Navigation | Sort by Relevance from the directory browser context menu.

Metadata, Comments, and Event Description filters support the use of up to 4 expressions, which can be flexibly combined with AND and OR. The last combination always has priority. For example "A and B or C" is interpreted as "A and (B or C)". "A or B and C" is interpreted as "A or (B and C)". The expressions may start with a colon to indicate NOT at the expression level.

Additional columns for search hit lists [INV, FOR]: Physical/absolute offset, logical/relative offset, description on the nature of the search hit (code page/Unicode, whether in decoded text,

whether in file slack), search hit with context preview. If the logical relative offset is printed in gray, that means the search hit was found in the decoded text and the offset is not an offset in the file, but in the decoded text.

Additional columns for event lists [INV, FOR]: Timestamp, event type, event type category, description.

Some more tips: Right-clicking a column header in the directory browser quickly activates or deactivates that column's filter without showing the settings dialog window. There is a symbol on the right-hand side of the caption line of the directory browser, a lower-case i in a circle, that you can left-click to get a textual summary of all active filters with their settings, if defined in dialog windows. You can sort by a column not only by clicking the header, but also through the dialog window that appears when pressing Ctrl+H. That window also allows you to activate and deactivate filters purely using the keyboard.

3.3.6 More about the Timestamp Columns

Timestamp columns designated with a superscript 2 contain alternative timestamps [SPE, LAB, INV, FOR]. In the case of NTFS these values are taken from 0x30 attributes and represent previously valid timestamps from when a file was last renamed or moved, or possibly before some backdating operation occurred. Backdating operations are often applied by setup programs and also Windows itself (the infamous creation timestamp tunnelling effect, cf. http://support.microsoft.com/kb/172190), and of course potentially by ordinary application programs as well as by users for various legitimate or less noble purposes. Note that these columns by default (but depending on the Notation settings) are populated only if these previously valid timestamps are actually different from their current counterparts, and additionally Modified² and Record changed² only if different from Created², to avoid cluttering the screen unnecessarily with redundant information. That means any ² timestamps that you see there by actually contain additional information and are not redundant.

Created² is also populated for HFS+ file systems, with the relatively new "Added date" timestamp from Mac OS X Lion and later as well as iOS, where available and if different from the regular Created date. That timestamp specifies when a file was added to the particular directory in which it is contained, even if originally created earlier.

For e-mail: The timestamp in the "Date:" line in the header of an e-mail message extracted from an e-mail archive (if accompanied by a time zone indicator like -0700 or +0200) is shown as the creation date & time of the e-mail message as well as all its attachments. The timestamp in the "Delivery-Date:" line (or alternatively, if not available, the first "Received:" line) is listed as the last modification date & time. E-mail attachments show the same timestamps in the Creation and Modification columns as the e-mail messages to which they belong, so that you can see directly when these attachments were sent and delivered without having to navigate to the parent e-mail message. The record changed timestamp tells you when data in the data structure about the e-mail message in the OST file has changed, for example for a sent e-mail message when the user clicked the Send button or for a received e-mail message when it was delivered to the e-mail client or generally when a message was copied to another PST/OST archive.

The combined filter for all the timestamp columns allows to filter for certain date ranges (typical application) or for mere times, matching any possible date. For example if you are interested in unusual activity occurring in the middle of the night when the rightful office computer user is not working, you could filter for times such as between 22:00:00 and 05:59:59 (on a 24-hour clock). Obviously, selecting the right local time zone for the timestamp filter is crucial for this.

There are two different types of AND combinations for timestamp filters. A strict AND combination (fully checked) requires that all targeted timestamps are actually present/available. A soft AND combination (half checked) requires only all available timestamps to meet the filter condition (and at least one must be available).

The timestamp filter allows to focus on NTFS 0x10 timestamps that look like significantly backdated compared to their their 0x30 counterparts, and also on creation timestamps from the file system that are earlier than content creation timestamps from the internal file metadata. You can define a threshold in milliseconds, seconds, minutes, hours, days, weeks, months or years that makes such timestamp discrepancies relevant to you, i.e. you can require that the main creation timestamp is that much older than the corresponding 0x30 timestamp or the content creation timestamp. If you compare UTC-based creation timestamps from the file system to content created timestamps that were recorded in an unknown local time zone (e.g. in PNG files), you could take into account how many hours difference could be attributed merely to this base effect. To assist you, the UTC creation timestamp is rendered more comparable to a local content creation timestamp by adjusting it to display time zone. Please note that backdating most often occurs automatically for various reasons (e.g. file archive extraction or setup programs) and is not necessarily the result of intervention by a suspect or by malware activity with malicious intent. If you are interested in potential manual interventions by a suspect, it could be useful to employ the file type filter at the same time and focus on file types for which timestamps could make a difference in your case e.g. documents.

NTFS 0x10 timestamp cells show a backdating icon if they predate their corresponding 0x30 counterparts (the columns with the superscript 2 in the header) by more than the threshold that is defined in the timestamp filter dialog window, only if the 0x30 timestamp is not considered redundant (see above). Creation timestamp cells show such an icon also if they predate the "Content created" timestamp, if any such timestamp exists and was extracted. The arithmetic difference between the two timestamps that are compared is displayed after the icon, rounded to milliseconds, seconds, minutes, hours, days, weeks, months or years. Note that the situation where a modification timestamp is earlier than a creation timestamp is not considered backdating, but merely the result of copying a file, where the copy is assigned a new creation timestamp. Such files can be identified as "copied" in the Description column depending on the Notation settings.

Please note that for FAT volumes, all timestamps are displayed by default as they are stored, in local time, i.e. unadjusted, unless in the properties of an evidence object with a FAT file system you declare which time zone you suppose was the original time zone, in which case timestamps are converted to the display time zone when they are displayed. For all other file systems the time zone concept applies.

Timestamps in the normal directory browser that meet the timestamp filter condition are

highlighted. Timestamps in an event list that are identical to the event timestamp are also highlighted.

Underflows and overflows in the timestamp columns (timestamps outside of the supported range) are marked with the text "out of bounds", and they can be distinguished from each and properly sorted and filtered. The supported range is May 5, 1829 through May 14, 2514.

When sorting timestamps in one of the many timestamp columns, it may happen that UTC-based time stamps have to be compared to local timestamps with an undefined time zone reference or local timstamps with a user-defined time zone reference (user-defined meaning defined by the examiner), to see which one is earlier and which one is later. That happens for example for file system based timestamps in the case root window if one evidence object has an NTFS file system and the other a FAT file system. It also happens within the same evidence object for example when sorting internal creation timestamps retrieved from file contents, such as ordinary Exif timestamps in JPEG (which are local) and GPS timestamps in JPEG (which are stored in UTC). Sorting all such timestamps takes into account how these timestamps are displayed (in original local time or in a user-defined display time zone) such that the order is consistent with the displayed values, and not with how the timestamps are internally stored. That means for example that the local Exif timestamp 2017-01-01 14:01 LT is sorted *after* a UTC GPS timestamp 2017-01-01 14:00 +2, which is right if the undefined local time zone is equal to the display time zone, which in this example is UTC +2. That order of course can be wrong, as the unknown time zone of a local Content created timestamp could be somewhere to east of UTC +2. The order could also be wrong if the user-defined time zone reference of timestamps from a FAT file system is wrong.

3.3.7 FlexFilters

Two so-called FlexFilters are available in WinHex Lab Edition, X-Ways Investigator and X-Ways Forensics, in the dialog window with the options of the directory browser. They can target any column in the ordinary directory browser (i.e. not search hit list or event list specific columns) that the user wishes to focus on, with an arbitrary number of substrings, and they can be combined with a logical OR or a logical AND.

For example, these new filters are useful if you wish to target files that were created or modified not in a particular contiguous period of time, but generally on certain weekdays or on weekends, i.e. where either of these columns contain the word "Saturday" or "Sunday" in the long date notation format. Also useful whenever the column-specific column filter does not give you as many options as you need (e.g. for Author, Sender, Recipients currently you can only enter one name or address or substring, and with the Description filter you cannot currently specifically target additional hard links that are optionally omitted from certain operations).

The color that indicates that a FlexFilter is active is violet instead of blue, so that it can be better distinguished from a regular column filter. Both FlexFilters come with a NOT option, and they may also target the same column, so that you can achieve results like "show all e-mail messages sent with the name John Doe in the sender field where the sender field does NOT contain the domain name company.com".

3.4 Mode Buttons

If a data window represents a storage device, a drive letter/partition/volume, or an interpreted image of the same, there are several buttons that determine what is shown in the lower half of the data window, below the directory browser. Forensic licenses only.

Disk/Partition/Volume/Container

This is the default mode. It shows the binary data in all sectors of the disk/partition/volume/container represented by the active data window as hexadecimal code, as text, or both. Offsets and sector numbers are relative to the start of the respective disk/partition/volume/container. If no sector-wise access to the data source exists, for example because the data window merely represents a directory listing provided by the operating system, then this mode remains practically useless.

File

Looks similar to Disk/Partition/Volume/Container mode, but shows only the clusters allocated to the file or directory that is currently selected in the directory browser, in the order as used by the file, defragmented if fragmented, decompressed if compressed, with offsets relative to the beginning of the file. When switching from File mode to Partition/Volume mode, X-Ways Forensics will automatically point you to the offset from the point of view of the partition/volume that is equivalent to the offset within the file where the cursor was positioned last, even if the file is fragmented, if there is an equivalent position (not if the file is a compressed or virtual attached file or an extracted e-mail message or an exported video still etc.).

Raw submode is available for NTFS-compressed and WofCompressed files in File mode to see the complete compressed data with slack. (The List Clusters command lists all clusters of such files including the slack as well. The slack area of the WofCompressed data is highlighted also in Partition/Volume mode.)

Preview

Checks the type of the file currently selected in the directory browser and displays the file with the help of the separate viewer component, except if the viewer component is not active or if it's a picture (supported file types see Gallery below) and the viewer component should not be used for pictures. Even incomplete pictures (e.g. files incompletely recovered because of fragmention) can usually be displayed partially. If the viewer component is not active and the file is not a picture in one of the supported formats, a rudimentary ASCII text extract from the beginning of the file is displayed.

If pictures in Preview mode are shown by the internal graphics viewing library, not the separate viewer component, they can now be rotated in 90° steps by clicking the left mouse button (to rotate to the left) and the right mouse button (to rotate to the right). Photos taken by mobile phones and digital cameras of certain major manufacturers in portrait mode are stored in landscape orientation and marked as to be rotated left or right. Preview mode with the internal graphics viewing library adjusts those photos to the correct orientation automatically. Clicking

the middle mouse button in Preview mode when a picture is shown by the internal graphics viewing library will mirror the picture (flip horizontally) or if the Shift key is pressed flip the picture vertically. Please note that this operation is applied in addition to any active rotation. The currently active rotation and flip mode are described by some symbols in the upper right corner. If no flipping has taken place, but a rotation, the letters "BR" indicate what in the original graphical data was the bottom right corner. You have the ability to zoom in and out when pictures are rendered by the internal graphics display library using the mouse wheel, in steps of 10%. (This does not change the center of the picture based on the mouse pointer position. If you wish to navigate within a greatly magnified picture, please use the View command for that.)

You have the option to see the presentations of Preview and Details mode for the same file at the same time, side by side, after clicking the "+" on the Details button when in Preview mode. Clicking the Details or Preview button again will make that mode the only active mode.

Details

Contains all the information on a single selected file from all the directory browser columns, including those that are not currently visible. Very useful for example if the path is very long and does not fit on the screen in the path column, maybe not even in the path tooltip display. The approximate scroll position in Details mode is restored selecting a different file in the directory browser or when closing and re-opening the data window or the application. Clicking the floppy disk icon in the status bar allows to save the contents of Details mode into an HTML file.

The Details mode also shows NTFS file permissions (stored in access control lists, ACLs). Each element has typically the property "Grant" or "Deny" and an SID to which the permission applies. The SID is translated into a friendly name if possible. The permission itself is either R = Read Permission, C = Change Permission, Full Control or Special Access. For a Special Access right, all individual rights are listed. For each permission there can be two inheritance flags: container inherit (CI), object inherit (OI) or two propagation flags: inherit only (IO), no-propagate inherit (NP). Usually the final list element is the group membership property.

The Details mode also extracts some essential internal metadata from OLE2 compound files (e.g. pre-2007 MS Office documents), MS Office 2007 XML, OpenOffice XML, StarOffice XML, HTML, MS Access, MDI, PDF, RTF, WRI, AOL PFC, ASF, WMV, WMA, MOV, MP4, 3GP, M4V, M4A, JPEG, BMP, EXE/DLL, JIDX (Java applet cache), THM, TIFF, GIF, PNG, GZ, ZIP, PF, IE cookies, DMP memory dumps, hiberfil.sys, PNF, SHD & SPL printer spool, RecentFilecache.bcf, WIM Vista image files, PhotoShop PSD, INDD (Adobe InDesign), DocumentSummary alternate data streams, tracking.log, .mdb MS Access database, manifest.mbdx/mbdb iPhone backup, IconCache.db, and many more. For MS Office documents, you will often see many more timestamps (e.g. Last Printed), subject, author, organization, keywords, total edit time, and much more. A lot of information is also gathered from JPEG files, see chapter “JPEG Details”.

In "IM" sub-mode, which can be activated by pressing the "IM" button, you will see only the internal metadata of a files. That makes it more efficient to check multiple files for that kind of metadata without having to scroll. In particular this is useful for forensic review of photos, to check the Exif data. Depending on some factors you may have the ability to toggle between single and double column IM mode when viewing internal JPEG metadata. Given a sufficient

screen resolution and window width, no scrolling is required to quickly review the entire internal metadata in double column mode, as the summary table is on the right-hand side. “IM” shows internal metadata as well as certain metadata from the file system, where available: NTFS permissions and NTFS object ID deconstruction.

Gallery

Checks the file signature of all the files in the currently visible portion of the directory browser if that was not done for these files in the volume snapshot before. If a file is identified as a supported picture file type, a thumbnail is displayed, otherwise you see a white tile with the filename or optionally thumbnails can even be generated for such non-picture files (see Options | Viewer Programs). By scrolling in the directory browser, the gallery view scrolls as well. The gallery reflects the selection in the directory browser. You may switch the directory even while the thumbnails are still loading. By double-clicking a thumbnail, you get a full-size view of a picture, where you may zoom in and out using the keys + and -. Even incomplete pictures (e.g. file incompletely recovered because of fragmention) can usually be displayed partially. Supported picture file types: JPEG, PNG, GIF, TIFF, BMP, WEBP (only the first frame if animated), HEIC, AVIF, some DICOM variants, PSD, HDR, PSP, SGI, PCX, CUT, PNM/PBM/PGM/PPM, ICO. The gallery does not go together very well with search hit lists.

The gallery can operate in an alternative mode, activated with the button left to the Sync button. In that mode the gallery does not present the items currently listed in the directory browser, but instead all the child objects of a single selected item, if there are any such child objects. Those are either only direct child objects or (in ² mode) child objects recursively. This is a unique way to get a quick overview of entire directories or file archives with a single mouse click. Also very useful for videos from which stills have been extracted. The gallery can have its own selection separate from the directory browser in that mode. You can right-click any listed child object in the gallery and perform various operations on that particular object. Most commands known from the directory browser context menu are available. In particular you can provide a child object with labels that way, exclude it, tag it, or navigate to see it in its native parent directory in the directory browser with all metadata (and then you can click the Back button to return to the previous view). The child objects are listed in the gallery in ascending order of internal ID. The selection in the gallery usually exactly replicates the selection in the directory browser. However, when representing child objects of a file that is selected in the directory browser, the gallery now allows a separate selection in itself, among the child objects.

When a View window displays a picture, if limited to one such window, that window will be updated with the next picture when you hit the cursor keys in the gallery. Useful especially if the View window is centered on the second monitor if the gallery is on the first monitor, on a spanned desktop. Avoids having to press the Enter key to view the picture and another key to close the View window to get the input focus back to the gallery.

Calendar

Gives a convenient visual overview of the timestamps of all listed files/directories, from all 6 timestamp columns of the directory browser, in the form of a calendar, or when in event list mode a similar overview of all listed event timestamps. Each day with at least one time stamp is marked in the calendar with a gray color. The more activity on a day, the darker the color.

Weekends (Saturdays and Sundays) are specially marked with x. Hover the mouse over a day to find out how many timestamps exactly fall into that day. Left-click a day to select that day as the left boundary of the timestamp filter, or right-click it to define it as a right boundary. Middle-click a day to filter for timestamps on that particular day only. If the same file is listed more than once (which can happen in a search hit list if it contains more than 1 search hit), then its timestamps are also represented more than once in the calendar.

When not showing events, you can now decide which column's timestamp should be included in the calendar. Columns that are hidden (have a width of 0 pixels) are excluded, all other columns are included. The status bar reminds you which columns are included even if not currently visible because of horizontal scrolling.

Years in the calendar with no timestamps are grayed out. The number of a year is displayed in a darker shade of gray the more timestamps are listed for that. All shades of gray try to give the examiner a better and quicker impression of peaks or absence of activity.

As the number of years represented in Calendar mode is limited, garbage timestamps in the far past can keep you from seeing the later years that you are interested in if you don't set a filter or don't delete events with garbage timetamps. You can specify the minimum year that will be represented by the calendar. Any timestamps in earlier years will be disregarded by the calendar even if no filter is active. By default, the minimum year is the year 2000. To change it, click the number of the first year on the left in Calendar mode.

Example: During which period of time were most JPEG files processed on a volume? Right-click the root directory in the directory tree (case data window) to recursively list all files from all subdirectories, then use the file type filter to limit the view to JPEG files, enable the calendar view.

Raw

In Preview mode, in conjunction with the viewer component, when viewing non-picture files, Raw mode renders the file as plain text. This can be useful for example for HTML files to see the HTML source code, for .eml files to the see complete e-mail header, and generally when in search hit list mode the viewer component cannot highlight a search hit in Preview mode (because then it might contained in metadata or control code that would be represented in raw Preview mode, but not normal Preview mode). You can make Raw preview mode persistent by holding the Shift key when activating Raw mode.

File mode now offers a "raw" submode for NTFS-compressed files. In Raw mode you can actually see the compressed data as well as the sparse clusters, not the decompressed state of the file. This is useful for research or educational purposes and because theoretically small amounts of data could have been manually hidden in the not clearly defined, but implicitly existing slack area of each compression unit, which follows the compressed payload data.

VC

The VC button is visible only in Preview mode when viewing pictures of types supported by the internal graphics viewing library. By default the internal graphics viewing library is used to

preview or view pictures. However, if the "VC" button is pushed, the viewer component is used instead, which is also responsible for displaying the thumbnails in the gallery.

Sync

Synchronizes the directory browser and the directory tree in that when in a recursive view you select a file in the directory browser, its parent directory will be highlighted. Sync mode in non-recursive exploration mode has a similar effect as the option "Automatically expand to current folder" in the Windows Explorer. That means that when navigating from one directory to another using the directory browser while Sync mode is off, the directory tree on the left will not reflect the current directory any more, will neither expand its parent if necessary nor select the current directory. Whether Sync mode is active or not is remembered separately for recursive and non-recursive exploration and separately for each data window.

The Sync button in the Case Root window has a special function. That is possible because directory navigation is not available in that window. The button controls whether switching focus from one data window to another, for example using the tab control, should highlight the corresponding evidence object or its currently explored directory in the Case Data window.

Exploration Mode

Button with a curly turquoise arrow. Toggles between normal and recursive exploration of a directory. When exploring recursively, you do not only see the contents of the current directory, but also the contents of all its subdirectories and their subdirectories, and so forth. To explore a directory recursively, you may also right-click it in the directory tree.

Multi-monitor support

It is possible to detach the lower half of a data window (with Disk/Partition/Volume mode, File mode, Preview, Gallery etc.) from the data window, by clicking the three dots that are located left to the mode buttons, twice. After that, you can freely move and resize it on the screen. On multi-monitor this allows you to have that part of the user interface on a separate screen and even maximize it there. Reintegrating it into the main window is done by clicking the same three dots again or by clicking the Minimize button.

Clicking the three dots for the first time will show that part of a data window on the right-hand side of the directory browser instead of below it. That can be useful on today's widescreen monitors, where vertical screen space is scarce, so that you can have a long vertical list of files visible and at the same time also fully utilize the available vertical screen space for example for previews of page-based documents that were meant to be viewed in portrait mode as opposed to landscape orientation. Also useful for the gallery, and very efficient for portrait mode photos, Details mode, and hex editor displays in Disk/Partition/Volume and File mode with traditionally just 16 bytes per line.

Right-clicking anywhere in the Mode button bar outside of all the buttons will show or hide a divider line between the directory browser and the lower half of a data window, above the Mode button bar. The mouse cursor shape changes when hovering over the divider line to indicate that you can move it up or down. If the divider line is invisible, you can adjust the window height by

left-clicking in the Mode button bar outside of all the buttons and moving the mouse cursor up and down while holding the mouse button. Without the divider it is more intuitive that the right-hand side of the Mode button bar is related to the directory browser and also acts as its status bar.

3.5 Status Bar

The status bar displays the following information when the hex editor display in on the screen:

1. Number of the current page and total page count, or sector number and sector count in the case of storage devices or disk images (and sectors are numbered starting with zero by convention)
2. Current position (offset)
3. BEN or LEN to identify the nibble order in the hex display
4. Interpretation of the hex values at the current position, prepended with “BE” if in big-endian mode
5. Beginning and end of the current block (if currently defined)
6. Size of current block in bytes (ditto)

Click the status bar cells in order to...
1. Move to another page/sector,
2. Move to another offset,
3. Toggle the nibble order,
4. Define the integer type for decimal translation and
5./6. Define the block.

Right-click the status bar in order to copy pieces of information from the status bar into the clipboard. Right-clicking the 2nd status bar cell permits switching between absolute (default) and relative offset presentation.

3.6 Data Interpreter

The Data Interpreter is a small window that offers possible translations for the data at the current cursor position. Whether it is shown or not can be controlled via the View menu, not with the options of the data interpreter. Contrary to popular belief among some users, it totally disregards any block if selected and always interprets from the byte where the cursor is. The options dialog lets you specify the data types to interpret. These are various integer data types (by default in decimal notation, optionally hexadecimal or octal), the binary format (8, 16 or 32 bits of a byte), four floating-point data types, assembler opcodes (Intel®), and date types.

Little and big endian variable-length representations of unsigned integer numbers can be decoded. In little endian this is compatible with (U)LEB128 (=Little Endian Base 128). In big endian this is compatible with SQLite3's varint. In templates the variable type is called "varint".

The *bits* in binary representation in the Data Interpreter, in templates as well as in the status bar can optionally be shown in ascending order, i.e. from the lowest-valued / least significant bit (LSB) to the highest-valued / most significant bit (MSB). This is not to be confused with and

works in addition to the *byte* order, which defines which of multiple bytes is the highest valued one (depends on the Big Endian switch). If the LSB-to-MSB order is active, a triangle with its smaller end of the left and its bigger end on the right is displayed before the word "Binary:" to remind you.

The Data Interpreter can interpret UNIX/C, Java/BlackBerry/Android, FILETIME, HFS/HFS+, APFS, and Mac Absolute timestamps stored as integer numbers in decimal ASCII text instead of in binary. You will find a context menu item for that as well as a checkbox in the options dialog. The Data Interpreter optionally translates timestamps of all formats except MS-DOS date & time to local time (the time zone defined in the General Options). You will find a context menu item for that as well as a checkbox in the option dialog.

The Data Interpreter is also capable of translating most data types back into hex values. Make sure a file is open in an edit mode other than read-only mode, enter a new value in the Data Interpreter, and press **ENTER**. The Data Interpreter will then enter the corresponding hex values into the edit window at the current cursor position.

Right-click the data interpreter to bring up a context menu. This will let you switch between big-endian and little-endian translation of integer and floating-point data. You may also choose between decimal, octal, or hexadecimal integer representation. See the Data Interpreter Options for more settings.

The decomposition of V1 GUIDs into timestamp, sequence number and MAC address in the Data Interpreter as well as in templates is optional. In the Data Interpreter options you can now choose to force the decomposition (fully checked) or prevent it (to always get the standard GUID notation with braces) or to see the decomposition only if the timestamp is not too implausible (half checked). The latter setting is helpful for example for Apple GPT values that claim to be V1 GUIDs, but contain twisted ASCII text instead of valid timestamps.

Hints:

- Some hex values cannot be translated into floating-point numbers. For these hex values the Data Interpreter displays NAN (**n**ot **a n**umber).
- Some hex values cannot be translated into valid dates. The value ranges of different date types are more or less narrow.
- There are redundancies in the Intel® instruction set, which show up in the Data Interpreter as duplication of both hex opcodes and mnemonics. Floating-point instructions are generally displayed as F***. More detailed reference can be found in the Intel® Architecture Software Developer's Manual Volume 2: Instruction Set Reference, available in PDF format on the Internet.

3.7 Position Manager

The Position Manager maintains a list of file or disk offsets and corresponding descriptions, called positions, which can serve as annotations. It is also used for search hits when not working with a case, but *much* less powerful than a search hit list. Navigating from one entry to the next is

easy if you press Ctrl+Left and Ctrl+Right. You may enter new positions and edit or delete existing entries. If a special offset in a file is important to you, you can add it to the Position Manager. This makes it a lot easier to find it again later, and you do not have to remember it. Descriptions may be up to 8192 characters in size. An appropriate description for instance could be “Data chunk begins here!”. Optionally all positions maintained by the Position Manager can be *highlighted* in the editor window in a unique color you specify, and their descriptions displayed in yellow tooltip windows when the mouse cursor is moved over them. You may also add or edit positions with the context menu of an edit window or by clicking the middle mouse button in an edit window.

Click the right mouse button in order to see a context menu in the Position Manager. The context menu provides additional commands. You may delete, load or save positions, even export the list as HTML. If the position list in the *general* Position Manager was changed, it is saved in the file *WinHex.pos* when exiting WinHex, so that they are still available in the next session. Only search hits are not permanently saved, unless they have been edited via the context menu.

There is the *general* Position Manager, which stores positions that are applied to *all* data windows, and there is the a Position Manager for each evidence object in a case, which stores positions that were defined for that particular evidence object and that are applied only to that evidence object's data window. The former is invoked through the main menu (Navigation | Position Manager), the latter by clicking the right-most button in the middle of the screen when an evidence object is open, with crosshairs on it. That may explain it if you cannot find the positions that you have defined previously. Near the top of the data window it says *which* Position Manager you are currently looking at if the Position Manager is active.

Search hits in the general Position Manager are by default deleted as soon as the general Position Manager is closed, to avoid confusion as positions in the general Position Manager have no reference to a particular file or disk and are intentionally applied to whatever data source is active when invoked. If you wish to keep search hits, please change the corresponding option in the general Position Manager's context menu.

The complete documentation of the POS file format is available from the WinHex homepage at http://www.x-ways.net/winhex/.

3.8 Useful Hints

- Menu commands that affect individual, selected items in the directory browser or in a search hit or in the Position Manager can be found in the context menu that opens when you right-click such items. You won't find such commands in the main menu.
- Use the mouse buttons as follows to define the block (if the context menu is switched off):
  - Double-clicking left sets the block beginning.
  - Single-clicking right sets the block end.
  - Double-clicking the right button clears the block.
- You may want to define the block using the keyboard (**SHIFT**+arrow keys or **ALT**+**1** and **ALT**+**2**).
- Use the **TAB** key to switch between hexadecimal and text mode.

- Use the **INS** key to switch between insert and overwrite mode.
- **ENTER** displays the Start Center.
- **ESC** aborts the current operation if any, otherwise clears the block, dismisses an active dialog or template window.
- **PAUSE** stops or continues the current operation.
- **CTRL**+**S** saves the current case if a case is active and not opened as write-protected.
- **F11** repeats the last Go To Offset command. **CTRL**+**F11** works in the opposite direction (from the current position).
- **SHIFT**+**F7** switches between three character sets.
- (**SHIFT**+)**ALT**+**F11** repeats the last Move Block command.
- **CTRL**+**SHIFT**+**M** invokes an open evidence object's annotations.
- **ALT**+**LEFT** and **ALT**+**RIGHT** allow for switching between records within a template (just as the "<" and ">" buttons). **ALT**+**HOME** and **ALT**+**END** access the first and the last record, respectively.
- **ALT**+**G** moves the cursor in the edit window to the current template position and closes the template window.
- **CTRL**+**F9** opens the Access button menu (disk edit windows only).
- Pressing **CTRL**+**C** in the directory browser copies the textual data of the selected items into the clipboard, with the same notation as in the directory browser itself, otherwise using the functionality of the Export List command.
- Use scripts to make your work with WinHex more efficient.
- WinHex is drag-&-drop capable. However, Windows prevents drag & drop if the receiving application was run as administrator and the sending application was not.
- "Invalid input": When clicking OK in a dialog box and getting the "Invalid input" error, pay attention to what control item in the dialog box is blinking, as the value in that item is the one that is not accepted.
- Switch from hexadecimal to decimal offset presentation by clicking the offset numbers.
- Try clicking the status bar cells (left and right mouse button).

3.9 Special Dialog Window Functionality

All edit boxes throughout the program (except for password edit boxes and column width boxes) remember a history of up to 10 last entries. The history can be seen when clicking the tiny button that appears in an edit box for which a history is available. Alternatively, you can press the F4 key just like in a normal drop-down box (combo box). If you select a previous entry from the pop-up menu, it will be inserted into the edit box automatically. Users who wish to delete these histories or pass them on to others, please be advised that they are stored in the file History.dat when the program is ended. That file can be simply copied or deleted. If you do not wish to keep histories between sessions, you can create an empty file named History.dat yourself and render it read-only. To delete a specific history entry for a particular edit box, selecting that entry from the pop-up menu with the Shift key pressed.

Since the days of Windows 95 (or perhaps even Windows 3.1?) users can press Ctrl+C to produce a plain-text representation of standard Windows message boxes in the clipboard. With message boxes in WinHex and X-Ways Forensics it works the same. Although this is an

elementary feature in Windows for 30 years already and should be known to any experienced Windows user and although WinHex and X-Ways Forensics make users aware of that ("Did you know? ..."), the great majority of users for some reason still take graphical screenshots of message boxes and paste them into HTML e-mails, for example when they report error messages, although that is more work than simply pressing Ctrl+C and Ctrl+V and although it inflates the size of the e-mail unnecessarily, as a few ASCII characters need much less space them thousands of pixel values. That also means the screenshot will get lost if the e-mail is converted to plain text when being replied on, and of course the error message text will not be searchable in a graphical screenshot and cannot be conveniently selected and copied to the clipboard as text by the recipient, and the recipient cannot be sure of the exact Unicode value of certain characters for which multiple variants exist.

In WinHex and X-Ways Forensics it is even possible to copy a rudimentary ASCII representation of dialog boxes and almost all their control items (static text, push buttons, check boxes, radio buttons, list boxes, combo boxes, and tree view controls) including their states (unchecked, checked, half checked) by pressing Ctrl+C with an active dialog box on the screen (not if an edit box with a selection has the input focus). There is also a dedicated command in the window menu of an dialog box. That menu is a.k.a. the system menu or control menu, and it pops up when right-clicking the title of a dialog box. This copy command is a very efficient way to show your settings in a certain dialog box to other users and let them copy strings for use in their own edit boxes, so that they don't have to type them, avoiding typos. The text representation is even more powerful than a screenshot because it shows the contents of edit boxes and list boxes completely, even if these controls have scrollbars and the contents exceed the physical boundaries of the controls on the screen. Unicode characters are supported. We suggest that users take screenshots of message boxes and dialog boxes only if absolutely necessary, for example if they wish to graphically highlight certain control items in a Photoshop or similar programs to get the message across.

Settings in practically all dialog boxes can also be conveniently saved to and loaded from files as needed, for example to share them with other users or for future use, via the system menu. This function can remember the selection states of the most important control types: check boxes, radio buttons, list boxes, combo boxes, and tree view controls. This works even if the controls are currently invisible. The settings are stored in files with the .dlg extension (for "dialog"), in the same directory as templates and scripts. The contents of edit boxes are also remembered. However, this function does not remember the contents/text labels of check boxes, list boxes, combo boxes, and tree view controls, e.g. which code page a check box represents in the Simultaneous Search dialog, which label names exist in the label filter list box, which external programs are listed in the Viewer Programs dialog window, which file types are listed in a tree view control etc. It also does not remember the order of controls or list items. It also does not remember settings in a dependent dialog window (which opens e.g. when clicking a "..." button).

You can *hide* controls in dialog windows before saving your settings in a .dlg file so that the values of those controls remain undefined in that .dlg file and cannot cause problems next time when you wish to use that .dlg file, perhaps without supervision through the command line. To hide a control, you hold the Shift key and roll the mouse wheel (in either direction) over that control. It is useful to prevent control values from getting saved in a .dlg file if those values are not general settings, but values for one-time use, such as the name of an image file that you are about to create or the last sector on a storage device to be covered when creating an image. On

the other hand, settings such as compression method and strength as well as block and segment sizes are probably settings that you keep using for a longer time unless you change your preferences. .dlg files created by different versions of the application are compatible with each other except if the dialog window controls have changed.

The .dlg file functionality is not available for the Directory Browser Options dialog window. For the directory browser options please save and load .settings files by clicking the icons in the directory browser caption line. The functionality to store dialog window selections in files is very useful for example for the Export List command, where some users repeatedly need different settings for different purposes, and where the items in the list box are always the same (just the available columns), except after changing the language of the user interface.

3.10 Command Line Parameters

1) You can simply specify the names of files that you wish to open automatically as command line parameters, including path if necessary. Physical disks can also be opened, e.g. specify :0 for hard disk 0.

2) The command line can be used to run file editing scripts. Just specify the .whs script filename as a parameter. It will be executed instead of opened.

3) You can run an X-Tension with a command named "XT", followed by a colon and the path and filename of the X-Tension.

4) The command line can be used to open an existing case. Just specify the .xfc case filename as the first parameter. You can add images to such a case with the AddImage: command (see below). If a parameter in the command line is the path or name of an .xfc file, and it's not the first parameter, and if at that point when the parameter is processed a case is already open, then the evidence objects of that .xfc file will be imported into the already active case.

5) The command line can be used in X-Ways Forensics (not X-Ways Investigator) to automatically a) create a case, b) add images, storage devices, directories and files, c) refine the volume snapshot of all added evidence objects, and d) run keyword searches. Example:
xwforensics64.exe "NewCase:D:\Cases\My case" "AddImage:Z:\Images*.e01" "AddImage:Z:\Images\My image.dd" RVS:~ auto

If no path is specified for the case, it will be created in the default directory for cases. The quotation marks are required only for parameters that contain spaces. If "NewCase" is followed by a semicolon instead of a colon, that generates a unique filename if the specified .xfc file already exists. With a colon, the existing case is deleted and overwritten (without prompt or mercy). The "NewCase" command supports relative case paths as well as references to environment variables.

To refine the volume snapshot of all evidence objects in the case ("RVS:~" command) or only newly added evidence objects ("RVS:~+"), X-Ways Forensics will run the same operations and use the same settings as were applied to a "virgin" (i.e. completely unrefined) volume snapshot

last time according to the WinHex.cfg file. A screenshot of the dialog window with the active refinement settings is automatically included in the case activity log. It's either textual or graphical in nature depending on your case activity log settings. Text in message boxes that usually need to be clicked away by the user is redirected to the Messages window while processing the command line parameters AddImage and RVS. Dialog boxes, if any, would still pop up normally (see below for remedy).

The command "LST" allows to load a list of search terms. If followed by a colon and the name or complete path of a text file with 1 search term per line and if this precedes an RVS run with an implicitly triggered simultaneous search, the terms will be utilized for that search.

6) The AddImage command supports asterisks. It also supports optional sub-parameters to force interpretation of an image as either a physical, partitioned medium (P) or volume (V) and to force interpretation with a certain sector size, where the sector size is optional, e.g.

AddImage:#P#Z:\Images*.dd
AddImage:#P,4096#Z:\Images*.dd

If you don’t specify these sub-parameters, a dialog window might pop up to ask the user for this input, but only in some very rare cases, only if not obvious to X-Ways Forensics from the data in the first few sectors what kind of image it is and if the image was not created by X-Ways Forensics or X-Ways Imager and if the image is not in .e01 evidence file format (e.g. raw image). Only if all three conditions are met at the same time plus you do not specify the sub-parameters, the dialog window will pop up.

7) The "AddDir" command is followed by a colon, and after that you specify which directory you wish to add to the case, e.g. AddDir:X:\. If the character after the colon in an asterisk, the root directories of all available drive letters will be added to the case: AddDir:*. However, network drives are optional because they can be excessively large and slow to explore. Addition of network drives depends on a new option in Options | Volume Snapshot. If you run X-Ways Forensics from a volume that has a drive letter, that drive letter will be ignored, assuming that you are doing this to triage a live system and run X-Ways Forensics from your own removable device. This command may also be used to add a single file to the case.

8) The "AddDrive" command is followed by a colon, and after that you specify which drive letter you wish to add to the case, in upper case, e.g. AddDrive:C. Unlike a directory, which is accessed and explored through the operating system, drive letters require sector-level access (and therefore administrator rights), and any present file system will be parsed by X-Ways Forensics itself, if supported. If the character after the colon in an asterisk, all available drive letters in the system will be added to the case: AddDrive:*. Network drives are optional again, and the drive letter with X-Ways Forensics will be skipped. If you specify the AddDrive:* command although you run the software without administrator rights, then the AddDir:* command will be run instead. If network drives are encountered as part of AddDrive:*, they will be added with the AddDir command internally as they have to be explored by the operating system without sector level access.

9) If you wish to apply different settings to different kinds of cases, you need to store these settings in separate WinHex.cfg files (in different directories or with different names) and restore

the desired one before executing X-Ways Forensics. Or you can use the command line parameter "Cfg:", which determines the name (not path) of the configuration file from which X-Ways Forensics will read during start-up and to which it will write when terminating, in situations when you need to use an alternative configuration (not the one stored in the main WinHex.cfg file). For example useful if for automated processing you need different settings than for manual execution, with specific volume snapshot refinement operations selected or to avoid the prompt whether a second instance should be started. Such a parameter looks like "Cfg:My other settings.cfg". As always, the quotation marks are required only if the name contains spaces. The maximum length of the name is 31 characters. Only ANSI/ASCII characters are currently supported. Command line parameters are usually processed in the order in which you specify them, except the Cfg: parameter is always processed before all the others, so it does not matter where it goes. Also, please note that a few settings are stored in other files, e.g. "X-Tensions.txt" and "Unwanted Metadata.txt".

10) You can load dialog window selections. This will usually override specific parts of the configuration that is initially read from a WinHex.cfg file, at the moment when the command line parameter is processed (not when those parts of the configuration might affect what the application does). The command is "Dlg:", directly followed by the path of the .dlg file. Relative paths are supported, and you may use file masks to load multiple .dlg files in the same directory at the same time. After you save dialog window selections for future use with the command line please verify that they can be accepted by clicking OK after saving them. Only .dlg files created in v20.2 and later can be used.

11) A command line parameter named "Override" overrides message boxes and dialog boxes until the last command line parameter has been processed. The text of those boxes will be output to the Messages window (and thus indirectly also to msglog.txt, unless disabled), and either an automatic click on OK will be simulated (if the parameter is "Override:1") or a click on Cancel (in case of "Override:2"). If a message box has only one button, it does not matter which parameter value was specified. All of this helps to avoid interruptions and delays of automated unsupervised processing when the program is waiting for user input.

The default setting and recommended behavior (if no Override parameter is specified) is like "Override:0", where message boxes and dialog boxes are shown normally and potentially alert the user of critical error conditions and anomalies such as incomplete images, undetectable image format etc. The parameter takes effect immediately upon start-up, before regular processing of other parameters begins, even if the Override parameter is specified last in the command line.

BitLocker password prompts are also skipped with either value 1 or 2. Or you can make X-Ways Forensics try the internal password collection (in Passwords.txt) if a value of 4 is combined with the usual 1, which gives Override:5. [Note that Override:5 is not compatible with versions of X-Ways Forensics before v21.6.]

The Override parameter also outputs the entire command line to the Messages window (even with the value "0"), and this happens at a time that depends on the position of the parameter within the command line. This allows users who study the log later to know what the simulated response to the suppressed message boxes and dialog boxes was.

12) It is also possible to image a physical device (e.g. local hard disk or remote hard disk or

RAM opened through F-Response) automatically via the command line, in X-Ways Imager and X-Ways Forensics. The first parameter should start with a colon and then specify the number of the device in Windows (e.g. ":1" for hard disk No. 1, i.e. the second hard disk). This will cause that device to be opened automatically upon start-up. The second parameter should start with a pipe, followed by either "e01" or "raw" to indicate the preferred image file format, followed by another pipe and the path and filename of the image, then optionally followed by a description and the examiner name (e.g. "|e01|G:\Output filename.e01|My description|My name"). You may create two copies of an image files if you like: The path of the second copy can be appended after the path of the first copy, delimited by a forward slash. Example: "|e01|Z:\First Copy.e01/V:\Second Copy.e01".

13) The last parameter can be "auto" if you wish to automatically exit X-Ways Forensics when finished.

14) Third-party tools that control X-Ways Forensics from outside via command line parameters may specify the command line parameter "GetLicID:" to find out the so-called nLicID, a hash value that uniquely identifies a dongle or a BYOD license. Nothing else will be done in a session started with that parameter, and X-Ways Forensics exits automatically. You could license your tool based on that ID and only allow use of your tool if the ID matches your expectations (if the ID is in your unlock list, if the user has a key file for that ID etc.). The first 4 bytes of the nLicID are returned as an exit code. Additionally, the full 16 bytes of the nLicID plus an 8-byte FILETIME value with the current timestamp in UTC can be written to a file whose path you designate optionally after the colon of "GetLicID:". By providing a unique, randomly generated filename, you can make extra sure that you get a freshly generated output file with an up-to-date nLicID and not a static, potentially outdated or manipulated value. And/or you can compare the first four bytes stored in the file with the exit code to make sure they match and/or check that the timestamp is not older than a second or so. If the first four bytes are all 0x00, that means that the X-Ways Forensics installation is not unlocked or that (re)writing the output file (if requested) has failed.

If you run X-Ways Forensics in an automated, unmonitored way and check the exit code, this can give your script an idea whether the execution was generally successful (0 or 710) or failed for some reason that you need to look into. Various failure indicating exit codes are possible, and they can be interpreted roughly in accordance with the error codes that Microsoft defines on the page https://learn.microsoft.com/en-us/windows/win32/debug/system-error-codes--0-499- and the following pages. In particular you could prepare for the following exit codes:
0 = normal program termination, no special exit code was set
353 = maximum simultaneous user count exceeded according to network dongle
710 = normal unlocked execution was possible
1067 = execution was aborted in an unexpected manner
10003 = dongle not found
11111 = server with network dongle not found

3.11 User-Defined Keyboard Shortcuts

There is a button in the Options | General dialog window that you can click to define up to 20

custom keyboard shortcuts for commands in the directory browser context menu and elsewhere. Currently available only in X-Ways Forensics. Shortcuts are meant to increase your productivity when performing your most frequently used activities. Only key combinations that involve the keys Ctrl, Ctrl+Alt, Alt Gr, Shift and Space are supported. Please note that if you use the Space key for any keyboard shortcut, you cannot use it any more to tag or untag items. The second key can be relatively freely chosen by just pressing it when the grayed out edit box has the input focus. In case no human-readable description of the selected key is provided and you later forget what key you had defined, you can check out this list of hexadecimal key codes: https://msdn.microsoft.com/en-us/library/windows/desktop/dd375731(v=vs.85).aspx

The dialog window reveals the ID of the last command used. So in order to find out the ID of the command that you wish to generate a shortcut for, you just need invoke that command (you can cancel it if that is an option), then immediately click your way through to the dialog window and can then see its ID. Most commands invoked in the main window, in a data window, in the directory browser or in the Case Data window are suitable.

There is a special ID that you can use to repeat the last command invoked, whichever that may have been. For example if you wish to manually categorize files as notable in multiple steps, you can do so through the directory browser context menu the first time, and after that just press the special key combination that you have defined for that ID. In fresh installations, the keyboard shortcut Ctrl+F5 is predefined to repeat the last action.

The following ~80 directory browser **menu command codes** can theoretically be used (not all tested) and have to be entered as a number:

9800: View with external viewer program #1
9801: View with external viewer program #2
9802: View with external viewer program #3
...
9831: View with external viewer program #32

9919: Define file type
9920: Go to related file
9921: Refine volume snapshot for selected files
9927: Run X-Tension on selected files
9928: Attach external file
9931: Edit metadata
9932: See this file in its directory
9933: See this file from volume root
9934: Find parent object
9935: Logical search within selected files
9937: Attach external directory
9938: Erase securely
9939: Leave search hit list for specific directory
9940: Delete duplicate search hits in list
9941: Select excluded items
9942: Edit comment
9944: Include

9945: Select tagged items
9946: Exclude all except tagged items
9947: Exclude tagged items
9948: Add to evidence file container OR skeleton image if active in the background
9949: Resize search hit
9950: Convert search hit to carved file
9951: Resize carved and virtual files
9952: Assign search hit to other search term
9953: Extract consecutive video frames
9954: Include search hit in report
9955: Mount as drive letter (makes sense only if a directory is selected, and only one)
9956: Watch with preferred video player
9957: View with preferred HTML viewer
9958: View with preferred text editor
9959: Execute/open in associated external program
9960: Select viewed items
9961: View with to-be-selected external program
9962: Remove duplicates based on hash
9963: Seek item based on int. ID
9964: Sort by relevance
9965: Print
9966: Seek item based on list item number
9967: Sort by nothing
9968: Select all
9969: Filter by the selected file's hash value (to find duplicates)
9971: Explore
9972: Mark search hit as notable
9973: Open
9974: Navigate to defining data structure
9975: Export list
9976: List clusters
9977: Recover/copy
9978: Explore/view
9979: Invert selection
9980: Include in hash database

You will notice a few suspicious gaps in between the incrementing numbers. The missing numbers are either unassigned or discouraged to invoke or simply don't make much sense to define for a keyboard shortcut. As an example for the latter, 9929 will delete selected search hits or events, something that can of course be accomplished already by pressing the Del key. This information shall reduce your urge to randomly try numbers not listed here, although who knows whether one undocumented number may trigger a secret "Find all evidence" command.

Please note that even without defining any such keyboard shortcut you can reach all directory browser context menu commands purely with the keyboard by pressing the context menu key. (Usually to be found between the right-hand Windows key and the right-hand Ctrl key.) Some menu commands already have a predefined keyboard shortcut. For example the Enter key is the same as a double click (either View or Explore, depending on your settings). The multiplication

key of numeric keypad triggers the Explore command. Del means Exclude. Ctrl+Del resets files to the "still to be processed by volume snapshot refinement" state and undoes some refinement operations. Ctrl+Shift+Del removes hash set matches, hash category, and PhotoDNA categorization. Ctrl+Caps Lock+Del removes the "file contents unknown" flag from a file. (Useful for example if because of temporary I/O problems X-Ways Forensics marked files that way although generally the files can be read just fine.) Ctrl+C copies the selected items into the clipboard using special settings of the Export List dialog window.

Main menu

The user-defined keyboard shortcuts should be able to invoke practically all commands from the main menu as well, and even if parts of the user interface other than the directory browser have the input focus. If the command code of a menu command changes in a future version, X-Ways Forensics will ensure that any keyboard shortcut targeting that code will automatically become inactive, to prevent accidental misuse. To find out the command codes of commands in the main menu (also called IDs of menu items), you can open the main executable file in a so-called resource editor and have a look at the menu resource in your preferred language. A highly recommendable light-weight example of such a tool is "Pelles C for Windows", which also happens to be a fine C compiler and complete development kit suitable for creating X-Tensions. Keyboard shortcuts for main menu commands should be less important than for directory browser context menu commands because the main menu already has many dedicated keyboard shortcut predefined, or even if not can be reached without taking one's hands off the keyboard starting with the Alt key. To give you some ideas about useful applications, FYI the command code to toggle between recursive and non-recursive exploration is 122, and the command code to take a new volume snapshot is 109.

Command codes defined for filters
(The order is the historical order in which filters were introduced.)

9700: Name
9701: Type
9702: Type status
9703: Category
9704: Size
9705: Path
9706: Sender
9707: Recipients
9708: Timestamp
9709: Attr
9710: Hash 1
9711: Hash set
9712: Hash category
9713: Labels
9714: Comments
9715: Metadata
9716: Analysis
9717: Pixels
9718: Int. ID

9719: Unique ID
9720: Search terms
9721: Owner
9722: Parent name
9723: Child objects
9724: ID
9725: Author
9726: Search hit description
9727: Event timestamp
9728: Event type
9729: Event description
9730: Search hit
9731: First sector
9732: Description
9733: Hash 2
9734: Full path
9735: Flex filter 1
9736: Flex filter 2

Command codes for the Mode buttons and related buttons

122: Recursive exploration
138: Access button popup menu
172: Toggle Directory Browser
186: Toggle Position Manager
223: Toggle Search Hit List
224: Toggle Event Hit List
225: Disk/Partition/Volume/Container mode
226: File mode
227: Preview mode
228: Details mode
229: Gallery mode
230: Calendar mode
231: Legend mode
232: Sync mode
249: Raw preview mode
250: Viewer X-Tension preview mode

4 Menu Reference

Note: Commands in the main menu (File, Edit, Search, ...) always apply to the active data window as a whole (which e.g. represents an open file or an open disk), or to files/disks that are still to be specified by the user. They never apply to the file(s) currently selected in the directory browser. That's what the directory browser context menu is there for.

4.1 Directory Browser Context Menu

The directory browser context menu allows the user to directly interact with the currently *selected* files/directories, notably *not* the *tagged* items. There are a number of menu commands which are available depending on the selected items. Double-clicking files and directories will, depending on the circumstances, either invoke “View”, “Explore” or the associated external program.

View

This command allows viewing the selected file with the internal viewers for Windows Registry files and various graphical file formats. If the separate viewer component that comes with X-Ways Forensics is active, all other files are sent to that viewer. If it is not, the first installed external program will be called instead. NTFS system files are always opened as data windows.

When viewing a file in a separate window, you may press (Ctrl+) Page Dn/Up to close the window and view the next file in the directory browser in a new window. If a View window displays a picture and viewing pictures if limited to one picture at a time, that window will be updated when you press the cursor keys in the gallery. Useful especially on a spanned desktop, if the View window is centered on the second monitor and if the gallery is on the first monitor. Avoids having to press the Enter key to view the picture and another key to close the View window to get the input focus back to the gallery.

When viewing pictures with the internal graphics display library, you have a choice to either center such view windows on the screen or remember their left top position or their center position after you move them somewhere else on the screen. To make your choice, open the system menu of the view window (i.e. click the icon in the left top corner of the window). You can also decide whether or not such view windows should always be in the foreground, even in front of windows of other applications. Last no least you can choose to roughly remember the window size. Especially useful in conjunction with the options to remember the left top position of the view window, to have only one view window at a time, and to update the view window automatically with just a single click on a file, so that at place on your screen of your own choice you essentially have a fixed preview of pictures while the lower half of the data window can show something other than Preview mode, for example Details mode.

Explore

Only available for directories and archives (ZIP, RAR, TAR...), this command allows navigating into them within the directory browser. Double-clicking archives or directories does the same. A command that allows listing the contents of directories as well as their subdirectories at the same time can be found in the directory tree's context menu instead (in the Case Data window, "Explore recursively").

Viewer Programs

Allows to send the selected file(s) to one of the external programs currently configured or the file's associated program in the current Windows installation. This association is determined

based on file extension as is usual within Windows.
You also have the option to open files in an external program that you select ad hoc. The program that you select will be saved as a standard custom viewer program if you have not used all slots for external viewer programs yet, and then also remembered for next time when you invoke the same menu command.

Open

Opens currently selected files or directories in separate data windows. Unlike File | Open, where files can be opened just like in any other application with the help of the operating system, this is a forensically sound operation in that it does not update any timestamps etc. because the operating system is circumvented and the logic to read the file's contents from the correct disk sectors is implemented in WinHex itself for various file systems. No changes can be made to files that were opened in this fashion, however. In the case of a directory, the directory's data structures will be opened.

Print

If the separate viewer component is active, you may select files (or even directories) for printing. You can print multiple selected documents without interruption/the need to click somewhere after each document, optionally along with child objects (e.g. e-mail attachments together with their respective e-mail message). The optional cover page contains the date and time when the print job was started and selected metadata, e.g. filename, path, evidence object title, file size, description, time stamps, comments, ... A preview of the file can be printed in the bottom part of the cover page. The format of this preview depends on the settings of the viewer component in Preview mode, e.g. "Best Fit" or "Actual Pixels" or "Fit to Window Width" etc. This is a 3-state check box. If only half checked, the preview is printed in much lighter colors, either to save ink/toner or to improve readability of the metadata fields if you output many of those and they spill over onto the preview. The cover page is printed by X-Ways Forensics itself, the following pages with the actual document are printed by the viewer component. You can print just the cover page or the selected file or both. The header line of the cover page, which specifies which user and which program and version created the print job, is optional. Useful if you wish to show the printout to witnesses or the suspect who should not know the username of the examiner.

You have the option to print the contents of Details mode (any of the four available submodes) for the selected files, where the check mark refers to the complete Details mode, including all cells of the directory browser, which may render the inclusion of a cover page obsolete. Another option is to have X-Ways Forensics print the filename and path on the first page. This option is not bound by the same path length limitations as the header optionally printed by the viewer component ("Print header", Job Name = %p). To avoid that the path is printed twice on the first page, have either X-Ways Forensics or the viewer component print it, not both.

Recover/Copy: see separate topic

Export List

Requires a specialist license. Exports data about the selected items in the directory browser to a tab-delimited text file or to an HTML file, which can be easily viewed in any web browser, also

imported and further processed e.g. in MS Excel and MS Word. A third option (except for search hit lists) is an XML file. The list can alternatively be copied into the clipboard in the format as chosen, for example to paste it directly into an externally edited report. The columns to export are freely selectable. Even the search hit column can be exported, with the textual context around each and every actual hit, where the search term itself can be visually highlighted with a yellow background color (not recommended for output to MS Excel). You may choose to split up the result into multiple files for example to avoid a huge HTML file that Internet browsers will choke on.

There is an option to copy files off the disk/image and link them from the HTML output. The links can be found in the Name column. The exact behavior depends on two case report options: "Name output files after" and "Embed attachments in parent .eml file". This option is an interesting layout alternative to the regular output of report tables in a report and also an alternative to the Recover/Copy command.

In the Name column you can output the original name, the alternative name if an alternative name exists (if the box for that is fully checked) or both (if the checkbox is in its middle state). When exporting a list of files or directories along with their child objects sorted by full path, so that child objects directly follow their respective parents, in TSV or HTML format, an option called "Indention" allows to indent the names of the child objects so that it is easy to see in the output which objects are child objects of which other objects even when not looking at or when not even including the potentially very long full path as an additional column. The indention can be strong (fully checked) or not so strong (half checked).

The Export List command remembers its own notation settings, different from the notation settings in the General Options. That is useful because the database or spreadsheet program of your choice in which you wish to import the data may not like the formatting that you prefer to see in the directory browser (e.g. fractions of seconds in timestamps, time zone bias, weekdays in dates, delimiter between date and time, integer digit grouping, ...). While the Export list dialog window is on the screen, the directory browser in the background reflects the notation settings of the Export List command, as a kind of preview.

Copy: Extracted text

Allows to copy text that is decoded or OCRed from selected files to other places. The scope can be limited to files that specifically need OCR (i.e. pictures and certain PDFs) if you are only after such files. The extracted text can be buffered internally in the volume snapshot for future logical searches or indexing and the context preview of search hits. It can be copied into comments of the respective files (suitable esp. for small amounts of text OCRed from pictures), for example to include the text in the case report or exported lists, optionally with an explanatory prefix like [OCR] or [Extracted text]. The extracted text can also be output as child objects (text files). Or it can be collected in a single text file on your own storage device, or copied into the clipboard, and any combination of the above is also possible.

Extract consecutive frames

Extracts all frames specifically from a defined section of a selected video. Useful if a certain part of a video is of high interest and you need to carefully check visual details in certain frames or

include them in the report. You can specify how many consecutive frames to extract and starting from which second. The number of frames that you need to cover a certain period of time can be deducted from the frame rate as shown in the Metadata cell (fps = frames per second). Please note that the start second may be interpreted very roughly only, depending on the frequency of keyframes (a.k.a. I-frames in MPEG) in the video. MPlayer can seek into a video file only based on keyframes. If for example a certain video file contains keyframes only every 4 seconds for example, then the start second of the extraction may be off by up to 4 seconds. Keep this in mind when you enter the number of frames that you need or the start second. That is, to be on the safe side, extract more frames than you may actually need and perhaps from an earlier start second.

The frames are saved as JPEG files in a directory of your choice on your own drive, where you can review them outside of X-Ways Forensics. If you like, you can of course attach the most relevant frames to the original video file in the volume snapshot as child objects. The frames are not stored within the volume snapshot by default so that the size of the volume snapshot does not unreasonably inflate with potentially mostly irrelevant and redundant pictures. If the output directory already contains extracted frames, files with identical relative frame numbers will be overwritten. Relative frame numbers always start with 00000001 for each extraction and increment with each frame. You may adjust the JPEG compression if necessary for stronger compression or better quality. (Of course you usually cannot expect a very good quality because videos are typically highly compressed already.)

Labels: for labels, see above

Edit Comment

Requires a forensic license. Use this command to add a comment to an item in the directory browser or to edit or remove an existing comment. When applying the same comment text to multiple selected files, you have a choice and can either replace the existing comments or append to them or prepend text, delimited in various ways. After entering comments, you can conveniently set the filter such that only commented items are shown or only items with specific comments, e.g. those with a certain relevance.

Edit Metadata

Requires a forensic license. Allows to edit the metadata field of a file once metadata was extracted. Useful if you wish to include selected metadata (not all extracted metadata) in a report.

Categorization

Files in the volume snapshot are by default considered unknown. This status may change to irrelevant, notable or uncategorized based on hash database matching, based on X-Tensions, by adopting data from evidence file containers, by using this submenu, and in other ways. The status can be seen in the Categorization column.

Tag/Untag

Requires a forensic license. Tagging files means highlighting them visually (placing a blue square at the beginning of a directory browser item), for various reasons, e.g. to mark them as

relevant, or memorize a position in a sorted list, or to limit volume snapshot refinements to tagged files. *Tagging* is not to be confused with *selecting*.

Select

You can select files based on various criteria in order to apply context menu commands to them. For example you could select all the files that you have tagged before and that are currently listed. One of the menu commands in this section enables you to "compare" directory listings, i.e. find out which files or subdirectories are present in the current listing in the active data window that are "missing" in another. This will match by name and optionally also require identical sizes, identical modification timestamps, and/or the same general existence status (existing vs. previously existing [of any kind]). Once the non-matching / additional items are selected, you can easily copy a list of them (Ctrl+C), export a list of them, copy them, hash them etc. This works with non-recursive and recursive listings alike.

Find duplicates in list: cf. topic Duplicate File Detection

Filter for duplicates

Ability to filter for duplicates of a single selected file that are also currently listed in the directory browser, only if a hash value is available for the selected file and the other files. Actually filters for that hash value at that time, and thus does not depend on previous mass identification of duplicate files using the above-mentioned command "Find duplicates in list". In X-Ways Investigator the actual hash values are not displayed and cannot be computed, but they are imported from evidence file containers that come with hash values for files and can be used to identify duplicate files.

Filter for similar files

Uses the "Structure type" filter to find files of the same type, likely and roughly created around the same time by the same application or device with the same settings or for the same purpose etc. This functionality is available only once the "Structure type" column has been populated, for supported file types.

In search hit lists you may
1) permanently delete *selected* search hits,
2) permanently delete *duplicate* search hits. Search hits are considered duplicates if they either have identical physical offsets or, if they don't have physical offsets, if their logical offsets and the corresponding internal file IDs are the same. When in doubt, X-Ways Forensics will keep the longer search hit (as "Smithsonian" for example is more specific than "Smith") and favors search hits in existing files.
3) Resize: Allows to resize or reposition the selected search hits. If for example you are searching for a signature that identifies records in some kind of database, and you get many search hits for these signatures, but what you are really interested in is the record data that follows the signature, and you wish to export that data, then you could adjust the offsets and the lengths of the search hits in a suitable way. Also, instead of exporting more context around the search hits with the Export List command, you could enlarge the search hits themselves prior to exporting them. The effect is visible immediately in the search hit preview in the search hit list (but not necessarily

immediately in the highlighting in the lower half of the data window).
4) Another context menu command in search hit lists allows to convert search hits to carved files. Useful if you wish to include your search hits as files in a report, label them in some way, comment on them, print the contents, Recover/Copy them etc. Note that search hits that have both a physical and a logical offsets will be carved at the sector level and will appear in the virtual directory for carved files. Search hits that only have a logical offset will be carved within the file in which they were found and will appear as a child object. Search hits in the decoded text of a file as well as search hits in directory browser columns cannot be carved and will be omitted.
5) Assign to other search term: Ability to categorize selected search hits by moving them over to other search terms, existing or new ones. If for example you get several relevant hits when running a search for the search term "invoice", and some hits are relevant in a different way than others, then you could assign them to other search terms like "Invoice ABC Ltd.", "Invoice XYZ Corp." etc. Those newly created search terms will appear in the search term list, but they function more like categories because they were not searched for literally themselves.

Add to Bookmarks

You can bookmark a selected file or directory and enter an individual description. Bookmarks are useful to quickly navigate back to an item of interest. To see a list of all bookmarks in the case, use the Edit menu of the Case Data window. All bookmarks can be seen and navigated to even if the evidence objects to which they refer are not currently open. When you create a bookmark, that creates a label at the same time, which is useful for filtering and because creating a backup of the volume snapshot and restoring such a backup will at least back up and restore the label (though not the bookmark).

Navigation

Commands in this submenu allows to sort files by their estimated relevance (cf. metadata extraction) or to avoid wasting time on sorting.

The Navigation group of commands also allows interaction with the currently selected file at a generally more technical level. It allows to directly locate the data structure in the file system that defines a file (e.g. FILE record in NTFS, inode in Ext2/Ext3/Ext4, directory entry in FAT).

The Navigation menu also allows to produce a list of all the clusters allocated to the selected file or directory. From the context menu of that list window, the cluster list can be exported to a text file. The context menu also offers the option to shortened the list and accelerate its creation by omitting clusters in the middle of a fragment. Omissions are indicated by ellipses. This option takes effect only when you produce a cluster list the next time. The concise representation is useful if you are merely interested in from where to where each contiguous series of cluster (=each fragment) runs.

Find parent object: Navigates to and selects the parent object of the selected object. Equivalent to pressing the Backspace key. The child object can be an ordinary file in a directory, or an e-mail message in an e-mail archive or a file attachment in an e-mail message or a picture in a document or a file in a compressed archive etc.

Find related object: This command allows you to conveniently navigate to the so-called related object if one exists for the selected file or directory. Alternatively, you can press Shift+Backspace.

See selected item in its directory: Will show you the selected file or directory among its siblings. Useful to quickly check out whether there are more notable files in the same directory or to better understand the function of the file when you see it in context.
See selected item from volume root: Will show you the selected file among all other files in the same volume, recursively explored from the root of that file system. Useful for example to see whether there are any files with the same name, the same ID (e.g. previous version from a volume shadow copy), same owner, same sender, or similar timestamps etc. in the same file system (just sort accordingly).
Both commands can be also be used from within the case root window and from within search hit lists (so the previous "Go to file in directory browser" command becomes obsolete). Remember you can click the Back button in the toolbar to conveniently return to the previous view.

Seek Path helps to locate a file or directory in the directory browser whose full path you specify. "Seek Int. ID" allows to conveniently seek the item with a given internal ID, no matter whether file or directory. If a filter prevents listing that item, all filters will be deactivated automatically.

"Seek Item #" will jump to the item that has the specified position in the current listing. The position of any item in the list is shown when you hover the mouse cursor over the icon of a file or directory. This command can optionally preserve the selection in the directory browser and merely move the indicated item number into view and highlight it. Note that if you wish to open the context menu without losing your selection, if you can't see your selection at the moment because you have scrolled up or down, right-clicking any unselected item in the directory browser with the Ctrl key pressed will achieve that. Alternatively, you can press the context menu key on your keyboard. The Seek Item # command has two extra buttons that allow you to easily navigate to the first and the last selected item in the directory browser. That is useful in a very long list of files if you have scrolled elsewhere, but need to return to a spot where you had selected files.

Exclude/Include

You may exclude selected items (press Del) or all tagged or all untagged items. If actually filtered out, excluded files are omitted from the directory browser, the gallery view, and all commands that can be run from the directory browser context menu. If you are only allowed to examine the contents of certain directories, you could initially exclude all files in all other directories to ensure that. Refining the volume snapshot can be limited to files that are not excluded. Excluded items are actually filtered out only if the corresponding filter is enabled in the directory browser options. If not filtered out, they are listed in gray and can be included again with the directory browser context menu or by pressing Shift+Del.

"Include all" allows to undo the exclusion of all files and directories in the volume snapshot of the evidence object in the active data window. To selectively include files, make sure they are not filtered out. Then you can include them with a context menu command after selecting them.

It is also possible to totally remove excluded items from the volume snapshot if irrelevant/not

needed, in particular meaningless garbage files found via a file header signature search. This will render the volume snapshot smaller, i.e. more efficient to handle, and save main memory. Useful also if you would like X-Ways Forensics to find certain files once again via a file header signature search, but for example list them with a different default file size if the originally specified default file size proved inadequate. The removal operation is faster if you delete seach hits prior to executing it. As part of the removal, internal IDs are shuffled, so they do not indicate any more the order in which items were added to the volume snapshot. Excluded items that have non-excluded child objects are not removed. It is highly recommended to work with a copy of your case when using this functionality, e.g. produced with the Save As command.

Refine Volume Snapshot, Simultaneous Search, Run X-Tensions

These commands are known from the main menu. From the directory browser context menu they can be applied to the *selected* files.

Include in Hash Database

Creates a hash set of the currently selected files and directories and their subdirectories directly within the internal hash database, either with ordinary file hash values or with block hash values or PhotoDNA hash values. For ordinary hash values there is an option to create multiple hash sets in a single step, where the hash values of the selected files are put into hash sets that are named after each file's label(s). This is useful if you categorize notable files in one case using labels (e.g. based on different types of CP), and wish to quickly identify the same files again in other cases later, and automatically see the category that you had originally assigned, as the hash set name.

The checkbox for that is "Name after labels, if any". If a selected file does not have any label, its hash value will be assigned to the hash set named as you specify, just like if you do not check that checkbox.

This command can also be used to create a separate file with PhotoDNA hash values of the selected files or to just update file descriptions of files in the PhotoDNA hash database with the comments stored in the volume snapshot.

Attach External File/Dir

Requires a forensic license. Ability to attach one or more external files or a directory including subdirectories to the volume snapshot and have them processed by X-Ways Forensics like regular files in the volume snapshot. Useful if you need to translate, convert, or decrypt original files and would like to reintegrate the result back in the original volume snapshot, in the original path, for further examination, reporting, filtering, searches etc. Such files will be completely managed by X-Ways Forensics once attached, with their contents copied to the internal evidence object subdirectory of the case, so that the source files can be removed.

You will be asked to classify the files that you are attaching as what they actually are, e.g. video stills produced outside of X-Ways Forensics, e-mails extracted from e-mail archives outside of X-Ways Forensics, OLE2 objects, attachments of various kinds (in particular of PDF documents), etc. etc. If properly classified as video stills, the attached pictures will be used as previews for the respective parent video file for example. The classification can be seen in the

Description column.

When attaching a single external file and holding the Shift key, X-Ways Forensics proposes a new name for that file that is based on the name of the file that is selected, and the attached file will be added to the same directory. Otherwise the external filenames of the files will be used and they will become child objects of the selected object. It is still possible to rename virtual files in the volume snapshot later at any time.

When attaching an external directory to the volume snapshot, you are prompted whether the selected directory itself should also be attached or just its contents. Usually X-Ways Forensics creates virtual files in subdirectories in new virtual directories in the volume snapshot. There is, however, an option to accommodate the files in existing directories in the volume snapshot of the same name at the same position in the directory tree. Useful if you copy an entire directory structure off the image to convert/decrypt/translate/... files outside of X-Ways Forensics, and then want to bring the results back into the volume snapshot and see the edited files next to their original counterparts in the corresponding subdirectories. This can help for example if you wish to OCR and convert PDF documents that X-Ways Forensics has deemed non-searchable, using Adobe Acrobat.

X-Ways Forensics can optionally adopt the timestamps of attached files in the volume snapshot (creation, modification and/or access). You can make use of this if you are sure that the timestamps are original and not the result of any of your own file copy/decoding/decryption activity etc.

Rename

Allows you to rename virtual directories and virtual attached files in a volume snapshot, or if the Shift key is pressed even ordinary files. Although the latter is not exactly forensically sound when dealing with original evidence, this can prove helpful in special situations, for example if a filename or directory name is too long to copy a file out of an image etc. The original filename will be kept as the alternative filename. Note that this does not rename the file in the file system (nothing is altered on the disk or in the image!), only in the volume snapshot, i.e. the internal database in X-Ways Forensics *about* the file system.

Specify type

Ability to specify the type of selected files yourself. Useful if you wish to identify types or subtypes in an individual way unknown to X-Ways Forensics, for example to be able to filter by these types later. For instance, how about categorizing TIFF pictures that are digitally stored faxes as type "fax"? Remember you can define your own file types in File Type Categories.txt.

Resize

Files found through a file header signature search and files that were carved within other files can be manually redefined by the user. You can reposition such files with a relative offset change (+/-), and/or to resize them, with either an absolute new size or with a positive or negative relative size adjustment (click the arrow button to toggle). You can resize multiple files at the same time with the same settings.

Redact data

Files and directories that are selected in the directory browser can be securely wiped in WinHex (not X-Ways Forensics). For files the scope can include or exclude the slack. Wiping clusters of a directory (e.g. containing INDX buffers in NTFS and directory entries in FAT) is optional and not active by default, so that you could conveniently and safely select entire directories to get only all the files in those directories wiped. Users are warned when they enable wiping of directory clusters as it will corrupt the file system and leave files orphaned. Also keep in mind that after you have purged directories, depending on which exact file system it is, X-Ways Forensics itself may be unable to find the files again when taking a new volume snapshot. The existence status of a file in its file system will not be changed if you merely wipe the file contents, i.e. the file will not be marked as deleted, the clusters will not be released etc. No file system level metadata such as timestamps or attributes will updated because no operating system file level write commands are used. No file system data structures are changed, and no filenames will be erased, only the contents of files will be overwritten. The only exception is that in case of NTFS the FILE records of the selected files in the MFT can optionally be erased additionally. Files that are compressed in archives or generally files within other files (e.g. e-mails and attachments in e-mail archives) cannot be erased. Previously existing files whose clusters are known to have been reused will not be erased. Note that by erasing deleted files you might erase data in clusters that belong to other files, so only select existing files if you want to avoid that (assuming consistent file systems). Also note that by erasing carved files you may erase too much or not enough data, depending on the detected file size and depending on whether the file was originally fragmented. And please note that wiping directories, i.e. erasing the data in the clusters allocated to a directory, will cause existing files in that directory to become orphaned. More typically users only wipe the contents of files with this function, not the contents (data) of directories, if they still wish to use the file system.

Useful for example if copies of images are forwarded to investigators/examiners/other parties involved in a case who are not allowed to see the contents of certain files. Useful also if you have to return computer media on which child pornography has been found to the owner after clearing these files. Also useful to selectively and partially sanitize your own physical storage devices for IT security purposes (if you do not expect or care about sector reallocation to occur upon write operations). Also useful if you are preparing images for training purposes that you would like to publish and if you would like to retroactively erase the contents of copyrighted files (e.g. operating system or application program files).

Both successfully erased files and files that could not be successfully erased will get labeled accordingly (when working with a case, with a forensic license only) by which you can filter to verify the result.

Mark hit as notable

In a search hit list, marks selected hits with a yellow flag and includes in them in the list of notable search hits. You may also press the space bar to mark a hit as notable or remove that mark. Holding the Shift key when invoking the menu command removes the "notable" flag from all selected search hits.

Include in report

In a search hit list, marks selected search hits for inclusion in the case report, with the green grid icon.

Hexadecimal

In a search hit list, you have the option to display search hits along with their context in hexadecimal notation. Useful especially for technical searches, i.e. not keyword searches, but searches for header signatures, delimiters, binary markers etc. The option will also affect the output of search hits in the "Export list" command.

4.2 Case Data Window Context Menu

Some of the commands:

Export subtree: This context menu command in the Case Data window allows you to export a pseudo-graphical representation of the selected subtree in a Unicode text file, which is best viewed with a fixed-width font. The exported tree reflects the current state of subdirectories (expanded or collapsed). The menu command is available for evidence objects and also for directories if you hold the Ctrl key when right-clicking a directory in the case tree. Remember to fully recursively expand a portion of the tree that you want to export, you can click the root of that portion and press the asterisk (multiplication) key on the numeric keypad.

Attach external files: This command allows to attach external files as child objects to their original counterparts (after decrypting, translation, convertion, OCRing, ...) in multiple evidence objects at the same time automatically if they are named after the unique ID of the original files. (The filename extension is ignored.) You can name the files after the unique ID when you copy them off the image with the Recover/Copy command, and you do not need to preserve the path, as the unique ID already fully identifies the file. Useful if you wish to apply external tools to the copied files which have problems with overlong paths, if you wish to bring back the result into the volume snapshot.

When attaching external files (e.g. after decrypting, converting, translating, ...), you are given four options:
1) the attached file can become a child object of the original file
or
2) the attached file can become a sibling of the original file (shown next to it, in the same directory)
or
3) the attached file can replace the original file (original file no longer present)
or
4) the attached file can replace the original file, and the original file can become a child object of the new file if still needed.
You can select the attachment method separately for ordinary files and e-mail attachments. The three latter methods are particularly useful for e-mail attachments because only direct child

objects of .eml files are embeddedd in the parent .eml file when recovering/copying those .eml files. So if you would like to have the decrypted/converted/translated version of an attachment embedded in the .eml file, that version should not become grandchild object. If you want original and new version both to be embedded, make them siblings. If you do not need the original version embedded, replace it completely or preserve it only as a child object of the new version (i.e. grandchild of the .eml file).

The attached files adopt the classification of the original files, e.g. as extracted e-mail messages or OLE2 objects. If the original files have no special classification, the attached files will be simply marked as attached files.

Export Files for Analysis: This menu command in the Case Data window can be applied to the entire case and from there to selected evidence objects, or to the active evidence object only. It uses the interface for external analysis of files to invoke external automated analysis tools such as DoublePics.

Seek Unique ID: Allows to locate the file or directory with a given unique ID. If the evidence object that contains that item is not open at that moment, it will be opened automatically.

There is a context menu for directories, too. It is displayed when right-clicking a directory depending on the General Options and depending on whether you hold the Shift key at the same time. Otherwise right-clicking a directory means to explore it recursively.

4.3 Data Window Context Menu

When you right-click the hex editor display (consisting of offset column, hex column, text column) of a file or a disk, you will get a context menu that allows you to define the boundaries of the block (start and end) and invoke a few more commands that apply to that block:

Add to User Search Hits: Forensic license only. Allows you to define search hits manually. Whenever you come across some relevant text, for example floating around in free space in Disk/Partition/Volume mode or within a certain file in File mode, you can select it as a block and right-click the block to add it as a so-called user search hit (i.e. some kind of search hit not found by the program). You can assign the search hit to an arbitrarily named search term/category. For example, if what you have found is related to suspect A, assign it as a search hit to a search term named after suspect A. If also related to suspect B, you can also assign it to another search term. You could also assign it to a real search term that you have used for an automatic search.
User search hits can be conveniently listed in and nicely exported from search hit lists just like ordinary (automatically generated) search hits. To distinguish then from ordinary search hits, in the search hit description column user search hits are marked with an asterisk (*). You can specify the correct code page for user search hits yourself when you define them, which may be essential to get the text displayed correctly. User search hits are stored related to an object in the volume snapshot if you define them in File mode. User search hits are forward compatible, i.e. older versions (v16.2 and later) can also see user search hits created by v16.6.

Add Block as Virtual File: Forensic license only. See Edit menu.

Add Position: Allows you to remember the position indicated by the currently defined block, either in the General Position Manager or in the Position Manager of the evidence object (when working with a case, if you right-click a block that is defined in an evidence object, forensic license only). Makes it easier to find the same position again later, and can be used to nicely highlight and explain (with tooltips) the structure of files or records of a certain format that you are analyzing/trying to reverse-engineer etc.

If search hits are highlighted in File mode (see General Options), you can also delete them via the context menu.

You can also get the complete Edit menu from here.

4.4 File Menu

New: This command is used to create a file. The file is principally opened in default edit mode. You have to specify the desired file size. Newly created files can be optionally held in memory instead of in a temporary file, for performance or storage quota or security reasons. If the checkbox is fully checked, that means the user insists on memory storage and the menu command fails if not enough memory is available (or not enough contiguous memory address space in case of the 32-bit edition). The same setting applies when pasting data from the clipboard into a new file via Edit | Clipboard Data | Paste Into New File. The Info Pane will show the buffer address in the memory address space instead of a path for the newly created file. Please note that the data you are handling could still be written to disk by Windows for example as part of pagefile.sys. Also, if your goal is to avoid disk I/O and the usage of temporary files, you may want to disable file backups for the Undo function, in Options | Undo.

In X-Ways Forensics you can also use this command to create dummy segments for .e01-Images.

Open: Lets you open one or more files. You may choose an edit mode if in-place edit mode is not active globally.

Also allows to open physical disks, partitions and volumes as a file, by clicking a button labeled "Device..." in the file selection dialog. You can enter a device path such as
\\.\PhysicalDrive1 (for hard disk 1)
\\?\Volume{12345678-9abc-11a1-abcd-0123456789ab} (for a volume with that GUID)
\\.\C: (for a volume mounted as drive letter C:)
This functionality allows to open volumes that are not mounted as drive letters. To get an overview of volumes known to Windows, type "mountvol" in a command prompt window. You can also try to open exotic devices supported by Windows such as tapes and changers (not tested). Also this is how you can open alternate data streams whose path and name you know, which cannot be opened through the ordinary File | Open dialog, without opening the volume on which they reside.

Opening a hard disk as a file can be useful for example if you wish to clone that disk and if source and destination disk have different sector sizes (whether it makes sense in the first place to

clone a hard disk despite the sector mismatch depends on the data). When treated as a file, there is no defined sector size and hence no possibility for a sector size mismatch. Device files can also be interpreted as disks like images can.

Save: Saves the currently displayed file to the disk. In in-place edit mode, using this command is not necessary. When using the disk editor, this command is named "Save Sectors".

Save As: Saves the currently displayed file under a different name.

Create Disk Image/Make Backup Copy: cf. "Images and Backups"

Create/Verify Skeleton Image: cf. "Skeleton Images"

Restore Image: Select an image that you would like to restore, i.e. whose sectors you would like to copy back to the original medium or some other medium, or select a or WinHex backup (.whx) file whose contents you would like to restore (could be a file or disk sectors). In the case of an image, the image will be preset as the source in the Clone Disk window (with a specialist license or higher, interpreted). Only with a specialist license or higher spanned raw images can be restored. Spanned WinHex backups can be restored with any kind of license.

Backup Manager: cf. "Backups"

Execute: Executes the current file if executable, or otherwise the associated program.

Print: Use this command to print a file, disk sectors or RAM contents. Define the printing range via offsets. You may select and set up a printer. Choose the character set for printing and accept or change the suggested font size. The recommended font size is calculated as follows: print resolution (e.g. 720 dpi) / 6 (e.g. = 120). If desired you may enter a comment which will be printed at the end.
In case you need more flexibility with printing, you can define a block and copy it using "Edit->Copy->Editor Display" as a hex-editor-formatted text into the clipboard. You may paste it in your favorite word processor. It should look perfect in "Courier New", 10 pt.

Properties: Allows you edit the size, the time stamp and attributes of a file or a directory in your own Windows system. Changeable attributes are: A (to be archived), S (system), H (hidden), R (read-only), X (not to be indexed), T (temporary), ~ (sparse). After entering new values in any area (size, timestamps or attributes), simply press the **ENTER** button to apply them. Click the button with the ellipsis to select a new file, or enter path and name directly into the edit box next to that button and press the **ENTER** key. The latter will also work for a directory.
Please note that setting or removing the sparse attribute does not necessarily change the allocation status of already assigned clusters, but will definitely have an effect on newly assigned clusters when you expand the file by setting a larger file size in the same dialog window.

Open Directory: Opens a window that represents a directory on your own computer and allows you to see all its files and subdirectories.

Open Files: This command is used open several files that meet special requirements at a time.

Select a folder in which to open files. Subfolders are browsed optionally. You may specify a series of file masks (like "w*.exe;x*.dll"). There is also a switch that permits opening only those files that contain a certain text or certain hex values. The standard search dialogs are displayed upon request for this purpose. If WinHex is not set up to work as in read-only or in-place edit mode (this can be done in the Options menu), you may choose an edit mode.

Save Modified Files: All files which have been changed are written to the disk.

Save All Files: All files that have not been opened in view mode are written to the disk.

Exit: Use this command to end WinHex. You will be prompted to save any modifications to files and disks.

4.5 Edit Menu

Undo: Reverses the last modification, in case the corresponding undo option was activated.

Cut: Removes the current block from the file and puts it into the clipboard. The data following the block is pulled to the former block beginning.

Copy Block/All/Sector:

- **Normally:** Copies the current block/the entire file/the current sector into the clipboard. The contents of the clipboard can be pasted or written later.
- **As Unicode/ANSI**: Specifically copies text from the text column as UTF-16 Unicode even when the text column is not displayed in Unicode, or specifically as ANSI-encoded text even when the text column is not displayed as ANSI ASCII.
- **Into New File:** Copies the data directly into a new file (not via the clipboard). For instance, this command can be used to recover a lost file from disk sectors.
- **Hex Values:** Copies the data as concatenated hex values.
- **Editor Display:** Copies the data as text, formatted as if it was displayed in the hex editor, i.e. with an offset, a hex and a text column.
- **GREP Hex:** Copies the data as hex values in GREP syntax.
- **C/Pascal Source:** Copies the data as C/Pascal-formatted source code into the clipboard.

Paste Clipboard: Inserts the clipboard contents at the current position of a file. The file data following this position is moved forward.

Write Clipboard: Copies the clipboard contents to the current file at the current position. The data at this position is overwritten. If the end of the file is encountered, the file size is increased so that the clipboard contents finds place.

Paste Clipboard Into New File: Creates a new file of the clipboard contents.

Empty Clipboard: This command is used to free the memory used by the clipboard.

Remove: Deletes the current block from the file. The data following the block is pulled to the

former block beginning. The clipboard is not affected by this command. If the block is equally defined in all open files (i.e. it begins and ends at the same offsets), this command can even be applied to all open files at the same time.

Paste Zero Bytes: Use this command to insert zero bytes at the current position of a file.

Add Block as Virtual File: (forensic license only) If you manually define a block in Volume/Partition/Disk/File mode, this command allows you to add it to the volume snapshot as a carved file, or (in case of File mode) as a child object of the original file. Useful if you wish to treat data in a certain area (e. g. HTML code or e-mail messages found floating around in free space) as a file, e.g. to view it, search it specifically, comment on it, add it to a report, etc. If you manually carve a file within another file in File mode, the resulting file will be marked in the Attr. column as an excerpt and can be filtered as such. Already carved areas in host files are highlighted in File mode. Useful to remind the user whether he or she already has created excerpts from a file and where (e.g. from a large free space virtual file) when continuing to look at that host file.

Define Block: This function is accessible from the menu and the status bar. A dialog box lets you specify the desired block boundaries or size. This command can also be applied to all open files.

Select All: Defines the beginning and the end of the current file as its block limits.

Superimpose Sectors: see below

Convert: cf. Conversions

Modify Data: see below

Fill Block/File/Disk Sectors: see below (Wiping and Initializing)

4.6 Search Menu

Simultaneous Search: see above

Indexing, Search in Index: see above

Optimize Index: see above

Export Word List: Available once an index has been created. Allows to save a list of all the word in the index to a text file. In that list, each word that occurs in the files that were indexed will be present, and only contained once. Useful for a customized dictionary attack.

Find Text: This command is used to search for a specified string of up to 100 ASCII characters in the current file, disk or RAM section (cf. Search Options). Only supports those Unicode characters that are in the 0x00...0xFF range. For a more powerful search variant try Simultaneous Search.

Find Hex Values: This command is used to search for a sequence of up to 100 two-character hex values (cf. Search Options).

Replace Text: Use this command to replace occurrences of a specified string with another string (each of up to 100 ASCII characters), cf. Replace Options. Only supports those Unicode characters that are in the 0x00...0xFF range.

Replace Hex Values: Functions exactly as the Replace Text command, but is applied to a sequence of hex values (100 at most), cf. Replace Options.

Combined Search: Provides a complex search mechanism. In the current and in a second file a common offset is searched, where each file contains the specified respective hex values.

Integer Value: Enter an integer (within the limits of the signed 64-bit integer data type). This function searches data in the current file, which can be interpreted as this integer.

Floating-Point Value: Enter a floating-point number (e.g. $12.34 = 0.1234 * 10^2 = 0.1234E2$) and select a floating-point data type. This function searches data in the current file, which can be interpreted as this floating-point value.

Text Passages: Use this command to look for a sequence of letters (a-z, A-Z), digits (0-9) and/or punctuation marks. It is useful for instance if you intend to translate text passages hidden somewhere in a file with executable code.
Set the sensitivity of the search by specifying how long a character sequence must be to be recognized. Click “Tolerate Unicode characters” in order to force the algorithm to accept zero bytes between two characters.

Continue Global Search: This command is used to continue a global search operation (i.e. a search operation applied to all opened files) in the next file.

Continue Search: Lets you continue a search operation in the current file at the current position.

4.7 Navigation Menu

Go To Offset: Moves the current position to the specified offset. Normally this is done relative to the beginning of the file (offset 0). You can also move the cursor relative to the current position (forward or backward) or from the end of the file (backward). An offset can be specified in bytes (default), words (2 bytes), doublewords (4 bytes), records (if defined), or sectors. Press **F11** to repeat the last position movement.

Go To Page/Sector: Browses to the specified page, sector, or cluster. Note that the data area on FAT drives starts with cluster #2. The Go To Sector dialog, when applied to a physical disk, optionally allows to jump to the designated sector within the respective partition window, so that you can immediately see the allocation status of the corresponding cluster. Only for ordinary partitions, not Windows dynamic volumes or LVM2 volumes.

Go To FAT Entry/FILE Record: Jump to a certain entry in the file allocation table on a FAT drive or to a certain FILE record in the master file table on an NTFS drive, respectively.

Move Block: Moves the current block *selection* (not the data within the block) forward or backward. Specify the distance in bytes. Press **ALT+F11** to repeat the last block movement, press **SHIFT+ALT+F11** to reverse the movement. This command may facilitate editing a file that consists of homogeneous records of a fixed length.

WinHex and X-Ways Forensics keep a history of your offset jumps within a file or disk and allow to go **back** and **forward** in the chain later. Forensic license only: With Back and Forward you can also conveniently go back to a certain directory browser setting. This takes into account: explored path, recursive or non-recursive, sort criteria, on/off state of all filters, settings of some of the filters, some directory browser options. The Back and Forward commands also allow to activate the previously active data window again when switching between windows.

Go To...

Beginning Of File: Display the first page of the current file and moves the current position to offset 0.

End Of File: Displays the last page of the current file and moves the current position to the last byte (offset = file size - 1).

Beginning Of Block: Moves the current position to the beginning of the current block.

End Of Block: Moves the current position to the end of the current block.

Mark Position: Marks the current position and thus enables you to find it again later.

Delete Marker: Removes the marker from the screen.

Go To Marker: Moves the current position to the marker set by Mark Position.

Position Manager: see below

4.8 View Menu

Text Display Only: Hides the hex column and uses the full width of the editor window for the text display.

Hex Display Only: Hides the text column and uses the full width of the editor window for the hexadecimal data display.

Character Set: Select a character set or code page for the text display. You may also use **SHIFT+F7** to toggle the active character set/code page. The default setting is ANSI ASCII. It

uses the most efficient and uncomplicated display method, invoking only the most simple Windows API functions, and it seems to always show character interpretations according to code page 1252, even if regional settings in Windows are different, if in the font selection dialog (accessible via General Options) the "Western" script is selected.

To better utilize widescreen monitors and to assist examiners in particular in Asia, who may encounter text encoded in many different character sets and code pages in the same case, it is possible to see multiple text interpretations of binary data in the hex editor's text diplay at the same time depending on the license type. This is also useful to walk through the raw data of Outlook PST files that use cipher coding, to be able to read encoded ANSI text, encoded Unicode text, and totally unencoded text at the same time.

Personal license for WinHex: no more than 1 character set at a time
Professional license for WinHex: up to 2 character sets at a time
Specialist license for WinHex, X-Ways Investigator: up to 3 character sets at a time
WinHex Lab Edition: up to 4 character sets at a time
X-Ways Forensics: up to 5 character sets at a time

You have the option to interpret data as misaligned text in UTF-16 LE as well as misaligned UTF-16 BE. Misaligned means starting at odd offsets. That makes a difference in non Western European languages and renders text stored in that fashion actually readable. It is more common to store UTF-16 text aligned at even offsets, but you cannot count on it. File formats and memory allocations of some applications do not mind alignment and may even use both aligned and unaligned UTF-16 text at the same time.

Please note that any text *input* from the keyboard (when not in read-only mode) is interpreted as being based on the ANSI code page that is active in Windows, except if the primary text column is set to the IBM/OEM/DOS code page 850 (Latin I), in which case input is based on that code page.

Record Presentation: When editing subsequent data records of the same size (for instance, table entries of a database) you may now have WinHex display every other record with a different background color, as a kind of visual aid. The color can be selected in the General Options dialog. Also, WinHex offers to display the current record number and the offset within that record (relative offset) in the status bar, based the record size and the offset of the first record as specified.
If any of the two record features is enabled, the Go To Offset command allows moving the current position in units of the current record size. If relative offsets are enabled, the Page Dn/Up keys move the cursor in units of the record size, except if you hold the Ctrl key.

Show: The **Case Data** window is part of the forensic user interface of WinHex/X-Ways Forensics and required for working with a case (when hiding the window, the case is closed). The **directory browser** is available for logical drives/partitions opened with the disk editor. The **Data Interpreter** is a small window that provides "translation services" for the data at the current cursor position. The **toolbar** is displayed optionally, too. A **tab control** makes each edit window accessible with a single mouse click only. The **info pane** provides in-depth information on any open object (file, disk, RAM).

Template Manager

Tables: Provides four conversion tables (cf. ANSI ASCII/IBM ASCII).

Lines & Columns

Synchronize Scrolling: Synchronizes up to four tiled windows on identical absolute offsets. Hold the Shift key when enabling this feature to tile the windows horizontally instead of vertically.

Synchronize & Compare: Synchronizes up to four windows and visually displays byte value differences. If no more than two windows are involved, WinHex maintains the initial distance between the offsets of the first shown byte in these windows when scrolling. Not synchronizing on absolute offsets is useful for example when comparing two copies of the file allocation table, which are obviously at different offsets. You may jump to the next or to the previous byte value difference by clicking the extra arrow buttons that are provided in one of the two edit windows.

Refresh View: Redraws the contents of the current edit window. In case the current file was updated by an external program, WinHex offers to dismiss any changes made in WinHex and reload the file from scratch.
Also refills the directory browser if the directory browser has the input focus. Useful for example when a filter for tagged items is active and you remove the tag marks of some of the listed files, if you wish to update the listing in the directory browser and get rid of those files that are no longer tagged.

4.9 Tools Menu

Open Disk: See chapter “Disk Editor”.

Clone Disk: See chapter “Disk Cloning”.

Explore recursively: Changes into a recursive view for the directory that is currently listed in the directory browser or back to the normal view. A recursive view means that not only files will be listed that are contained directly in the current directory, but also all files in all subdirectories of that directory and their subdirectories etc. For example, this allows to copy/recover selected files from different paths in a single step.

File Recovery by Type: See below.

Take New Volume Snapshot: Available for partitions with one of the supported file systems. WinHex traverses all cluster chains and thereby generates a drive map. This enables WinHex to fill the directory browser and to display for each sector which file or directory it is allocated to. It is recommended to invoke this command again after file operations on a drive to keep the information displayed by WinHex up to date. Cf. Security options.

Initialize Free Space: Confidential information is possibly stored in currently unused parts of a drive as a result of normal delete, copy and save actions. Free space on a drive can be initialized for security reasons. This effectively overwrites all data in unused parts of the disk and makes it impossible to recover this data. Available for partitions opened as drive letters. *Available in WinHex only, not in X-Ways Forensics.*

Initialize Slack Space: Overwrites slack space (the unused bytes in the respective last clusters of all cluster chains, beyond the actual end of a file) with zero bytes. This may be used in addition to "Initialize Free Space" to securely wipe confidential data on a drive or to minimize the space a compressed disk backup (like a WinHex backup) requires. Close any running or resident program that may write to the disk prior to using this command. *Available in WinHex only, not in X-Ways Forensics.*

Initialize MFT Records: On NTFS volumes, WinHex can clear all currently unused $MFT (Master File Table) FILE records, which may contain metadata (e.g. names) and even contents of previously existing files. *Available in WinHex only, not in X-Ways Forensics.*

Initialize Directory Entries: On FAT volumes, WinHex can clear all currently unused directory entries, to thoroughly remove traces of previously existing files or earlier names/locations of existing files from the file system. Useful especially in conjunction with the function to initialize all free space. *Available in WinHex only, not in X-Ways Forensics.*

Scan For Lost Partitions: Formerly existing hard disk partitions that were not automatically found when opening a physical hard disk (or an image of a physical hard disk) may be found and properly identified with this command. This command searches for the signature of master boot records, partition table sectors, FAT and NTFS boot sectors via the 0x55 0xAA signature plus for Ext2/Ext3/Ext4 superblocks, optionally only from the first sector that follows the last (location-wise) partition that was already found, and lists newly found partitions in the directory browser. Works with sector size 512 bytes only.

Interpret as Partition Start: When you find the start sector of a volume (e.g. lost partition) on a physical disk, this menu command allows you to make such a partition easily accessible via the Access button menu. If no known file system is detected starting at the currently displayed sector, you will be asked for the number of sectors that you wish to include in the newly defined partition.

Set Disk Parameters: Using this command on a physical disk, you may override the total number of sectors or optionally (can be left blank) the number of cylinders, heads, and sectors per track (all practically meaningless nowadays). This might be useful to access surplus sectors at the end of the disk (in case the total number of accessible sectors was not detected correctly), or to adjust the CHS coordinate system to your needs. Alternatively, you have the option to change the detected sector size of a physical hard disk or image, as used internally in the program for various navigation and computation work. If you should adjust the sector size, the sector count is adjusted accordingly. For example, if you change the detected sector size from 512 bytes to 4 KB (i.e. you multiply it by 8), then the total number of sectors is automatically divided by 8 to keep the same total detected disk capacity (assuming the capacity was detected correctly).

Open Memory: See chapter “Memory Editor”.

Capture Processes: Allows to acquire all data in the memory of running processes on a live system contiguously (i.e. pages in the order as allocated by the process). The creation times of processes can be seen as the creation timestamps of the memory dumps. Pages marked as containing executable code (PAGE_EXECUTE* styles) are optional and if omitted will suitably reduce the amount of data if you are merely interested in keyword searches or carving and not malware analysis. Carving in the memory dumps (files shown as type "mem") can be performed by uncovering embedded data, one of the functions of volume snapshot refinement. The output folder of "Capture Processes" is by default either a subdirectory of the case or - if no case is active - a subdirectory of the directory for images. It can be automatically explored in Windows File Explorer once the output is complete and/or added to the active case as a directory. The memory dumped by "Capture Processes" can also be useful on your own system if an application in which you type text (e.g. an e-mail client) suddenly freezes and you want to recover what you wrote.

This command can also produce a tab-delimited list of all top-level windows with their titles and corresponding processes plus (comma-delimited) the titles of their child windows. Screenshots of some of the top-level windows are taken and output automatically. If this functionality is used without administrator rights, only processes of the current user are covered, otherwise all processes. A filter is available for process dumping. You can use it like other file mask filters in X-Ways Forensics. For example "explorer.exe" will only dump memory and windows of the Windows File Explorer process. ":C*" will dump all processes except those whose names starts with the letter "C", i.e. for example not "Chrome.exe". The file mask is not case sensitive. Multiple file masks can be concatenated with semicolons. (However, the total length is limited.)

View: Available only with a forensic license. Invokes the internal viewer.

External Viewer: Invokes external file viewing programs such as Quick View Plus etc., as selected in the Options menu, and opens the current file.

Invoke X-Ways Trace: Available only if X-Ways Trace is installed. This software can analyze the history/cache files of various Internet browsers.

Calculator: Runs the Windows calculator “calc.exe”. Switching to scientific mode is highly recommended.

Hex Converter: Enables you to convert hexadecimal numbers into decimal numbers and vice versa. Simply type in the number and press **ENTER**.

Tables: Provides four conversion tables (cf. ANSI-/IBM-ASCII).

Compare: This command is used to compare two data windows (files or disks) byte by byte. Decide whether different or identical bytes shall be reported. You may specify how many bytes to compare. If desired, the operation can abort automatically after having found a certain number of differences or identical bytes. The report can be stored as a text file, whose size might otherwise grow dramatically. The comparison starts at the respective offsets specified for each

edit window. These offsets may differ, such that e.g. the byte at offset 0 in file A is compared to the byte at offset 32 in file B, the byte at offset 1 with the one at offset 33, etc. When you select an edit window for comparison, the current position will automatically be entered in the "From offset" box.

In X-Ways Forensics there is also an option to output identified different or identical data areas as search hits (1 entry per matching area) instead of a text file (1 line per matching byte), for convenient review and navigation right within the program in the search hit list, similar to block hash matches. This option is only available if at least the 2nd data source is an evidence object. The result can be seen in the search hit list of that evidence object. Useful for example for users who wish to compare cloned disks with minor changes, if they have different hashes or one of them has been used a little more, to actually locate the differences and better understand what has caused them. Useful also to compare component disks of a hardware RAID level 0 system or a mirrored volumes, to check whether they are really absolutely identical, and if not to easily find the areas that differ, see how large they are, what kind of data these areas contain, and assess whether the second copy requires full treatment itself including carving, keyword searches etc.

There is another compare function: You may compare edit windows visually and synchronize scrolling in these windows, with the Synchronize and Compare command (View menu).

Analyze Block/File: Scans the data within the current block/the entire file and counts the occurrences of each byte value (0...255). The result is graphically displayed by proportional vertical lines. The number of occurrences and the percentage are displayed for each byte value when moving the mouse over the corresponding vertical line.
Use this command for instance to identify data of unknown type. Audio data, compressed data, executable code etc. produce characteristic graphics. Use the context menu of the window to switch zero byte consideration on or off, to print the analysis window, or to export the analysis to a text file.
When analyzing small amounts of data (<50,000 bytes), the compression ratio that zlib achieves for that data is displayed in the analysis window caption, which also allows to draw conclusions about the nature of the data.

Compute Hash: Calculates a checksum/hash value of the data in the the current data window (e.g. entire current file, storage device, partition) or the currently selected block. You can choose from 8-bit, 16-bit, 32-bit, 64-bit checksums, CRC16, CRC32, MD5, SHA-1, SHA-256, SHA-512, …

4.10 File Tools

Concatenate: Select several source files that are to be copied into one destination file. The source files are not affected.

Split: This command creates several destination files using the contents of a single source file. Specify a split offset for each destination file. The source file is not affected by this function.

Unify: Select two source files and one destination file. The bytes/words from the source files will

be written alternately into the destination file. The first byte/word originates from the source file that was specified first. Use this function to create a file with odd and even bytes/words originating from separate files (e.g. in EPROM programming).

Dissect: Select a source file and two destination files. The bytes/words from the source files will be written alternately into the destination files. The first byte/word will be transfered to the destination file that was specified first. Use this function to create two separate files each containing either the odd or the even bytes/words of the original file (e.g. in EPROM programming).

Create Hard Link: Function to create hard links of files in NTFS volumes. Useful for example to play around with hard links when attending NTFS file systems training, or if you would like to add the same image to the same case again, which is only possible under a different name, or if you would like to create a hard link to xwforensics.exe named WinHex.exe, in order to run X-Ways Forensics as WinHex. First you select the existing file, then a path and name for the additional hard link.

Copy Sparse: Can copy a selected file and preserves the sparse nature if it is an NTFS sparse file, in the destination file. That means for example when copying a 1 TB skeleton disk image that only has 100 MB of data allocated, the copy process will finish almost instantly because only 100 MB out of 1 TB of data have to be copied. Conventional copy functions do not preserve the sparse nature of a file and copy the amount of data as indicated by the nominal file size, even if most of the data is internally unallocated and read virtually as binary zeroes.

Replicate Directory: Copies a directory with all its files and subdirectories, recursively, and recreates individually NTFS-compressed source files as NTFS-compressed in the respective output folder if supported by the destination file system and any layer in between. The command does not retroactively compress such files after their creation, but writes them immediately as compressed, which is more efficient. However, it still has to copy/send the decompressed amount of data of the source file. Supports overlong paths. Select the source directory first, then specify/create the destination directory. This function is useful for example if you wish to copy or move a case directory, which contains a few NTFS-compressed files that would be inefficient to store as uncompressed. Note that alternatively you can open a case and use the Save As command in the Case Data window for the same effect.

Get Full Access: Take control of all the files in a directory in the currently used Windows system and give all users full access, recursively.

Wipe Securely: This command is used to erase/overwrite the contents of one or more files irrevocably. Optionally, the selected files are also shortened to a length of zero, they are deleted in the file system and their name entries are also purged. Therefore this command should be applied to files with confidential contents that are to be destroyed. *Available in WinHex only, not in X-Ways Forensics.*

Delete Recursively: This command can be used to recursively delete a directory with all its subdirectories if they cannot be deleted with Windows Explorer or other Windows tools and commands because of illegal characters in the directory names or filenames or because of

missing rights (for example if "Trusted Installer" is the owner) if you can get those rights (if you are running WinHex with administrator rights, and you may be prompted for your consent to use administrator power to take ownership of the selected directory structure prior to deletion). Note that you cannot apply this command to such a problematic directory itself, only to a parent directory.

4.11 Specialist Menu

Specialist and forensic licenses only.

Refine Volume Snapshot: see separate chapter

Technical Details Report: Shows information about the currently active disk or file and lets you copy it e.g. into a report you are writing. Most extensive on physical hard disks, where details for each partition and even unallocated gaps between existing partitions are pointed out. Under Windows XP, WinHex also reports the password protection status of ATA disks.
Forensic license only: X-Ways Forensics tries to detect hidden host-protected areas (HPAs, a.k.a. ATA-protected areas) and device configuration overlays (DCO areas) on ATA/SATA hard disks. A message box with a warning will be displayed in case the disk size has been artificially reduced. The real total number of sectors according to ATA, if it can be determined, is listed in the details report. Some important SMART status information is also displayed, for hard disks connected via [S]ATA that support SMART. Useful to check for one's own hard disk as well as that of suspects. For example, you can learn how often and how long the hard disk was used and whether it has had any bad sectors (in the sense that unreliable sectors were replaced internally with spare sectors). If a hard disk is returned to a suspect and he or she consequently complains about bad sectors and accuses you of having damaged the disk, a details report created when the hard disk was initially captured can now show whether it was already in a bad shape at that time. Also, seeing that spare sectors are in use means knowing that there is additional data to gain from the hard disk (with the appropriate technical means).

The following metadata about BitLocker and BitLocker To Go volumes is output: Volume creation timestamp, textual volume description, encryption method, protection type, and volume master key last modification timestamps. BitLocker-related timestamps are also output to the event ist.

The Technical Details Report also checks for certain read inconsistencies that can occur with flash media (for example USB stick of certain brands/models, but not others) in data areas that have never been written/used, where the data is undefined. The data that is read in such areas, for example when imaging the media, may depend on the amount of data that is read at a time with a single internal read command. The result is mentioned in the report. If inconsistencies are detected (“Inconsistent read results!” in the report), you will see a message box, which offers to read sectors in smaller chunks from that device as long as it is open, which likely yields the expected zero value bytes instead of some random looking non-zero pattern data when reading such areas. Use of this option does not give you data that is somehow more accurate or original (undefined is undefined and does not mean zeroed out) or contains more or less evidence, it can just have a big impact on compression ratio achieved and reproducibility of hash values with

other tools, which may use different chunk sizes for reading and thus produce different data and hash values. Note that it is possible that read inconsistencies occur that are not detected by X-Ways Forensics, because a complete check would be very slow. Again, these inconsistencies are not fatal and not the fault of the software, and they can be explained. Note that the Technical Details Report is routinely created already when you start disk imaging with the File | Create Disk Image command, so you do not need to invoke the report yourself prior to imaging.

There is an option to show a byte-swapped version of a hard disk serial number in addition to the serial number reported through the operating system, when in doubt. Some users of certain interfering hardware write blockers may find that useful.

Interpret Image File As Disk: see separate chapter

Mount as Drive Letter: see separate chapter

OS-Wide Write Protection: Allows to write-protect locally attached physical storage devices (including removable media, but not optical media) with all their volumes everywhere in the operating system, in all applications, even at the sector level in WinHex itself, no matter which edit mode is active. This can be useful to protect original disks that need to be acquired or analyzed (but only after Windows has detected and accessed them) and your own disks that contain images, from accidental alteration, deletion, or data corruption. The effect will last until you remove the write protection again or unplug the devices or reboot your computer. To keep Windows from touching newly attached physical storage devices before you can write-protect them (i.e. to keep them in "offline" mode first), you would need to disable automatic mounting in Windows (and verify that this works). Turning on write-protection for an offline disk will automatically bring the disk online, at the same time while rendering it read-only. Careful, do not write-protect disks that your Windows system needs to write to for proper functioning!

This command also allows to selectively write-protect only specific volumes (if mounted as drive letters) on GPT-partitioned disks, not the entire physical storage device. Please note that the read-only status of a volume cannot be lifted selectively if the entire underlying physical storage device is read only. When write-protecting a single volume on an MBR-partitioned disk fails, you will be offered to target all volumes on that storage device instead.

Reconstruct RAID System: see separate chapter

Gather Free Space: Traverses the currently open logical drive and gathers all unused clusters in a destination file you specify. Useful to examine data fragments from previously existing files that have not been deleted securely. Does not alter the source drive in any way. The destination file must reside on another drive.

Gather Slack Space: Collects slack space (the unused bytes in the respective last clusters of all cluster chains, beyond the actual end of a file) in a destination file. Otherwise similar to Gather Free Space. WinHex cannot access slack space of files that are compressed or encrypted at the file system level.

Gather Inter-Partition Space: Captures all space on a physical hard disk that does not belong to

any partition in a destination file, for quick inspection to find out if something is hidden there or left from a prior partitioning.

Gather Text: Recognizes text according to the parameters you specify and captures all occurrences from a file, a disk, or a memory range in a file. This kind of filter is useful to considerably reduce the amount of data to handle e.g. if a computer forensics specialist is looking for leads in the form of text, such as e-mail messages, documents, etc. The target file can easily be split at a user-defined size. This function can also be applied to a file with collected slack space or free space, or to damaged files in a proprietary format than can no longer be opened by their native applications, like MS Word, to recover at least unformatted text.

Evidence File Container: see above

External Virus Check: (Forensic license only.) Sends all files or all tagged files in an evidence object's volume snapshot to an external virus scanner, optionally only files with a size below a certain threshold. Files that are locked, deleted, or renamed by the virus scanner in the output directory will be labeled “Virus suspected”. It is the responsibility of the user to verify that a virus scanner is active, that it watches the folder for temporary files, and that it will indeed lock, delete or rename infected files. After verifying whether the file has been locked, deleted, or renamed externally, X-Ways Forensics deletes it itself if it still exists.

Bates-Number Files: Bates-numbers all the files within a given folder and its subfolders for discovery or evidentiary use. A constant prefix (up to 13 characters long) and a unique serial number are inserted between the filename and the extension in a way attorneys traditionally label paper documents for later accurate identification and reference.

Trusted Download: Solves a potential security problem. When transferring unclassified material from a classified hard disk drive to unclassified media, you need to be certain that it will have no extraneous information in any cluster or sector "overhang" spuriously copied along with the actual file, since this slack space may still contain classified material from a time when it was allocated to a different file. This command copies file in their current size, and no byte more. It does not copy entire sectors or clusters, as conventional copy commands do. Multiple files in the same folder can be copied at the same time.

4.12 Options Menu

General Options: see below

Viewer Programs: see below

Undo Options: see below

Security Options: see below

Data Interpreter Options: cf. Data Interpreter

Edit Mode: Allows you to select the edit mode used in Winhex globally. (The info pane's context menu allows to select the edit mode specifically for the active edit window only.)

4.13 Window Menu

Window Manager: Displays all data windows and provides “instant window switching” functionality. You may also close data windows and save changes.

Save Arrangement As Project: Writes the current window constellation (open case, open data windows, position of the data windows on the screen, cursor position in data windows, block selection, ...) into a project file. From the Start Center you will then be able to load the project and restore editing positions in each document at any time, to conveniently continue your work right where you left it or to begin your work in case of a recurring task.

Close All: Closes all data windows and thus all open files, disks and RAM sections. If you have edited any data, you will be prompted for each and every data window if there are unsaved changes to the data in it, so you can decide whether to save or to discard those.

Close All Without Prompting: Closes all data windows and thus all opened files and disks without giving you the opportunity to save any changes to the data in all those windows, without prompting you for each and every data window that has changes. As this is a potentially dangerous command (you may lose a lot of work if you have edited data in many data windows), there will be warning and you can still abort. You know that a window will be shown first that expects additional confirmation because of the ellipsis at the end of the command name, as is the convention.

Cascade/Tile: Arranges the data windows in the aforementioned way.

Minimize All: Minimizes all data windows.

Arrange Icons: This command neatly arranges all minimized data windows within the main window.

4.14 Help Menu

Contents: Displays the contents of the program help.

Setup: Allows you change the language of the user interface. When you set English as the language, you have the option to adjust American spelling to British spelling in most parts of the user interface (more after restarting). British spelling is also the default setting in a new installation if the Windows installation language is UK English, Australian English or New Zealand English.

With **Initialize** you can restore the default settings of the program.

Uninstall: Use this command to remove the running program from your system. This works even if you did not install it using the setup program.

UI Text Adjustments: You can rename many directory browser columns to your liking, for example in order to keep continuity in the user interface between earlier and future versions, or for compatibility in data transfers (e.g. Export List command), or because a certain column title has not been translated to your preferred Latin-based user interface language and you would like to see your own translation of the English title, or because you prefer to see "Attributes" instead of the abbreviation "Attr.", etc. In the dialog window with the directory browser options you can simply right-click a column title for that, and will then be given the opportunity to replace the title with your own wording.

Many more text fragments (strings) in the user interface are customizable, through this menu command in the Help menu. You would need to identify the exact standard text fragment to replace and provide your own version of it. If the text that you are looking for is not found and you don't know exactly how it is stored internally, you can search for it in the file "language.dat". Your customizations are stored in the file "UI Text Adjustments.txt" and can be shared with other users. The file can presumably be used in future versions as well, as long as the original text fragments remain the same. It simply consists of one adjustment per line, with the original text first and the replacement second, delimited by a tab character (meaning those few original texts that already contain a tab character cannot be adjusted). You may also edit that file manually. Please note that the translations of non-Latin languages are available as simple text files and can thus be changed in those files much more directly.

Online: Opens in your browser, if you have an Internet connection, the X-Ways web site, the support forum, the newsletter subscription page, and a page where you can check your license status, retrieve the latest download links and get upgrade offers. There is also an option to check for updates online occasionally upon start-up of the software or at any time when you like. This can report the availability of later versions or new service releases of the currently used version (not pre-release versions) and allow to start the download. Does not send any data from within the program to the Internet, for example no system or user information or dongle ID, neither directly nor encrypted nor anonymized, of course no case data, not even the currently used version number, nothing. This option is active by default only if the program determines that it is running on the user's own system (if it is executed from the C: drive or if it was installed using the setup program). The check does not occur when running the program for the first time, so that you definitely have a chance to turn off this option before anything happens. Given the fact that most systems on which X-Ways Investigator and X-Ways Forensics are run do not have an Internet connection, this option has a limited effect.

Click on the version number on the far right of the menu bar: Displays information about the software, such as the program version, unlock status, how much free space is available to it on the drive for temporary files and image files, whether the program is running with administrator rights, whether the MS Visual C++ 2013 Redistributable Package (for the latest version of the viewer component and Dokan) is installed and if not whether at least the MS Visual C++ 2005 Package is installed (for v8.5.2 of the viewer component and older). Some of this information can be important when running X-Ways Forensics on a live system, i.e. a system that is not your own

and that you wish to examine.

4.15 Windows Context Menu

The Windows shell displays the context menu when the user clicks an object with the right mouse button. WinHex is present in the context menu only if you enable to corresponding option (see “General Options”).

Edit with WinHex: Opens the selected file in WinHex.

Open in WinHex: Lets you open all files of the selected folder in WinHex, just like the Open Folder command of the File menu.

Edit Disk: Opens the selected disk in the disk editor of WinHex. If you hold the **SHIFT** key, instead of the selected logical drive the corresponding physical disk is opened, if any.

WinHex provides its own context menus on the status bar, the Data Interpreter, and in the Position Manager.

5 Forensic Features

5.1 Interpret Image File As Disk

This command in the Specialist menu treats a currently open and active disk image file as either a logical volume (potentially with a supported file system) or physical (potentially partitioned) disk. This is useful if you wish to closely examine the file system structure of a disk image, extract files, etc. without assistance from any operating system. If interpreted as a physical disk, WinHex can access and open the partitions contained in the image individually as known from "real" physical hard disks. The same functionality is also used internally when adding images to a case in X-Ways Forensics and re-opening them later, or when using the File | Open command and you already tell the application in advance that the file that you are about to open is an image file.

It is also possible to interpret spanned raw image files, that is, image files that consist of separate segments of any size. For WinHex to detect a spanned image file, the are a few possibilities supported for naming:
1) The first segment may have an arbitrary non-numeric filename extension (e.g. .dd or .img), and then the second segments must be named .002, the third segment .003, and so on.
2) The first segment may have one of these numeric filename extensions: .001, .0001, .00001, .000, .0000 or .00000. The following segments must directly continue with incrementing numbers and the exact same number of digits, i.e. either three, four or five.
3) abc.bin, abc_1.bin, abc_2.bin, ...
For 1) and 2) all segments must have the same base filename (the part of the name before the

extension). The Create Disk Image command can image disks and produce canonically named file segments. Image segmentation is useful because the maximum file size supported in FAT32 file systems or on media such as DVD is considerably limited. It might also help in risk reduction (the smaller the segments, the less catastrophic the amount of lost data if a file is lost due to file system errors) and might have a performance benefit (if the operating system more effectively buffers frequently required image data if stored in smaller segments).

In some rare cases WinHex may be unable to correctly determine the nature of the image, i.e. whether it is an image of a physical disk or of a volume, consequently interprets the data in the image in a wrong way. If so, hold the Shift key when invoking this command. That way WinHex will ask you and not decide on its own. That will also make WinHex prompt you for the correct sector size and in the case of raw images for an additional storage location of further image file segments (in case you had to spread them across two different drives). Should there be any problems with detecting the file system in a volume, you may hold the Shift key when opening the volume to indicate the file system type you suppose in the volume.

Mode 1 and Mode 2 Form 1 ISO CD images with 2,352 bytes per sector are also supported, if they are not spanned, and (with a forensic license) also main memory dumps. Also VMware's Virtual Machine Disk images (VMDK) can be interpreted and dynamic Virtual PC images (VHD, VHDX) and Virtual Box disk images (VDI) of the default subtype "sparse" and the subtypes "fixed size" and "diff" (snapshots). Snapshot images can only be interpreted if the parent is available and open and interpreted itself beforehand. VMDK images with ESXi Host Sparse Extents (also referred to as "Copy-on-Write Disks" or COWD), as used by ESXi servers e.g. for virtual machine snapshots, are not supported. Only allocated areas in virtual machine images can be edited. X-Ways Forensics and X-Ways Investigator can also interpret .e01 evidence files, which can be created with the Create Disk Image command, plus unencrypted Ex01 evidence files.

It is also possible to interpret images of various kinds (raw images and most VHD/VHDX/VMDK/VDI as well as backup bundles created by Apple Time Machine via the file “com.apple.TimeMachine.MachineID.plist”) and nature (disk/volume) even if they are stored within other images (forensic disk images created by yourself), without copying them off the outer image first, as long as they do not consist of multiple segments. That can save a considerable amount of time, especially if after interpreting the contained image you can quickly see that it is not really relevant, and of course also drive space. First right-click the image in the directory browser and open it with the context menu's Open command in a separate data window. After that, interpret the image using the command in the main menu. And then, once the volume snapshot has been taken, if you think that the image is relevant, you can add it to the active case as usually with the "Add to active case" command in context menu of the data window's tab or with the Add command in the Case Data window's File menu. Image files within TAR archive should also work, which is handy for VMDK virtual machine disks within OVA files (open virtualization archives in TAR format).

Loose $MFT files can be directly and conveniently interpreted as if they were images of NTFS volumes, to get at least a full listing of all files and directories, with their paths, timestamps and attributes. It's possible to open resident files (files whose contents is small enough to fit into the FILE records), but no other files, of course. Useful if in special situations all you have is the $MFT, not the entire volume.

5.2 Case Management

The integrated computer forensics environment with a case and evidence objects can only be used with a license for X-Ways Forensics or X-Ways Investigator. It supports multiple examiners per case, automated log and report file generation, and much other functionality that is not available otherwise. In order to work with a case, make sure the "Case Data" window is visible on the left of the main window. If not, enable View | Show | Case Data.

From the File menu, you may create a new case (start from scratch), open an existing case, close the active case, save the active case, or automatically generate a case report. You may add storage devices as evidence objects to the case, or images (files that will be interpreted like media), or memory dumps of old Windows versions, or directories or single files including archives on your own computer. Adding a directory instead of a whole partition or disk can be useful if a directory or a file of interest resides on a drive with many irrelevant files, if you merely wish to view, hash, or search a few of those files, check their metadata or copy them to an evidence file container etc.

A case is stored in a .xfc file (xfc stands for X-Ways Forensics Case) and in a subfolder of the same name, just without the .xfc extension. This subfolder and its child folders are created automatically when the case is created. You may select the base folder for your cases in General Options. It is not necessary to explicitly save a case, unless you need to be sure it is saved at a given time. A case is saved automatically at latest when you close it or exit the program. The only exception is when closing the case with the "Close Case (don't save)" command. For example if you have accidentally lost your carefully set tag marks (by untagging all, with a misdirected click in the column header) or if you accidentally lost labels (by pressing Ctrl+0 for all selected files), it is important to invoke that special menu command as soon as possible, before the auto-save interval elapses next time, to avoid that the volume snapshot(s) will be saved. Afterwards you can open the case again, and find everything as it was last time when the case was saved, which means that on average you will only lose half the amount of work that you get done within the auto-save interval, not everything.

In the case properties window, you may name a case according to your own conventions (e.g. title or number). The date and time you create a case is recorded and displayed. The internal case filename is displayed as well. You may enter a description of the case (of arbitrary length) and the examiner's name, the examiner's organization's name and address. You may enable or disable the automated log feature for the whole case. Optionally, the evidence object subfolders in the case folder are always suggested as default output folders for files recovered/copied off a file system. You may wish to disable that feature if your preference is to copy files from various evidence objects into the same output folder.

You may select up to two code pages related to the case (more precisely: related to the locale where the original media related to the case were used). These code pages are used when naming .eml files based on subject lines (.eml files extracted from e-mail archives). If both code pages are identical, that does no harm. If identical to the currently active code page in Windows, they do not have any effect. These code pages are also used to convert the filenames in zip archives to Unicode. The first code page can be defined as an expected code page in Outlook PST files. The second code page can be used as a fallback code page or even expected default code page when

taking a volume snapshot of Ext* file systems to decode filenames and directory names, which could be necessary when dealing with some legacy systems or purpose-built environments where encodings other than the standard UTF-8 were enforced.

Two kinds of proactive filters, based on names and timestamps, can be activated in the properties of a case. Proactive filters allow you to restrict the initial volume snapshot. Files that don't pass these filters will not be included in any volume snapshot that is taken while such filters are active. Directories are still included. This pertains only to partitions/volumes and file archives that are evidence objects, and all the files that are found in them directly, following the defining data structures of the file system or the archive. It does not restrict the addition of files that are found in any other way, for example by a file header signature search or when checking files that are already contained in a volume snapshot for embedded data etc. Proactive filters are special in that they can prevent files from involuntarily getting into a volume snapshot at all, files that you do not need or want to be there or that you are not supposed to see. Either if your task or search scope is limited to specific files whose names or timestamp ranges are known beforehand or if the evidence object (image or file archive) is so big that by avoiding hundreds of millions of other files you save time and main memory or can make the volume digestible for X-Ways Forensics at all (i.e. keep the volume snapshot size within the supported boundaries). The creation of the volume snapshot itself may be noticeably accelerated that way if the evidence object is an image file, plus all subsequent steps (navigating, listing, sorting, filtering, volume snapshot refinement) are less computationally expensive if you proactively prevent the inclusion of large numbers of unwanted files. A count of how many files are proactively omitted during the creation of the volume snapshot is displayed in the progress indicator window. After completion, the total number of such files can always be checked in the status of the volume snapshot in the dialog window for volume snapshot refinement. A warning that a proactive filter is active is output in the Messages window once per session, when a volume snapshot is taken.

There is an option to automatically verify the hash value when adding an image to a case, if such a hash value is present, or (if the checkbox is fully checked) to compute the hash value from scratch if the image doesn't have one. Newly created cases inherit the state of this option from the last case whose settings were defined. This also means that you can verify images from the command line, with the AddImage command. The result will be output 1) in the Messages window, 2) in msglog.txt if desired, and 3) in the properties of the evidence object, i.e. the representation of the image in the case.

Cases files can be password-protected. This does not involve encryption and is just a kind of lock. Without the password, a case can either not be opened at all or can only be opened as read-only (write-protected), meaning changes will not and cannot be saved. If the password is lost by a user, case files saved by X-Ways *Investigator* can be unlocked with a super-user password if such a password had already been entered in the installation used at the time when the case file was saved (details on how to do that are available on request).

When creating a new case, you have the option to make X-Ways Forensics recognize evidence objects that are physical media (not images) by their own intrinsic properties, not by the Windows disk number. Using this option will prevent earlier versions of X-Ways Forensics from opening the case. The advantage is that you may add multiple hard disks or external USB disks or sticks to the case that are attached to the computer at different times and get the same disk

number assigned by Windows. Another advantage is that if the number of the same disk as assigned by Windows changes, X-Ways Forensics will still recognize the disk. Useful especially for triage, when not working with images. Please note that X-Ways Forensics may be unable to recognize external media already known to the case if next time they are attached through a different hardware write blocker. In that situation you can still use the "Replace with new disk" command in the evidence object context menu to point X-Ways Forensics to the correct disk. Note that component disks of an internally reconstructed RAID (read disks, not images) are still remembered by the Windows disk number when re-opening a RAID that you have added to a case.

When clicking the “Passwords...” button, the case's password lists for encrypted general purpose file archives will open in your preferred text editor for editing.

When clicking the “SIDs...” button you can see a collection of all SID/username combinations encountered in that case (gathered from SAM registry hives in all Windows installations on images/media ever added to the case). They are used by X-Ways Forensics to resolve SIDs to usernames when working with that case.

The most powerful concept in X-Ways Forensics, that allows to systematically and completely review files on computer media, is the so-called *refined volume snapshot*. It is possible to refine the standard volume snapshot for all evidence objects of a case in one step, and to search all evidence objects with volume snapshots logically with the help of the virtual global case root window. Note that it is possible to generate a flat overview of all existing and deleted files from all subdirectories on an partition or image file of a partition by recursively exploring the root directory. In order to explore a directory recursively (i.e. list its contents plus the contents of all its subdirectories plus their subdirectories), *right*-click the directory in the directory tree in the Case Data window. In order to *tag* a directory, you can click it with the middle mouse button in the directory tree.

Backups

The command "Back up/Restore" in the Case Data context menu allows you to conveniently make a backup of the selected evidence object's volume snapshot. Backups can be restored at any later time with the same command, and they can also be deleted with the same command (right-click an item in the list of backups to get the Delete command). Such a backup is like a snapshot of the volume snapshot. Useful if you think you might want to revert to a certain processing stage later (i.e. undo changes to the volume snapshot), for example after having carefully tagged thousands files that you don't want to lose, before running a file header signature search with experimental settings that might produce a lot of garbage files, before attaching external files with options that you had never tried before, before running an X-Tension made by a 3rd party, before totally removing excluded items from the volume snapshot etc. Labels, events, and search hits are also included in the backup. Search hits can be restored from a backup only if the search term list of the case did not change in the meantime. Indexes are not included in the backup, but can be manually backed up, of course.

The same command applied at the case level (right-click the case title in bold for that) allows to make a backup of the entire case, covering all evidence objects' volume snapshots, all labels, events, search terms, search hits, indexes, image file paths, etc. etc. Such backups can be restored

from the same dialog window. Such backups can also be opened directly with the Open Case command if necessary, as they are complete copies of a case. (Backup .xfc file are created with the "hidden" attribute, though, as they are meant to be dealt with within X-Ways Forensics only.)

In order to completely *delete* a case or the backup of a case manually, you need to delete its .xfc file and the corresponding directory with the same name and all its subdirectories.

5.3 Multi-User Coordination For Large Cases

X-Ways Forensics and X-Ways Investigator can distinguish between different examiners working with the same case at different times or at the same time and keep their results separate. Multi-user support is especially helpful for large cases. A maximum of 255 users (examiners) is supported per case. Examiners are recognized internally by their Windows user accounts.

Multiple users may open the same evidence objects in the same case simultaneously for examination. By same case we mean the same case file, not a copy, stored in a shared network location or on a terminal server. X-Ways Forensics is responsible for synchronizing labels, comments and additions of files to the volume snapshot, and for making users aware of access conflicts before they occur and preventing them in most situations.

All related options can be found by clicking the button labelled "Multi-user support options" in the case properties dialog window. In particular, when creating the case (and only then), you can choose to make X-Ways Forensics *not* distinguish between different users. That would be useful if you know that only you will process that case and if you wish to process it on different computers where you have Windows accounts with different SIDs, so that you will always be treated as the same user. Also useful if multiple examiners are going to process the same case at *different* times and wish to share all their results directly.

Another multi-user support option coordinates certain kinds of accesses to volume snapshots (related to adding items to the snapshot as well as editing comments and metadata) *more carefully*. It may have some performance benefits if disabled. Disabling this synchronization is recommendable only for cases that are definitely only processed by 1 user at a time.

Labels and comments of different examiners can optionally be visually distinguished, by showing the creating examiner's initials (default), or alternatively other abbreviations of their names or (if no abbreviation is specified) their complete usernames. Examiners can choose whether or not they get to see labels of other users or only their own (or, if half checked, only their own associations plus those of unknown users). The same file can get a particular label only by 1 examiner. X-Ways Forensics imports and shows newly created labels of simultaneous other users in shared analysis mode when re-opening an evidence object or when case auto-save interval elapses or when manually invoking the Save Case command. The option to show initials for labels is represented as a 3-state checkbox. If half-checked, it has an effect on the directory browser only, not for the Export List or Recover/Copy command for example and not in the case report. It is not possible to remove labels of other examiners in shared analysis mode, only in normal mode.

X-Ways Forensics remembers the "tagged", "already viewed" and "excluded" status of files separately for each examiner. You can choose to adopt the "already viewed" status of files in volume snapshots from all other examiners when opening evidence objects. That is useful if the goal is to avoid duplicate work, if you do not wish to review files that were reviewed by any of your colleagues already. Please note that individual file statuses ("tagged", "already viewed" and "excluded") as well as search hits of other users are lost if one examiner *removes* items from the volume snapshot.

Search hits and search terms are stored on a per-user basis as well. The first examiner opening an older case with v17.5 or later will absorb the search hits and search terms that were stored in the case by v17.4 or earlier. The "Multi-user support options" dialog window contains a button that allows you to import the search hits and search terms of another user. An option is available to limit the import of another user's search hits to search hits that are marked as notable or to that user's manually defined search hits (so-called user search hits). Another option allows to *take away* the search hits from the other user when importing them. Useful if the other user is going to resume his work later and will want to import *your* search hits back when he or she is taking over again, to avoid duplications of search hits, because your search hits include his or her hits already after you have imported them.

To view *all* the results of a colleague (labels, search hits, tag marks, already viewed status of files, exclusion status of files), you can open the case in read-only mode as him or her. For that, try the "Options..." checkbox when opening a case. You may prevent your colleagues from opening the case in read-only mode as you.

The "Options..." checkbox allows you to open a case in any of the following three modes:

1) entire case read-only (case file and volume snapshots), so that you can work with the case, but cannot save changes
2) shared analysis mode (ability to cooperatively produce labels, comments, search hit hits, and virtual files; tag files; remember already viewed files, exclude files)
3) full access

If the *same* user wishes to open the same case (the same copy) in more than 1 instance of the program simultaneously, that user has two options. Either

1) in the second instance the entire case (including evidence objects) is opened as read-only, *or*
2) the user opens the case as a separate, fictitious user (called his or her "alter ego") with separate file statuses, search hits, labels etc. (shared use of the case and the evidence objects is coordinated by X-Ways Forensics exactly as if the alter ego was a real, different examiner, even though the username is the same).

The aforementioned "Options..." checkbox allows you at any time to open the case as your alter ego, not only when opening the same case in a second instance of the program. It also allows you to open a case in shared analysis mode if it is not open anywhere else at the moment.

Multiple users running searches, creating labels, entering or editing comments, editing extracted metadata, tagging files, excluding files, marking files as already viewed is all supported for the same evidence object at the same time. *Removing* items from a volume snapshot while the evidence object is open somewhere else, however, is forbidden and will be refused by the

program. The goal of the multi-user coordination in v17.5 and later is to support concurrent *analysis*/*review* work by multiple examiners. *Removing* files from a volume snapshot is not considered ordinary review/analysis work. Volume snapshot refinements should be done systematically *in advance*.

The initials of the examiner who has attached files to the volume snapshot or manually carved files in v17.5 and later can be seen in square brackets next to the filename, so that it is easy to tell who has introduced such files to the case.

Technical changes to the way how multiple simultaneously users are coordinated are reserved. To be on the safe side, please make sure that simultaneous users are running the same version of the software.

Last not least v17.5 allows you to review the processing history of a case in its properties. This reveals which versions were used on it (recorded only by v17.3 SR-10 and later, v17.4 SR-4 and later and v17.5 and later) and by which users (recorded only by v17.5 and later).

You may turn *off* "Coordinate processing by simultaneous users more carefully" for some performance benefits if there is only user of a case at a time.

There is an option to always suggest shared analysis mode when opening a case. That mode can be useful even for the first of many simultaneous users that open the same case because only in that mode newly created labels are shared out to other simultaneous users at regularly intervals (depending on the case auto-save option).

Alternative Ways of Sharing Analysis Work

Option #1: Multiple computer forensic examiners can work simultaneously with their *own* copy of the same case simultaneously (always copy both the .xfc file and the corresponding subdirectory) and exchange results with each other or reconcile all results in the main copy of the case, by exporting and importing labels (i.e. their categorization of all the relevant files, e-mails, etc.).

Option #2: Potentially relevant files are copied from the original evidence objects to multiple evidence file containers. The containers are examined by different investigators simultaneously in newly created cases (in X-Ways Forensics or X-Ways Investigator). They also can export their labels, which can then be imported back into the original case.

Both commands, the export and import of labels, can be found in the context menu of the case tree. Export is supported at the case and evidence object level, import at the case level. Please note that you cannot import labels in the original case any more if you have taken a new volume snapshot or if you have removed objects from the volume snapshot in the meantime, because in that situation it is not guaranteed that the internal IDs of the file remain the same and that a reliable association is possible. The import works only if you import into the same evidence object that you export from (the same evidence object in the same case or a copy of the same case). It does not help if it's the same image or disk in a different case. Even if it is the same case and the disk or image was removed from the case and later added again, it will not be considered the same evidence object any more. However, you (e.g. as a user of X-Ways Investigator) can

export from an evidence file container in a new case and have a user of X-Ways Forensics import the labels into the original evidence object in the original case, from which the files in the container originate. That is possible because the evidence file container has information that allows to identify the original evidence object.

Distributed Volume Snapshot Refinement

X-Ways Forensics allows to refine the volume snapshots of *different* evidence objects of the same case using multiple machines on the same network, simultaneously, to save time through parallelization.

Each user/computer opens the same .xfc case file (the same copy on the same computer). All participating users/computers or all except for one (the master session) have to open the case as partially read-only, i.e. only allowing for shared analysis work/distributed volume snapshot refinement. This can be done by checking the "Options…" box in the Open Case dialog window, or you will be prompted automatically when opening the case if the case if already open in another session as not read-only (i.e. in the master session). Other sessions will see the refinement results at latest when refinement has completed and when the respective evidence object is re-opened. The case does not have to be closed and re-opened.

You have the option to specifically open individual evidence objects (not the entire case) with the volume snapshot treated as read-only, using a dedicated command in the evidence object context menu in the Case Data window. Please note that this has nothing to do with how the evidence object itself (the disk or the image) is treated. X-Ways Forensics never alters data in sectors of disks or interpreted images files when opening them as evidence object. Only the volume snapshot, i.e. the database with information about all the files and directories found, is either read-only or, and that is the normal state, changeable.

5.4 Evidence Objects

You may add any currently attached computer storage device (such as hard disk, SSD, memory card, USB stick, CD-ROM, DVD, ...), any image file, directory, file archive or ordinary single file to the active case. It will then be permanently associated with this case (unless you remove it from the case later), displayed in the tree-like case structure, and designated as an *evidence object* or *source of evidence*. A subfolder is created in the case folder for each evidence object, where by default files will be saved that you copy/recover from that evidence object, so it will always be obvious from which object exactly (and from which case) recovered files originate.

In the evidence object properties window, you may enter a title or number for that evidence object according to your own conventions. You may change the order of evidence objects in the case tree using the small arrow buttons in the upper left corner, except for "dependent" evidence objects (partitions that belong to a physical disk). The date and time it was associated with the active case is recorded and displayed. The internal designation of the evidence object is displayed as well as its original size in bytes. You may enter comments of arbitrary length that apply to the evidence objects, and a technical description of it is added by X-Ways Forensics automatically (as known from the Technical Details Report command in the Specialist menu, plus some

essential information about Windows installations, if found in a partition). You may have the program calculate one or two hashes (checksum or digest) on the evidence object and verify them later, so that you can be sure that data authenticity has not been compromised in between. Hashes stored in evidence files are imported automatically when added to a case. The button with the folder and magnifying glass allows to quickly open the default output directory for the evidence object. Hold the Ctrl key while clicking to navigate to the internally used directory instead, where the volume snapshot is stored.

To add lose files, images or storage devices to a case you can use the "Add" commands in the case data window's File menu. When adding images, you can indicate that the volume snapshot of newly added evidence objects should be refined immediately.

File archives can be added as files. Spanned/segmented Zip and 7z archives in 7-Zip and WinZip styles are also supported. For archives created in 7-Zip style just please make sure that you add the first segment to the case, which is the segment named .001. For WinZip style the first segment is named .z01 and the last segment is .zip, and you can add either one. Even if you add the last segment (.zip), which is intuitive because of its well-known extension, all segments will be found and internally concatenated as needed. This also works if the extension of the last segment is .ufdr instead, i.e. in the case of a spanned Cellebrite UFDR report. For password-protected (encrypted) Zip, 7z and Rar archives you will be prompted for the password. The password can be saved in the case so that you do not need to enter it again when re-opening such an evidence object. You can add Cellebrite UFDR report files to a case just like normal Zip archives. However, an extra effort ist made to present you with the original filenames, original paths, original timestamps, and also additional hard links. Directories that are not original directories and just used to accommodate files whose paths are unknown are marked as virtual, as are files with metadata for examination purposes.

Another way how to add opened images or disks to the case is the "Add" command in the context menu of the data window's tab. If the images of a case are stored in the case directory (not to be confused with the directory for cases), then they will be found automatically even if the path of the case changes. A dedicated case-specific default path for images can be defined in the properties of a case, which then overrides the generic default path for images. The case-specific path may be a relative path, where a . refers to the case directory and .. to the parent directory of the case directory. Please note that for performance reasons it can be advisable to store cases and images on different physical storage devices.

The command “Replace with New Image” in the context menu of an evidence object allows you to replace a disk that is used as an evidence object in your case with an image (useful if you first preview the disk before you acquire it, i.e. created an image of it), without losing your volume snapshot, search hits, comments, etc. Can also be used to simply tell X-Ways Forensics the new path of an image in case the image was moved or the drive letter has changed, or if the image filename was changed, or if the type of the image was changed (e.g. raw image to be replaced with a compressed and encrypted .e01 evidence file). In the case of a physical, partitioned evidence object it is recommended to apply this command to that parent object (i.e. the physical disk). The change will then automatically also be applied to the child evidence objects (i.e. partitions). If the new image is an image of a different disk or a different evidence file container or an evidence file container that has been filled further, i.e. if the volume snapshots cannot match, you will likely get a warning because the size of the new image is different from the size

of the previous image. Time and again, users of X-Ways Forensics try to use this command to replace an evidence object in a case with a *different* evidence object, although that doesn't make any sense because that way the technical description, the volume snapshot, any search hits, comments and labels don't match the other evidence object. These users then typically complain that they receive an error message. The message is displayed because X-Ways Forensics usually notices based on the size that the new image is a totally different image. If you don't need evidence object A any more in your case and you need add an evidence object B, then you can simply remove A and add B. There is no alternative to that, and an alternative is neither reasonable nor required.

It is possible to open an evidence object even if the disk or image is not currently available, via a special command in the evidence object's context menu, to see at least the volume snapshot. That means you can see all the file metadata stored in the volume snapshot (filename, path, file size, timestamps, attributes, etc.), can use most filters etc., but cannot see any data in sectors and cannot open/view any files. You can also open evidence objects specifically as read-only, meaning changes to the volume snapshot will not be saved. And there is a special command in the context menu of an evidence object that is a BitLocker volume that X-Ways Forensics knows how to decrypt. That command allows to open such a volume *without* decrypting the data in any of its sectors, to see what data are actually, literally stored in them. In that state you could for example run physical searches or carve data automatically or manually.

In the Case Root window, evidence objects can be marked as important with a yellow flag, via the context menu or by hitting the Space bar. You will see that yellow flag in the Case Data window and when selecting evidence objects, for example for recursive exploration from the Case Root or when generating a report.

In the properties of evidence objects with a FAT file system you can optionally define which time zone the local timestamps in that file systems are based on, if you have an idea/opinion about that. That time zone depends on the settings of the computer or device that wrote to the file system. (Keep in mind that those settings may have changed over time and thus a single time zone may not be adequate to get all timestamps right.) If you define the time zone reference, file system level timestamps are presented according to the selected display time zone and not in their original local time any more. They are internally converted from local time to UTC (based on your time zone reference) and then from UTC to the display time zone, at the moment when the timestamps are displayed. The effect is not permanent, the reference time zone settings can be changed at any time. The definition of a time zone reference is lost if you open a case in versions older than v19.3.

When copying files from FAT file systems to an evidence file container, file system level timestamps of these files are usually marked in the container as based on an unknown local time zone so that they will not be time zone adjusted when reviewing the container in the future. If however you are certain about the original time zone and define the time zone reference for the source evidence object, the timestamps are converted to UTC within the container based on the reference time zone and marked in the container as timestamps in UTC, permanently. In that state the timestamps later will be adjusted according to the selected display time zone, even if you change your mind and change the reference time zone in the source evidence object. The evidence file container is self-contained and separate from the source evidence object once files have been copied.

A command in the case context menu allows to import selected evidence objects from another case into the current case, for example when you wish to merge different cases (that may have been worked on by different users to split up the workload) into a single case. This will also import (actually: copy) an evidence object's volume snapshot with labels, comments, bookmarks, search hits, indexes, events, RAID reconstruction parameters, time zone selection, and much more, but not volume snapshot backups and not the users (examiners) of the other case and the distinction between their own labels and search hits. The current user who conducts the import will absorb those results. Label names in the selected imported case that are identical to existing label names in the target case are merged with the latter. The timestamp recorded when the evidence object was added to the original case will be taken over into the new case. The unique IDs of files will be different in the new case. However, labels for that evidence object can be exchanged (exported and imported) between the source and the destination case because the volume snapshot IDs and internal IDs are retained. The command to import an evidence object from another case can also be used to simply duplicate an evidence object in the same case. Simply select the .xfc file of the currently active case to do that for the tagged evidence objects. This could be useful to maintain and see and compare two volume snapshots at the same time, experiment with file header signature searches with untested signature definitions etc.

5.5 Case Log (Activity Log)

When enabled in the case properties, X-Ways Forensics logs all activities performed while the case is open. That allows you to easily track, reproduce, and document the steps you have followed to reach a certain result, for your own memory, to show to your colleagues, for the court room, etc.

The following is recorded:

- when you a select a menu item, the command title (or at least an ID), and the name of the active edit window, if not an evidence object, preceded by the keyword "Menu",
- when a message box is displayed, the message text and what button you pressed (OK, Yes, No, or Cancel), preceded by the keyword "MsgBox",
- when a small progress indicator window is displayed, its title (like "Recovering files...") and whether the operation was completed or aborted, preceded by the keyword "Operation",
- a screenshot of each displayed dialog window with all selected options, e.g. for a complex operation that follows, preceded by the window's title,*
- the extensive log produced by Clone Disk and File Recovery by Type,
- your own entries (free text) that you add with the Add Log Entry command, either to the case as a whole or to a certain evidence object.

The destination path of each file copied/recovered with the directory browser context menu, along with selected metadata of that file (e.g. original name, original path, size, timestamps, ...), is logged in a separate file “copylog.html” or “copylog.txt” in the “_log” subdirectory.

All actitivities are logged with their exact date and time, internally in FILETIME format with 100-nanosecond interval precision. Logs are by default associated with the case as a whole.

However, logs of activities that apply to a certain evidence object are directly associated with that evidence object. This determines where they appear in a report. To output the activity log, please generate the case report. Screenshots are saved separately as PNG files in the „_log“ subfolder of a case folder.

*If the checkbox for case log screenshots in the case properties is half-checked, that means that no actual graphical screenshots of dialog windows will be taken, just a simple text representation will be stored in the log (the same that you get when via Ctrl+C). These details are included in a special way in the HTML output, so that they do not detract too much from the main log entries. Either they are output in a smaller font and gray color (if "With screenshots" in the report options is fully checked) or simply as a pop-up when hovering with the mouse cursor over a space-saving placeholder rectangle (if half checked) or not at all (if not checked). The placeholder rectangle and pop-up work best when viewed in Google Chrome, as that browser does not truncate the text if lengthy and even shows a preview of the first line in the placeholder rectangle. If you have X-Ways Forensics take conventional (graphical) screenshots of dialog boxes in the log, pixels with the gray background color can be changed to pure white, to save toner/ink in case you are going to print your log at some time (anyway, please think twice and save paper).

5.6 Case Report

You may create a report from the File menu of the Case Data window. The report is saved as an HTML file and can thus be displayed and opened in a variety of applications. For example, you may view it in your favorite Internet browser and open and further process it in MS Word. The application to open the report in can be specified in Options | Viewer Programs. If no such program is defined, the report file will be opened in the application that is associated with the file extension on your computer. With the Open Report command you can select any existing file and open it in the defined or associated application.

The report can consist of the following elements:

- **Basic report**: Starts with an optional header line, an optional logo, an optional preface (in which you may use HTML code), the case title and details, followed by a list of hyperlinks to the individual evidence object sections. For each evidence object, the report specifies its title, details, and technical description, your comments, your annotations. If only half checked, technical details about the evidence objects are not included in the case report, the evidence objects are merely listed. There is an option to not show internal information such as examiner name and case path and image paths in the case report, if the report is generated for people outside of your organization. There is also an option to not show the technical description of evidence objects. That could be useful to avoid unnecessary discussions with computer laypersons in court or elsewhere about what a "sector size" is etc.

- **Report tables**: All files with selected labels can be output to the report in so-called report tables, with selected metadata such as filename, path, timestamps, comments. Files can be optionally copied off the evidence objects into a subdirectory of where the report is saved. Then they will also be linked from the report. Either all files can be copied or merely

pictures. If only pictures, for videos at least the first still image (if available) will be copied and used to represent the video in the report. By default, pictures will be displayed directly in the HTML report file and not merely linked. They are resized to the maximum dimensions you specify while retaining their aspect ratio. If you specify maximum dimensions of 0×0, then the pictures will only be linked, just as other files. If you choose to reference multiple files in the same line (to render the report more compact when printing), you will appreciate that long filenames and paths can be artificially broken into multiple lines after a user-defined number of pixels, to make sure the width does not exceed the paper size.

There is an option to only make a copy of tagged files for inclusion in a case report instead of all or none. Useful if you wish to reference all notable files with their metadata in your report, but show only a subset of those. Files of certain supported types can be converted to PDF format, for recipients of the report that otherwise would not have suitable applications to view the files. You can define the file types that do not need to be converted, e.g. those that can easily be displayed by a web browser or with Windows tools. If a conversion is not possible, the original file will be copied unconverted.

Files can be output either grouped by evidence object and sorted by internal ID or in the order as they are currently listed in the case root window, where you can freely change the order thanks to up to 3 sort criteria. If no files are currently listed in the case root (because it has not been explored recursively), then the second option is grayed out. Explore the case root recursively first to make it available (right-click it). Note that if you choose the second option, files that are not listed in the case root window will not be output, even if they are part of a report table (have the appropriate label). That means that current filter settings have an effect on the generation of the report, too. If files are omitted because they are not listed in the case root window at the time of report generation, you will be notified of that in the report and in a message box.

If the box to output report tables is only half checked, then only the number of items that have the label in question will be reported.

Many different settings allow to tweak the report to your liking. For example, you can name output files after their unique ID to ensure filenames are succinct, unique, trackable and reproducible, and this will also ensure that if the same files is included in multiple report tables, it will be copied to the report subdirectory only once. That saves time and drive space. You can also name files after their hash values or various other more or less unique properties. If those happen to be blank, the original name will be used.

"List each file only once" is a 3-state checkbox. If fully checked, no file will be referenced in the report by more than one report table. Note that you can still see all labels of a file when it is listed in its first report table in the report, if you output the field "Labels". If the checkbox is half-checked, that means that a file will still be referenced (listed) by additional report tables in the report if it has multiple labels, but copied only once and linked only from the first report table.

A special option allows to output the complete internal metadata from a file in the case report as known from Details mode, in HTML format, instead of the extracted subset in

the Metadata column in plain text. If you wish to output hash values of the files in your case report, and you did not compute hash values previously by refining the volume snapshot, the hash values can optionally be computed on the fly when generating the report.

Smaller versions of pictures can optionally be generated specifically for the report, to greatly reduce the memory requirements of the Internet browser or word processing application when loading the HTML report, and to accelerate loading. This can make a big difference for reports with many high-resolution photos. The JPEG compression factor is user-definable. The resolution depends on the specified "maximum dimensions of pictures". The checkbox that represents this option is a 3-state checkbox. If half checked, the smaller versions of the pictures are used only for the preview directly in the HTML report. If fully checked, even when clicking the picture in the report you will only see the smaller version, and the original larger file is not included in the report at all. This can be beneficial if your main concern is the drive space requirement of your report with linked files, not the output quality of pictures.

The report can optionally also show previews/thumbnails of non-picture files, e.g. Office documents, e-mails, web pages, programming source code, etc. etc., similar to the gallery. You can shrink the preview representation slightly or a lot or not at all, to either be able to read some of the text right in the report without opening the document or to get a better impression of the overall formatting of the text and just see logos etc.

- **Search hits** that are marked for inclusion in the report can be output optionally, with their context to the left and to the right. File-related search hits are output in the report table section about the respective file, along with all the selected file metadata, if the file is part of a report table and that report table is actually output in the report. If not, such search hits can be found in the section about the evidence object to which they belong. Purely physical user search hits (defined in Disk/Partition mode, not File mode) are always output in in the section about the evidence object.

- Selected **events** from all selected evidence objects can be included in the case report, near the end, in the order that was last defined in an event list, e.g. sorted by timestamps for a chronological timeline view.

- **Case log**

By default, the report is created for the entire case. Optionally it is created for selected evidence objects only. It is relatively easy to use CSS (cascading style sheets) for case report format definitions. In addition to defining the parameters for standard HTML elements, key elements of the report are assigned "class" parameters to simplify targeting those for formatting purposes. Example style sheets are available to use as a basis for further modification. The report options allow picking or editing a CSS file as part of the reporting process. The default is "Case Report.txt". The default look from v18.0 and earlier is still available as "Case Report Classic.txt".

You have the option to convert the HTML case report to PDF format. This cannot be used in conjunction with the option to split the report file after a certain number of files. If the box with

the PDF option is fully checked, that means that you wil receive *only* a PDF version of the report. If half checked, that means that you you will receive both an HTML and a PDF version of the report. Please note that if you delete one of them in the Windows Explorer/File Explorer, this will automatically also deleted the corresponding subdirectory with the copied files if there is one, even if it is still needed for the respective other version of the report.

5.7 Labels

In the directory browser of an evidence object, you can add labels to notable files. A label is a user-defined (virtual) marker for a file, for basically any purpose that you can think of. Files with certain labels can easily be included in the case report with all their metadata and even links (pictures can be included directly), and you can filter by their labels in a recursive view in order to easily locate these files later (like bookmarking files). The filter can reference multiple labels at the same time (with OR, AND and NOT operators) and even has an option that allows to additionally include siblings of the files with a certain label, i.e. files in the same directory. That is useful, especially when exploring recursively and sorting by path, to check whether there are any further notable files in the neighborhood.

E.g. you could create labels like "related to company X", "evidence against suspect A", "incriminating pictures", "unjustified expenses", "forward to investigator B", "print later", "get translated", “show to witness C” etc., and later when you are done viewing files, you can get the big picture of all relevant files by using the label filter (e.g. "Show me all files related to company X that are also considered evidence against suspect B"). You are practically assigning files to certain custom categories defined by yourself. This also allows you to revisit files later that are still to be closely examined.

Having files associated with a particular label also allows to conveniently recover/copy them in a single step at a later point of time or get a gallery overview of these files specifically. The same file can get multiple labels. This can be done in the dialog window that appears when invoking the Labels command in the directory browser context menu, for one file or several selected files at a time. This dialog window does not show the existing labels of the selected file or files (that would be quite complicated to achieve anyway for multiple selected files, for that instead simply look at the "Labels" column), but creates new labels in a convenient and user-configurable way and/or removes existing labels. The program remembers the label names selected last for creating new labels. In the same dialog window you can also create new label names, rename or delete existing ones, and remove/override previous labels. For each label name you can specify whether you would typically like to associate only the selected file or directory with that label name and/or at the same time the selected file's parent file (if any) and/or the file's or directory's child objects and/or any identified duplicates of the selected file in any currently open evidence object (duplicates that have been identified based on hash values and marked accordingly in the Attr. column, as well as hard links except in HFS+).

Another option allows to automatically label siblings of selected files at the same time. Useful for example when reviewing search hits, if you find a relevant search hit in the attachment of an e-mail message and want to be sure to include other attachments of the same e-mail message in further processing, even if they do not contain search hits.

If you need to categorize a lot of files with the help of labels, you can also use keyboard shortcuts. X-Ways Forensics automatically assigns the shortcuts Ctrl+1, Ctrl+2, ..., Ctrl+9 to your label names. In the dialog window for labeling you can also assign these shortcuts to label names yourself, by simply pressing the keys while a label name is selected. Alternatively you may simply press the keys in the numeric pad on your keyboard if Num Lock is active, without Ctrl. This will not be considered normal input in the directory browser although the Ctrl key is not pressed. The numpad keys may not work on all computers. Ctrl+0 removes all labels from the selected file, except those that were assigned automatically by X-Ways Forensics and serve as hints for the user or that represent detected picture content. Alt+1, Alt+2, ..., Alt+9 removes the corresponding label from the selected files.

Optionally the next item in the directory browser can be automatically selected after attaching a label to one item. A 3-state checkbox allows you to do that either never or only for labels created with keyboard shortcuts or for all association methods.

You may enter a free text description for any label name, by clicking the button with the "properties" icon in the dialog window for labeling. The description will be included in the case report if the files with that label are output. Useful for some explanation of what a report table is about. Helps to keep the label name itself, which appears at many places in the user interface, more concise.

Label names can be alphabetically sorted in the dialog windows for filtering and for labeling. By default, they are listed in the order in which they were created. Label names that were created by the application as hints for the user are listed optionally, and they are shown as indented. You can change the order of labels in those dialog windows if they are not sorted by name, using the arrow buttons. Changing the order has an immediate effect also on the order in which labels are listed in the Labels column of the directory browser. That way you can make sure that the labels that are most important to you are listed first.

There is an option to label files based on search terms that they contain according to the "Search terms" column. Useful if you wish to keep the information about which file contains which search terms even after deleting search hits, or to preserve it in evidence file containers.

Another option allows to convert matching hash sets to labels. This can be useful for example if you wish to recreate your hash database from scratch or delete your hash database, and do not only wish to preserve the categorization of known files in the volume snapshot, but also the exact matching hash set names. Also useful if you wish to add files to an evidence file container and wish to let the recipient know the original hash set matches, not only the categorization. These auxiliary label names are highlighted in a different color to distinguish them from other kinds of label names. Hash set based labels can also be created on the fly when copying files to an evidence file container.

In total there are really many kinds of labels: 1) user-created labels, which may or may not be meant for report purposes, 2) labels created by X-Ways Forensics to make the user aware of special properties of files (hints), 3) labels representing search terms that are contained in a file, 4) labels representing hash sets in which a file was found, 5) labels representing groups of duplicate files, 6) labels representing photo content identified by Excire. To avoid a bloated list

of report tables available for selection during report creation, labels are offered in that dialog window only if they are actually intended for report purposes. That is assumed by default for all user-created labels. You can toggle the report purpose of each label name in the dialog window for labeling with one of the check boxes. Optionally you can associate labels with light bulbs in four different colors. Those light bulbs appear in the Name cell of labeled files. They can also simply help you locate certain labels in long lists of label names, e.g. in the label filter dialog window.

It is possible to save and load lists of label names and their descriptions in the dialog window for labeling. This is useful to start right away with a set of predefined label names as typically needed for a certain kind of case. The file saved is human editable. The format is either of:
a) One label name per line (if no descriptions required). No blank lines permitted.
b) A label name, preceeded by the Unicode character U+25B8 (Black Right-Pointing Small Triangle), and followed by newline and any description. Repeat as required.
In both cases the file must be in UTF-16 format with an initial byte order marker. The maximum number of report tables in a case is 1000.

Labels themselves (the association of files with label names) can also be exported and imported. See Alternative Ways of Sharing Analysis Work.

An easier-to-use and simplified version of the dialog window for labeling is available, with less settings that might confuse new users, which is the default in X-Ways Investigator, and optionally available in both X-Ways Forensics and X-Ways Investigator. In the simplified version labels that are created by the application to make the user aware of something will not be listed.

In order to output files with certain labels in report tables in a report (the original purpose of labels), use the Create Report command in the Case Data window.

As mentioned before, labels are also used internally and created automatically by X-Ways Forensics, to make the user aware of various potential specialties of certain files. It is up to you whether you wish to follow up and take a closer look at those files or not. The names of internally created labels are displayed as indented and in a different color, to avoid mix-up with your own labels. Automatically generated labels include:

No detectable textual contents
Unable to decode text
For error messages see Metadata
Unable to explore
Empty archive?
Spanned archive
No e-mails found
Path too long.
Large non-resident $EA
Animated GIF
Animated PNG
Multi-page TIFF
Multi-page JPEG marker

Phone screenshot?
Zip bomb? Not fully processed
Unexpected tail (SFX?) / Contains unknown segment (SFX?)
FSG Packer / PECompact / UPX / Unknown segment / Binder?
Contains embedded document(s)
Contains embedded object(s)
Contains embedded file
Contains hidden file
Hybrid MS Office document!
RAR hybrid
Contains embedded non-JPEG/non-PNG picture
Contains invisible old revisions
Concatenated-PDF
Contains private chunk
No pictures extracted
Reason for crash?
Unsupported file type variant
Omitted
Not copied
Virus suspected
Unable to read
Not decompressed

5.8 Viewer Functionality

The internal viewer can be invoked with the "View" command in the Tools menu and in the directory browser's context menu, plus in Preview mode. It shows picture files of various file formats (JPEG, PNG, GIF, TIFF, BMP, WEBP (only the first frame if animated), HEIC, some DICOM variants, PSD, HDR, PSP, PCX, CUT, ICO, using an internal graphics viewing library) plus the structure of Windows registry files, Windows Event Logs (.evt and .evtx), Windows shortcut files (.lnk), Windows Prefetch files, $LogFiles, $UsnJrnl:$J, Ext3/Ext4 .journal, .ds_store, Windows Task Scheduler (.job), $EFS LUS, INFO2, Restore Point change.log.1, wtmp and utmp log-in records, MacOS X kcpassword, MacOS X finder bookmarks (flnk), AOL PFC, Outlook NK2 auto-complete files, Outlook WAB address books, Internet Explorer travellog files (a.k.a. RecoveryStore), Skype Chat Sync, MS Outlook Express DBX and many other files internally. If you try to view a file that is not supported by the internal viewer, the separate viewer component is invoked instead.

There is an additional separate viewer component that integrates seamlessly and allows to conveniently view more than 270 (!) file formats (such as MS Word, Excel, PowerPoint, Access, Works, Outlook; HTML, PDF, StarOffice, OpenOffice, ...) directly in WinHex and X-Ways Forensics. This component is provided to all owners of forensic licenses issued for v12.05 and later. It can be enabled in Options | Viewer Programs, optionally also for pictures that could be displayed by the internal graphics viewer library. More information online. The folder for temporary files used by the separate viewer component is controlled by WinHex/X-Ways Forensics, i.e. set to the one the user specifies in General Options. However, unlike X-Ways

Forensics, the viewer component does not silently accept unsuitable paths on read-only media. Please note that the viewer component since its version 8.2 creates files in the Windows profile of the currently logged on user, in which it stores its configuration and settings. In earlier versions, if actually used, not when merely loaded, it left behind entries in the system registry.

The viewer component allows to view or preview certain password-protected documents if the password is available. Only certain encryption variants of Microsoft Office and PDF documents, Microsoft Outlook PST 97-2013, and Zip files are supported. When previewing such a file, the password will be taken from the Metadata cell of that file (if available from there in a line that starts with "Password: ") or otherwise all passwords from the currently active case's password collection will be tried automatically. If one of the passwords from the password collection matches, it will be remembered in the Metadata cell of the file for future re-use and the user's information. When viewing such a file, if no matching password is found, the user will be additionally prompted for the password repeatedly until he or she provides the correct password or gives up (clicks Cancel).

Registry Viewer

MS Windows maintains an internal database called registry which contains all important settings for the local system and installed software in a tree-like structure. The data is persistently stored in files called registry hives. You can open and view hives by double-clicking them in the directory browser or using the context menu. This will open them in the integrated registry viewer. Supported formats are NT/2K/XP/Va/7 hives. Win9x and WinMe hives can only be loaded by the registry viewer of X-Ways Forensics 15.9 and earlier. NT/2K/XP/Va/7 hives are located in the file “ntuser.dat” in a user profile and in the directory \system32\config.

Up to 32 hives can be opened in the registry viewer at the same time. The registry viewer has the ability to find deleted keys and values in hives that contain unused space and lost keys/values in damaged/incomplete hives. If no complete path is known for keys, they will be listed as children of a virtual key called "Path unknown".

With a right-click a pop-up menu can be opened anywhere in the window, which lets you invoke the commands “Search” and “Continue Search”. Clicking “Search” invokes a dialog that lets you specify a search expression and where you want to search. You can browse either keys or names or values or all of them. The search always starts at the topmost root of the first loaded hive and spans all opened hives. "Continue Search" finds the next match after at least one match has been found. The currently selected element is not relevant for where the search continues. The "search whole word only" option is not guaranteed to work for values.

In the right-hand window the pop-up menu also contains the command "Copy" which lets you copy the value of the selected element to the clipboard.

When clicking a value of a loaded hive in the Registry Viewer, if the data window with the drive/image from which the hive was loaded is in File mode, the cursor will automatically jump to the selected value in the registry file, and the value will automatically be selected as a block in that file. Useful as that allows to see the value in hexadecimal and text and as that allows to easily copy binary values in either binary or as text, not only as hex ASCII.

The Export List command in the registry viewer context menu allows to export all values in the selected hive to a tab-delimited text file.

When selecting a value, an edit window in the lower right corner tells you the logical size of that value and the size of its slack. It also interprets registry values of the following types, as known from the registry report: MRUListEx, BagMRU, ItemPos, ItemOrder, Order (menu), ViewView2, SlowInfoCache, IconStreams (Tray notifications), UserAssist, Timestamps (FILETIME, Epoch, Epoch8), MountedDevices, OpenSavePidlMRU, and LastVisitedPidlMRU. The edit window also displays the access rights/permissions of the registry keys if (Default) is selected.

$LogFile Viewer

Basic Concepts:
Each statement falls into one of the three categories:
1) Log-Operation
The on-disk data at (LCN,Byte offset) is to be replaced in case of a Redo/Undo-Operation with the one specified within the log operation.
2) The PAGE statement indicates the start of a new log page (multiple of 4 KB). The LSN specifies the last end LSN for this page. A * marks a stale page.
3) The CheckPoint statement specifies a LSN to restart with.
Each statement is preceeded by an byte offset pointing into the $LogFile.

Abbreviations:
LSN=Logical Sequence Number
LCN=Logical Cluster Number
VCN=Virtual Cluster Number
FID=File ID

Limitations:
Only log operations are shown which affect on-disk structures. FILE records and INDX buffers are not completely dumped. For complete data, follow the byte offset displayed for the operation of interest. An NTFS journal is only processed if the path of such a file contain the string $LogFile.

5.9 Registry Report

From within the registry viewer, WinHex can create an HTML report, listing values of possibly relevant registry keys, when you invoke the command "Create Registry Report" in the right-click pop-up menu. The registry keys that are to be reported in all open hives are defined in text files like the pre-supplied “Reg Report *.txt”, which can be tailored to your needs. The registry files you view must have their original names, or else the report may fail. You may edit the list of registry keys in this files to tailor the report to your own needs.

Standard tables have 4 columns: description, extracted value, registry path (provided as a tooltip), and last modification date of the corresponding key. The dates are displayed in gray for values

that are not the only values in their respective key, as a visual aid to remind the reader that they are not the modification dates of the values themselves.

Free space in registry hives can be analyzed with the report definition file "Reg Report Free Space.txt". The free space can be as large as several MB, especially as a consequence of the use of virus scanners and registry cleaning programs. Deleted registry values are now highlighted in the report in red color.

Also registry value slack has a relevant size in NTUSER.DAT hives. This fact is exploited with 2 measures:
1) If the slack contains text strings, it will be output in the registry report (in green). This new feature can optionally be turned off the registry viewer context menu.
2) For values that contain item lists (i.e. are binary) you can use the "Reg Report Free Space.txt" definitions to output registry report will output lists of filenames with timestamps in green. The first timestamps is an access date, the second one is a creation date. If no timestamps can be output, these are artifacts from "RecentDocs".

Format of entries in "Reg Report *.txt"

(*type*) (*tab*) (*registry path*) (*tab*) (*description*) (*linefeed*)

type:
?? definition for any Windows version
NT for Windows NT through XP
VT for Windows Vista and 7
** new function (without absolute paths)
FR query in free space of the hive

registry path:
Full path of registry keys

HKLM: HKEY_LOCAL_MACHINE
HKCU: HKEY_CURRENT_USER

If an asterisk ("*") is provided as the last key, all keys on the same level and deeper and their values will be included in the report.

example:
NT HKLM\Software\Microsoft\Windows\CurrentVersion* report whole Windows branch

If you wish to report a particular value that exists in all subkeys of a certain key, you can as well write an "*" for all subkeys and include the value after that.

The generated report contains the registry path with its timestamp, the filename of the registry hive that the key was found in, the description that was provided in the "Reg Report *.txt" file, and the value.

The description field may contain an additional statement at the end that starts with a %

character. If the % is followed by a numeric character n, the n-th element of the registry path will be appended to the description in the report. This can be very useful if the path and not the value (or not only the value) contains the relevant information. If the % is followed by a letter, the value will be preferably interpreted as the data type that the letter stands for. The following letters and data types are defined at the moment:

%f Windows FILETIME timestamp
%e Epoch (Unix) timestamp
%E Epoch8 (Unix) timestamp as QWORD
%D decimal number
%T Windows system time timestamp
%s ANSI-ASCII null-terminated
%S UTF-16 string null-terminated
%b binary data not to be interpreted as characters (REG_BINARY)
%P Windows PIDL data structure
%I ItemPos data structure (covers Shell Bag, desktop shortcuts, and more)
%B conditional: if value TRUE
%F conditional: if value FALSE
%- no empty mode
%+ recursion of the subtree
%i value case-insensitive
%d deleted values only

It is also possible to combine numeric characters and letters (e.g. %10f). In that case the numeric character must precede the letter.

// at the start of a line comments out that line (will cause it to be ignored).
at the start of a line will output explanatory text into the report.

Additional output

In a second phase of the creation of the registry report, additional data will be analyzed and output as tables at the end of the HTML file. The specifications in the definition file which belong to this second phase are marked with "Dummy". This causes the first phase to prevent any normal output. If you would like to get the output of the first phase, you merely need to change the description in the definition to anything other than "Dummy".

The table "Attached devices by serial number" is created according to the algorithm that Harlan Carvey describes in chapter 4 of his book. Furthermore you can find the tables "Partitions by disk signature", "Windows portable devices", "Drivers installed", "File systems installed", "Services installed", "Networks", and "Network cards".

Another table is called "Browser Helper Objects", compiled with data from the hives NTUSER.DAT and SOFTWARE, about browser usage. "External Memory Devices" is a table which can be retrieved from Software hives of Windows Vista and later that lists external media with access timestamps, hardware serial number, volume label, volume serial number and volume size (size often only under Vista). Select the definition file "Reg Report Devices.txt" to get the table.

5.10 Simultaneous Search

This search command in the Search menu is available for owners of specialist and forensic licenses, and offers all options only for owners of forensic licenses. This search is simultaneous in that it allows the user to specify a virtually unlimited list of search terms, one per line. The occurrences of these search terms are either saved and listed in an evidence object's search hit list (forensic licenses, when working with a case, for full functionality, highly recommended), or in the general Position Manager.

Search hits for identical search terms are by default merged and made accessible through the same item in the search term list. This is useful for example when running searches for the same keywords / regular expressions incrementally (in multiple runs) in different evidence objects. However, there is a check box that you can uncheck in order to always produce a new item in the search term list, even if the keyword that you are looking for is identical to a previously used keyword or a keyword in the same run. This is useful if you run the searches with different settings (e.g. same keyword as a whole word and not as a whole word at the same time), in order to be able to distinguish the resulting search hits later.

You may use the simultaneous search to systematically search multiple hard disks or disk images in a single pass for words like “drug”, “cocaine”, (street synonym #1 for cocaine), (street synonym #2 for cocaine), (street synonym #3 for cocaine), (street synonym #3 for cocaine, alternative spelling), (name of dealer #1), (name of dealer #2), (name of dealer #3), etc. at the same time. The search results can narrow down the examination to a list of files upon which to focus.

The simultaneous search can be used to search physically in sectors or logically in file or in a previously created index. Physically, it searches the sectors on a medium in LBA order (except if you search upwards, then in reverse order). If you do not have WinHex list the hits of a physical search, you may use the F3 key to search for the next hit. Logically, the search proceeds file by file, which is preferable and much more powerful and thorough. More about the logical search.

You can search the same search terms simultaneously in in up to 6 code pages. The default code page, that is active in your Windows system, is marked with an asterisk and initially preselected. E.g. on computers in the US and in Western Europe, the usual default code page is 1252 ANSI Latin I. The code pages named "ANSI" are used in Microsoft Windows. "MAC" indicates an Apple Macintosh code page. "OEM" indicates a code page used in MS-DOS and Windows command prompts. If a search term cannot be converted to the specified code page because of characters unknown in that code page, a warning is issued. Code page independent RegEx searches for exact byte values are possible when searching in a "non" code page called "Direct byte-wise translation for regular expressions", which translates byte values without any mapping for certain code pages or case matching. X-Ways Forensics also allows to search in both little-endian and big-endian UTF-16, and in any regional Windows code page plus UTF16 with the MS Outlook cipher (compressible encryption) applied.

In X-Ways Forensics and X-Ways Investigator you have the ability to apply a character adjustment list. (The inner workings of that are slightly different here from the corresponding indexing option.) That list is expected in a UTF-16 text file named "Character Adjustment.txt". It

starts with a little-endian byte order mark, followed by one instruction per line, with an arrow (greater than symbol) in the middle, which maps one character to another. You can edit it as you see fit for searches in your own language. An example for French language searches: The line É>E means that the letter É in the original data to which the Simultaneous Search is applied (when searching in suitable code pages) will be accepted as a variant of E in your search term. You only need to search for Edith Piaf and will find both Edith Piaf and Édith Piaf. Both variants will be searched internally. ç>c means that searching for Francois (which you may find preferable if your keyboard cannot easily produce the ç character) you can find both Francois (simplified spelling) and François (original French spelling). The other way around can also make sense: c>ç means that searching for François (which you may prefer if it looks more correct to you) you can find both François and Francois. However, the latter kind of substitution is not recommended for indexing. Even if you are not interested in matching multiple spelling variants, you could define such substitutions once (e.g. using copy & paste) if you cannot easily produce special letters with your keyboard.

Case insensitivity does not work on top of the character adjustment. So for example with the adjustment é>e active, a case-insensitive search for e will find e and é as well as E, but not É. For that you need to add the adjustment É>E. Note that you could theoretically define your own case-insensitivity rules solely using character adjustments. Up to 16 mappings are possible for the same target character. Character adjustments also work in conjunction with regular expressions (only with target characters that have no special meaning in regular expressions and are not contained in [] sets).

You can define which characters should be considered to be parts of words. This is useful to avoid false hits for short real language words in binary garbage data or Base64 code and generally for users that consider numbers to be parts of words (such as in “GIF89”). Example: An undesirable hit for "band" in "7HZsIF9BAND4TpkSbSBS" can be prevented if you search for it as a whole word only if you redefine the alphabet to include digits 0-9, i.e. consider them word characters.

It is possible to review the (incomplete) search hit list in the middle of an ongoing simultaneous search. You can click the search hit list button at any time to view the preliminary search hit list. Additional search hits that have been collected as the search continues will be listed when you refresh the search hit list, by clicking the Enter button in the search term list as usually. This approach to view preliminary search hits is useful e.g. when previewing a live system on site to determine whether a medium might contain relevant files and should be captured. If after searching 5% of the data and reviewing the search hits gathered so far the answer is Yes, the search can be stopped already and a lot of time is saved.

5.11 Logical Search

Powerful subvariant of the simultaneous search. Allows to search either all files, all tagged files, or (if invoked from the directory browser context menu) all selected files. The logical search has several advantages over a physical search:

- File slack can be specifically targeted (for all files or, if only half checked, for files that

are not omitted) or ignored. In X-Ways Investigator slack space will be covered; just the checkbox for that is not visible for reasons of simplicity.

- The search scope can be limited to certain files and folders, through tagging or selecting files. Please note that the amount of data to search that may be displayed in the dialog window is an estimate only. The actual scope of the search may vary because of slack space.

- Searching in files (usually = in the cluster chains allocated to files) will find search hits even if the search term happens to be physically split in a fragmented file (occurs at the end and the beginning of discontiguous clusters).

- A logical search can be successful even in files that are compressed at the NTFS file system level, as they are decompressed for searching. This holds true even for files that were found via a file header signature search, if that was specially adapted for NTFS compression.

- There is a dedicated checkbox to control whether certain slack areas of NTFS compression are targeted. It's unlabeled, but has a tooltip. If fully checked, the undefined slack area at the end of each compression unit of ordinary NTFS-compressed files is searched raw (as is, without decompression). If that check box is at least half checked, the well-defined slack of WofCompressed files is targeted (searched raw, without decompression).

- If the contents of archives (files in ZIP, RAR, GZ, TAR, BZ2, 7Z, and ARJ, if not encrypted, forensic license only) and individual e-mail messages and attachments have been included in the volume snapshot, they can be searched as well.

- Text decoding in documents and interpretation of binary storage of numbers and dates in spreadsheets as searchable text.

- OCR capability.

- If you are not interested in each and every search hit, but merely in which files contain at least one the specified search terms, a logical search can be greatly accelerated by telling X-Ways Forensics that only one hit per file is needed, so that it can skip the remainder of a file once a hit has been recorded and continue with the next file. The resulting search hit list will be inherently and systematically incomplete, and no assumption must be made that somehow “the most useful” search hit in each file will be collected, or, if multiple search terms are used, a search hit for a search term that you consider more important will be collected. However, it is guaranteed that it contains all the files for which there was at least one hit (for one of the search terms used), and each such file once only. Such a list is sufficient (and efficient!) to manually review the affected files, comment on them, copy the files off an image or pass them on to other investigators in an evidence file container etc. Note that of course it is not possible to combine search terms with a logical AND if only 1 hit per file was recorded. That consequence is typically forgotten by unsuspecting users.

- Files that are known from the hash database (either only known irrelevant and uncategorized files or, if fully checked, even known notable files) files that have been excluded by the user or that are filtered out by an active filter can be omitted from a logical search to save time and reduce the number of irrelevant search hits. The slack of such files is still covered if the option "Open and search files incl. slack" is fully checked, so that this option has a higher priority. If only half checked, the slack of such files is omitted, too.

- The recommendable data reduction specifically omits certain files from the search to avoid that time is wasted or duplicate hits are produced unnecessarily.

  File archives of the supported types (ZIP, RAR etc.) will not be searched if the files that they contain have already been included in the volume snapshot, in order to save time. In that case *only* those extracted files will be searched, in their natural (uncompressed) state. This may be reasonable for keyword searches and in particular for indexing (which has a hard time processing e.g. Base64 code), but not necessarily for technical searches for signatures etc. Using this option constitutes a compromise. The slack of archive files is still included if the file slack option is enabled, as that option has a higher priority.

  A file that that is marked as renamed/moved will not be searched either if data reduction is enabled and if principally all files in the volume are to be searched (as opposed to tagged or selected files only) because the same file will already be searched under its current name/in its current location.

  If *.docx;*.pptx;*.xlsx;*.odt;*.odp;*.ods;*.pages;*.key;*.numbers are decoded for the search, the contained .xml files with the main contents (document.xml, content.xml, index.xml, ...) and in case of .pages any existing Preview.pdf are also omitted, to avoid redundant search hits.

  Files with a red X icon will not be searched, except if they are specifically targeted via a selection or tagmark.

- In NTFS, all "real" hard links (i.e. hard links other than SFN) except for one can be optionally omitted from logical searches and indexing. Nowadays on Windows installations often between 10,000 and 100,000 hard links of system files exist, for example 27 links to a file like "Ph3xIB64MV.dll" in directories such as
  \Windows\System32\DriverStore\FileRepository\ph3xibc9.inf_amd64_neutral_ff3a566…
  \Windows\System32\DriverStore\FileRepository\ph3xibc2.inf_amd64_neutral_7621f5…
  \Windows\System32\DriverStore\FileRepository\ph3xibc5.inf_amd64_neutral_22703…
  \Windows\winsxs\amd64_ph3xibc9.inf_31bf3856ad364e35_6.1.7600.16385_none_a…
  \Windows\winsxs\amd64_ph3xibc5.inf_31bf3856ad364e35_6.1.7600.16385_none_9…
  \Windows\winsxs\amd64_ph3xibc12.inf_31bf3856ad364e35_6.1.7600.16385_none_6…
  etc.
  By searching only in one hard link of a file, you can typically exclude several GB of duplicate data and yet don't miss anything if you search all other files. Those additional hard links that are omitted are those whose hard link count is grayed out. Search hits in

the only hard link that does get searched are marked with the hint "→Links" in the Descr. column to remind you of the other hard links of the same file in case those search hits are relevant.

- There is an option to apply logical simultaneous searches to various metadata of files in addition to the file contents. More precisely, they can be applied to the cells of any selected directory browser column such as Name, Author, Sender, Recipients or Metadata. That can spare you from pasting your keywords in the filter dialogs of various directory browser columns. That methodology is also more thorough because all the text addressed by this feature is searchable in UTF-16, whereas elsewhere the same data may be fragmented (e.g. filenames in particular in FAT), specially encoded (e.g. sender and recipients as quoted printable in e-mails), compressed, or stored in unexpected code pages. It is also convenient because any hits will be presented and listed in the same fashion as ordinary search hits in file contents, just specially marked in the search hit description column with the name of the column that the text that contains the search hits actually belongs to and highlighted in a different color. You can also filter for search hits in metadata.

  When selecting a search hit in metadata, it is automatically searched for and highlighted in Details mode, just as ordinary search hits in file contents are automatically searched for and highlighted in Preview mode.

  Note that the simultaneous search in metadata does not search in additional cell text that is displayed in a different color, such as alternative filenames and file counts in the Name column.

- Some blind spots that logical searches have in other computer forensics software products do not exist in X-Ways Forensics. Special areas in volumes are addressed even by logical searches, namely any transition from file slack to directly following free space, and in NTFS and exFAT also from known uninitialized (but physically allocated) tails of files to directly following free space, plus RAM slack in NTFS compression units.

Should this operation freeze on a certain file, remember the internal ID and the name of the currently processed file are displayed in the small progress indicator window. If this operation is applied to an evidence object and it crashes, X-Ways Forensics will tell you which file when you restart the program and assign a label to it (depends on the Security Options). All that happens so that you can exclude and omit the file when trying again.

A parallelization option (currently still considered experimental) allows you to better utilize multiple processor cores by employing multiple threads. It has an effect only when searching in evidence objects that are images or directories, not disks. The faster your mass storage solution performs (in terms of seek times and data transfer speed), the more time you save percentage-wise. In perfect conditions, this can more than double the speed of logical searches. If you select just no extra threads for the logical search, it will work as in X-Ways Forensics versions before 18.9. If you select 1 or more extra threads, searching is done in additional worker threads, and the main thread of the process will be idle, which means the GUI will remain highly responsive. In X-Ways Investigator up to 3 worker threads may be used, in X-Ways Forensics up to 16,

depending on the number of processor cores detected.

5.12 Search Hit List

Available only with a forensic license, when working with a case, for evidence objects with a volume snapshot. (Otherwise the Position Manager will list search hits.)

The directory browser can show search hits. To get into this display mode (search hit list instead of ordinary directory browser), click the button with the binoculars and the four horizontal lines, in the same bar where the Mode buttons are. It is only available for evidence objects. In that mode of operation there are four additional columns: physical/absolute offsets of the search hits, logical/relative offsets, descriptions that include the code pages in which search hits were found and hints if found in file slack, and the search hits themselves (usually with a context preview, sortable by search term, context preview not accurate for Arabic and Hebrew text or hits in UTF-8). The directory browser's grouping options have no effect when search hits are sorted by one of these three columns. The search hit description column comes with a filter that allows to focus on notable hits, hits to include in the case report, user search hits, hits in a certain code page, hits in the text extraction of documents, and hits in slack space or uninitialized tail areas of files. Search hits in all variants of UTF-16 that are not aligned at even offsets are marked in the Descr. column as "unaligned", as a small hint and explanation why you can read the text only in the alignment-aware context preview of the Search hits column, and not in the text column.

Almost all commands in the directory browser context menu are available for search hit lists as well, notably the ability to copy, view, tag and comment files. The dynamic filters based on the usual directory browser columns can be used in conjunction with search hit lists e.g. to focus on hits in files of certain types and with a certain last modification date.

The search hit list is based on the position and level in the directory tree where you click, so that you can e.g. see all search hits in files in \Documents and Settings and subdirectories of the same, and even search hits from all evidence objects of the entire case at the same time, using the case root window. Also it's possible to conveniently select one or several search terms for search hit viewing, in the search *term* list in the Case Data window. Like that it's also an easy task to find out how many search hits there are for any given search term for any level in the case tree, as that number is displayed in the directory browser's caption based on the current search hit list.

Search hit lists are “dynamic” in that they are composed “on the fly” depending on selected search terms, explored path, current filter settings and based on the settings of the search term list (logical AND combinations and the "1 hit per item" option).

You can filter search hits via the “Search hits” column filter, for example based on their context (an additional keyword appearing or not appearing to the left or to the right or within the search hit) or based on whether you have marked them as notable or not. All the filter options can be combined with a logical OR or a logical AND, and you can focus on search hits that either meet the defined criteria or do NOT meet the defined criteria.

Search hits can be marked as notable (such that a yellow light bulb is displayed on the left) with

the directory browser context menu or by pressing the Space key. With the Space key you may also remove that mark. You may unmark multiple selected search hits as notable by holding the Shift key when invoking the "Mark as notable" context menu command. Another context menu command allow to unmark all search hits in the evidence object(s) represented by the current data window as notable. This allows for incremental filtering. Example: You filter for search hits whose context contains the word "Hello". Then you mark those hits as notable (Ctrl+A plus context menu command). Then you filter for search hits that are notable AND contain the word "Hey". Then you unmark all search hits (even those that are currently not listed!), which has no immediate effect on the presented list, and mark those that are listed as notable again. The result is that all search hits that contain both "Hello" and "Hey" in their context are now marked as notable.

If you no longer need certain search hits, you can select and delete them. For example because there might be duplicates or because you would like to run a search for the same search terms in the same files again with slightly different settings. If you no longer need any search hits of certain search terms, you can select these search terms in the search term list and delete those along with all their search hits.

The context preview around a search hit directly in the search hit list can be turned on and off in the context menu in case of problems such as extreme slowness or instability.

Another context menu command allows to reposition search hits, with a relative offset change (+/-), and to resize search hits, with either an absolute new size or with a positive or negative relative size adjustment (click the arrow button to toggle). You can resize multiple search hits at the same time with the same settings.

5.13 Search Term List

Displayed in the Case Data window when in search hit viewing mode (after clicking the button with the binoculars and the four horizontal lines). The search term list contains all the search terms ever search for in the case unless deleted by the user. The search terms can optionally be sorted alphabetically in ascending order or by the listed search hit count in descending order, via the context menu of the search term list, to make it easier to locate a certain search term in lengthy lists.

Selecting search terms in the search term list and then clicking the Enter button allows you to list all the search hits for these search terms in the currently selected path, subject to filters, in the search hit list. You can select multiple search terms by holding the Shift or Ctrl key while clicking them. You may press the Del key to delete selected search terms and all their search hits permanently.

To reduce a search hit list to a list of unique files that contain at least one search hit, check "List 1 hit per item only" and then click Enter. This can be very useful if you are going to review all such files manually, ensuring that each such file is listed only once. No assumption must be made that somehow "the most useful" search hit in each file is the one that makes it to the list, or if multiple search terms are selected the one listed search hit is for a search term that you consider

more important. The reduction is non-destructive. Bringing back the original complete search hit list merely requires that you uncheck this special box and click the Enter button again.

The option to list 1 search hit per item only does not filter out search hits in slack space or in uninitialized parts of files (in the part exceeding the so-called valid data length). This is useful because the slack of a file is typically not related to the contents of that file, so any search hits in these special areas would likely have a totally different context than search hits in the logical portion of the file (and especially search hits in the uninitialized part of a file may reside in data from various different sources) and thus they need to be reviewed additionally. Please note that it is still necessary to unselect the "1 hit per item" option to separately check out search hits in conglomerates such as pagefile.sys and the virtual "Free space" file, which contain data from totally different sources. The "1 hit per item" option is most useful for documents, for which you can often tell after one quick look in Preview mode whether that particular file is relevant or not.

It is possible to see (and via the Export list command in the context menu copy) the hit counts for selected search terms in the search term list. These hit counts are based on the current settings for the search hit list that is on the screen, take all filters into account, the explored path, any active AND combination etc. It is the numbers of hits that are actually listed, not the numbers of hits that have been recorded/saved. To see the total numbers of hits, deactivate any filter and select all search terms. Note that the "List 1 hit per item only" option also functions like a filter for search hits.

You can rename search terms with a command in the context menu of the search term list, for example so that lengthy regular expressions are replaced with a more concise and easier-to-understand friendly name such as "IP addresses", "Credit card numbers", "E-mail addresses" etc. Such names can be remembered by the program such that future searches for the same expressions will immediately add entries in the search term list with the more easily recognizable friendly names. Friendly names and corresponding regular expressions are stored in the text file "Regular Expressions.txt", which you can share with your colleagues and from which you can easily copy and paste regular expressions when needed. The file can be opened from within the Simultaneous Search dialog window by clicking on the button with the yellow lightbulb (lightbulb for "ideas" for expressions to search for). You can edit the file directly with any text editor. Just keep the structure intact: Always 1 friendly name followed by 1 regular expression, 2 lines for each such pair, in UTF-16.

There are two ways how to logically combine multiple search terms with Boolean operators:

1) By default, multiple selected search terms are combined with a logical OR. To force a search term, select it and press the “+” key. To exclude a search term, select it and press the “-“ key. To return a search term to normal OR combination, press the Esc key. You may also use the context menu of the search term list for all that. The below examples describe the effect of selecting the search terms A and B depending on their “+” or “-“ status.

A
B
= search hits for A and search hits for B that occur in any files (normal OR combination)

+A

B
= search hits for A and search hits for B that occur in files that contain A

+A
+B
= search hits for A and search hits for B that occur in files that contain both A and B (AND)

A
-B
= search hits for A that occur in files that do not contain B

2) For a logical AND combination, if the search terms are *not* marked with “+” or “-“, you may also use the small scrollbar that appears when you select multiple search terms. Allows you to see only search hits in files that contain all the selected search terms *at the same time*. You can combine up to 7 search terms that way. If you select more than 2 search terms, you also have the option to be less strict and only specify a *minimum* number of different search terms in the same file, e.g. require that of search terms A, B, C and D any combination of two of them in the same file is sufficient, e.g. A and B, or A and C, or B and D, etc. (fuzzy/flexible AND combination).

In addition to the "Min. *x*" option, the search term list also offers a "Max. 1" option when multiple search terms are selected that are not forced with a + or excluded with a -. "Max. 1" will list search hits only if they are contained in files that do not contain any of the other selected search terms. For example for 3 search terms, to get the same results otherwise, you would have had to list search hits for search term A while excluding B and C, then list search hits for B while excluding A and C, and then list search hits for C while excluding A and B, which of course is not as elegant and does not show you all such singular search hits at the same time.

When 2 search terms are selected in the search term list and combined with a logical AND (using either of the two available methods), additionally you can now require that search hits must be "NEAR" to each other to be listed, to find more likely relevant combinations of both search terms in the same file, exactly like with a proximity search. The maximum distance between the search hits that constitutes "NEAR" can be defined by the user in bytes. A NEAR combination may also be applied for more than 2 selected search terms. The effect is that a search hit is listed only if *any* of the other selected search terms occurs nearby.

This paragraph quoted from wikipedia.org: The basic, linguistic, assumption is that the proximity of the words in a document implies a relationship between the words. Given that authors of documents try to formulate sentences which contain a single idea, or cluster related ideas within neighboring sentences or organized into paragraphs, there is an inherent, relatively high, probability within the document structure that words used together are related. Where as, when two words are on the opposite ends of a book, the probability there is a relationship between the words is relatively weak. By limiting search results to only include matches where the words are within the specified maximum proximity, or distance, the search results are assumed to be of higher relevance than the matches where the words are scattered.

What's more, the search term list offers a "NOT NEAR" option (abbreviated NTNR) in addition to "NEAR". With 2 selected search terms, NTNR will ensure that only search hits are listed that are *not* located in vicinity of any search hits of the respective other search term. With more than 2

selected search terms, the results are currently undefined.

5.14 Hit Count in Search Term Lists

Question: Why when all the search terms are selected with "List 1 hit per item only" are the counts returned different from when I click on each search term individually with the same setting?

Answer: Because the option is "List 1 hit per item only", and not "List 1 hit per search term per item only". Many users do not understand that. Imagine if in the same file there is 1 hit for search term A and 1 hit for search term B, and you select both A and B with that option enabled, then only 1 hit is listed, either the one for A or the one for B (up to X-Ways Forensics to decide). So the displayed hit count is 1 for one search term and 0 for the other one. If then you select the other search term only and click "Enter", the count for that search term will change from 0 to 1 because that is now the only possible search term from which hits can be listed, and up to 1 search hit is listed per file, so that 1 hit is listed.

5.15 Event Lists

Available only with a forensic license, when working with a case, for evidence objects with a volume snapshot.

When extracting metadata (part of volume snapshot refinements), X-Ways Forensics can compile a list of events from timestamps that can be found at the file system level as well as internally in files and in main memory. Conceivable sources are browser histories, Windows event logs, Windows registry hives, e-mails, etc.. An event list works exactly like a search hit list and can be displayed by clicking a button which is located next to the search hit list button, with a clock icon on it. Just like a search hit list, an event list comes with additional columns: the event timestamp, event type, event category, and some events have an individual description/additional text, for example events recorded in the Windows registry and in Internet Explorer index.dat files.

If an event list is sorted chronologically, by timestamps, it works like a timeline, which may allow you to figure out a sequence of events of different kinds stored in different places (e.g. e-mail received, attachment saved, application started, document printed, file deleted) that otherwise could not be seen together in context. You may see events from different evidence objects at the same time from the case root window, explore recursively or by path, sort by event type or event category, see all the usual file properties, view files, navigate to the definition of an event within a file (if a relative offset is available) and filter for certain date ranges.

You may mark events as notable just like search hits and filter for notable events via the Timestamp column.

Event-based analysis instead of file-based analysis is a progressive new approach with a totally different perspective that may lead to knowledge about activities recorded on computers that otherwise could hardly be gained. You may see connections (related activity) that otherwise

could be overlooked, and may be able to better explain the logic behind what has happened.

The sources of events that are exploited by the metadata extraction in this version include all the supported file systems (i.e. all the timestamps listed in the timestamp columns of the directory browser; modification, record update and last access are omitted if identical to the corresponding creation timestamp), processes in supported memory dumps, extracted or processed e-mail, as well as files of these types:
index.dat
Internet browser SQLite databases
.firefox (~55) fragments
_CACHE_001_ and _CACHE_002_
.lnk shortcuts
.automaticDestination-ms
.chrome Chromium cache data_1, data_2
.usnjrnl fragments
Registry hives*
Windows .evt event logs
Windows .evtx event logs (Most extracted events come with a description that includes the event source, the event ID and the record number. The record number allows you to quickly search for the record in the HTML preview if you need further details about that particular event.)
DataStore.edb (MS Windows operating system update events)
.hbin Registry hive fragments
.doc (last printed)
.msg
rp.log XP restore point
INFO2 XP recycle bin
.recycler Vista recyle bin
.snapprop Vista volume shadow copy properties
.cookie
.gthr;.gthr2 Gatherer and Gatherer fragments
.pf prefetch
attach timestamps from EDB
signing date from EXE/DLL/SYS/...
boot time from ETL (event trace log) files
OLE2 last modification
last saved in Office documents and RTF
Skype main.db (chats, calls, file transfers, account creation, ... - you can read entire chats if sorted chronologically)
Skype Chat Sync
internal creation from miscellaneous file types, including Exif timestamps from photos
JPEG GPS
Unix/Linux/Macintosh system logs (These events are practically of significance especially for USB device history examinations.)

* More specialized events than just standard registry timestamps are output optionally when you create a registry report, depending on the report definitions used!

The event type is displayed in gray if the timestamp is a previously valid timestamp, for example

such as those found in NTFS in 0x30 attributes or index records of INDX buffer slack or in $LogFile.

Timestamps from 0x30 attributes in NTFS file systems are output as events only if actually different from their 0x10 counterparts and not identical to the 0x30 creation timestamp. They are marked as "0x30" in the Event Type column. Malware might give itself harmless looking timestamps after deployment, so that it does not seem to be related to the time of intrusion/infection. The 0x30 attribute timestamps, however, remain unaltered (except if the file is renamed or moved later), and that is the reason why some examiners are interested in them. If the time frame of intrusion/infection is known, related files would be found in the event list thanks to the original 0x30 attribute timestamps.

0x30 timestamps are marked in the Event Type column with “> 0x10” if they are later than the corresponding 0x10 timestamps, which seems unnatural and in some rare cases might be the result of backdating by the rightful users of the computers themselves. Under certain circumstances, backdating documents is seen as fraudulent and illegal. However, much more commonly 0x10 timestamps predating 0x30 timestamps is just the work of installation programs or the result of copying a file or moving a file from one volume to another or extracting a file from a zip archive, where Windows or other programs artificially apply the original creation time of the source file to the destination once copying turns out to be successful (internal programmatic backdating).

The selections in the event type filter are not remembered by the program from one session to the next.

Please see the description of the timestamp columns for more information.

5.16 Mount As Drive Letter

Available in X-Ways Forensics and WinHex Lab Edition. (For evidence file containers with no more than 1,000 objects with any license type for WinHex, even in the evaluation version, free of charge.)

Allows to mount the volume that is represented by the active data window as a Windows drive letter, either entirely (if the command is invoked in the Specialist menu or in the case tree context menu for a whole volume) or partially (if applied to a directory or file with child object using the directory browser context menu or the case tree context menu). This allows for convenient and quick access to all files with external programs where necessary (without the need to copy the files to your own local drive letter first). Very efficient in particular if you wish to check a whole volume or directory or certain files with a virus scanner. Mounting works for all the file systems that are supported, for all partitioning methods supported and all image types supported (in X-Ways Forensics: raw images, .e01, VDI, VMDK, VHD, and of course evidence file containers), even for images within images, also for partitions of physically attached disks formatted with a file system unknown to Windows. Access to all the files is complete read-only, mounting volumes in images or disk partitions will not change anything in the image/on the disk. To unmount a drive letter, simply invoke the mount command in any of the menus again and click

the Cancel button.

You can choose to see all existing and optionally all known deleted files from the volume in the drive letter, exactly the same files as known from the very thorough volume snapshot of X-Ways Forensics itself, which depends on whether you have refined it already or not. Optionally filtered out files can be omitted from directory listings. Child objects of files (files in files) are optionally exposed as well, presented as files in an artificial directory that has the same name as the parent file, with just a single character appended to render the name unique, as you may know it from the Recover/Copy command. By default, that suffix character is invisible, i.e. a Unicode character with no width, to make the path of the child objects look as original as possible. You may wish to replace that character with something else, e.g. an underscore, for example because you are working with an external program that is not Unicode-capable. For that you need to remove the invisible character from the edit box first, for example by pressing the Backspace key, which works even if it does not have any visible effect. After that you can insert any other character.

Previously existing items are listed optionally, and if listed, they are presented with the "hidden" attribute, so that they can be visually distinguished from existing items even in the Windows Explorer a.k.a. File Explorer. Existing files are listed optionally as well (but existing directories mandatorily, as they are potentially needed to navigate to certain previously existing files). Virtual directories are presented in the same way. (Of course, hidden files are displayed in Windows only if you choose to see them, see Tools | Folder options | View.) Virtual files in a volume snapshot as well as internal files of the file system (e.g. $MFT in NTFS and Catalog in HFS+) are included optionally, and so are original names and locations of files that that have been renamed/moved. Special objects like alternate data streams, extracted e-mails, video stills, embedded thumbnails, manual file excerpts, etc. etc. are presented in the mounted drive as ordinary files. File slack is not exposed. Files with identical names in the same directory (e.g. 1 existing, 1 previously existing file, up to 16) are not problematic with mounting. Such files can be opened from within mounted volumes through the drive letter as if they had unique names.

An option named "Apply recursively" is available to present files from all subdirectories of the currently active evidence object or the selected directory in a flat list. This is useful if you wish to use an external program to view many of the files and don't wish to bother with directory navigation. When using this option, the int. IDs of the files are inserted into the filenames to make the files better identifiable to X-Ways Forensics.

This function requires Windows 7 and later and the installation of a driver (which will be started when you use any of the mount commands for the first time) and the Microsoft Visual C++ 2013 Redistributable Package (which is not included in Windows by default and may need to be downloaded). That means that this particular part of X-Ways Forensics is not portable, but it's not a typical function for previews of live systems anyway.

Interactivity: Deleting a file in a volume mounted by X-Ways Forensics in Windows of course does not delete the file in the image or on the disk, but under Windows 7 (not later versions) can optionally trigger one of the following actions in the volume snapshot:
1) exclude the file in the volume snapshot
2) mark the file as already viewed, or
3) assign a label of your choice to the file.

The latter is very useful if you mount the volume in order to check the files for malware with an external virus scanner. Should the virus scanner delete or quarantine any of the files, X-Ways Forensics will notice that and label the file as well if selected. Note that if you manually move a file off the volume to some other drive letter this will trigger the same action, because that kind of moving is identical to copying followed by deletion. Moving a file within the same volume is not allowed.
Renaming a file in a mounted volume in Windows also renames the file in the volume snapshot. (The original name is preserved and displayed in the directory browser additionally.)

5.17 File Type Categories.txt

This customizable file defines of which file types categories are comprised. The name of a category is preceded by three asterisks and a space (***). Following is a list of file types that belong to that category, one per line. Such lines must start with either a "+" or a "-", where "+" simply means that type is checked in the file type filter. After that, typical extension for that file type follows, plus a space character, followed by a description of the file type. Only lower-case letters are to be used in extensions. The same file extension/type may occur in multiple categories (see Category column description for limitations).

Alternatively to extensions, entire filenames are supported as well. This is useful for certain files with a well-defined name whose extension alone is not specific enough or which do not have any extension. Complete filenames have to be enclosed in semicolons. Examples:
-;index.dat; Internet Explorer history/cache
-;history.dat; Mozilla/Firefox browser history
-;passwd; Existing users

There is a virtual "Other/Unknown type" category, which is not specifically defined in the file and simply covers all files that do not belong to any other, defined category.

File types are **rank**ed by importance/relevance and you may filter by this rank. For example, filtering out those file types ranked #0 will exclude font files, cursors, icons, themes, skins, clip arts, etc. Files with a low rank are of importance just in very specific investigations, for example source code, in which you would not be interested when looking for office documents or pictures for example, but definitely when hunting a virus programmer. Higher ranked file types are relevant in more cases. Generally the rank is useful in simple cases where you can expect to find what you are looking for in file types that are fairly well known. As another idea, you could make it a habit to only index files with higher ranks.

You also have the option to assign file types to a so-called **group**, a concept that is not identical to a file type category. Useful for example if your standard procedure is to let examiner A check out pictures and videos, examiner B documents, e-mail, and other Internet activity, and examiner C operating system files of various kinds, because of their specializations. You can give these groups meaningful names and filter for them, also using the Type Status dialog window. The groups are displayed in the Type filter.

All the definitions about file type ranks and file type groups are made in the "File Type

Categories.txt" file. Suggestions for ranks and an example of a group of files that may deserve special attention are already predefined. Both ranks (from 0 to 9, where missing means 0) and groups (letters from A to Z) can be optionally specified following a tab at the end of a line, in any order, for example as "2P" or "DI3". So up to 10 rank levels are possible, but it is not necessary to fully utilize this range. Up to 26 groups are possible. You do not have to start alphabetically. The case of the letters is ignored. You may also define ranks and groups for an entire category, following a tab in a category line. File types that have no rank and group inherit both from the category to which they belong.

To give a group a more descriptive name than just a single letter, insert group definition lines at the end of the text file that start with a equal sign, e.g.
=P=Photos and videos for image group
=D=Docs, e-mails and Internet
=I=File types to index

You may store additional custom definitions of file types and categories in a separate file named "File Type Categories User.txt", which will be read and maintained in addition to the standard definitions in "File Type Categories.txt" and has the same structure and is not overwritten by updates of the software if contained in the installation directory, so that you can easily continue to use it even when overwriting your installation with a new version.

5.18 Hash Database

Functionality only available with a forensic license. An internal hash database, once created, consists of 257 binary files with the extension .xhd (X-Ways Hash Database). The storage folder is selected in the General Options dialog. Such an hash database is organized in a very efficient way, which maximizes performance when matching hash values. It is up to the user to decide on what hash type the database will be based (MD5, SHA-1, SHA-256, ...), and it is up to the user to fill the hash database with hash sets and hash values, either by creating hash sets in X-Ways Forensics yourself or by importing hash sets from other sources. The same hash database can be shared and used simultaneously by multiple users/instances if the same storage folder is selected. However, it cannot be *updated* while other users/instances are using it.

It is possible to maintain two separate hash databases at the same time, databases based on the same hash type or different hash types. Useful for example if you receive hash sets from different sources with different hash types (e.g. some with MD5 and some with SHA-1 values) and wish to use them simultaneously. The second hash database may be stored on a different drive. Useful if for example the primary hash database for general use is shared with colleagues on a network drive and the user wishes to create or import new hash sets, either for temporary use only or while the primary hash database is locked by other users, into a locally stored second database.

Each hash value in the hash database belongs to one or more hash sets. Each hash set belongs to either the category “irrelevant” / “known good” / “harmless” or “notable” / “known bad” / “malicious” / “relevant” or can remain uncategorized (meaning "not decided yet" or "uncertain").

Hash values of files can be computed and matched against the hash database when refining the

volume snapshot. The directory browser's optional columns “Hash Set” and "Category” will then reveal for each file to which hash sets and category it belongs, if any (which allows you to sort/filter by these aspects and ignore irrelevant files easily or focus on files you are looking for). If the hash value of a file is contained in multiple selected hash sets, the program will report all matching hash sets and indicate the category of one of the hash sets. It also checks whether the matching hash sets all belong to the same category, and if not, will show a warning.

An optional second, separate hash database of *block* hash values (instead of normal file hash values), stored in a separate directory, allows you to search for incomplete remnants of known highly relevant files block-wise on other storage devices. You can define the block size for block hash databases yourself. 512 bytes is the default and recommended unless you are certain of what you are doing. A larger block size of 4 KB for example can be compatible with volumes/partitions that have a cluster size of 4 KB and hard disks with a sector size of 4 KB physically and logically, but thwarts any attempt to find the data that you are looking if the clusters in the target file system are not aligned at 4 KB boundaries themselves from the point of view of the evidence object. The latter may be the case for example because the file system has an irregularly sized header area before the first cluster (like FAT) or because you apply the block-wise hashing (only) at the level of a partitionable storage device in which the partitions are not aligned at a 4 KB boundary. The good news, however, is that, just like the file header signature search, block-wise hashing is applied specifically to partitions if partitions are known on a partitionable storage device (or image thereof), and only the area outside of known and explorable partitions is processed at the level of the partitionable storage device.

Via the Tools menu you get invoke the dialog window to manage the active hash database(s), which allows you to

- start a fresh, blank hash database (and discard the existing current database, using the "Initialize" command, where you have the opportunity to select a new hash type),
- view a list of the hash sets that are contained in the database,
- rename hash sets,
- merge hash sets (note that duplicate hash values in the resulting hash set are not removed immediately, but next time when you add a hash set, and note that you are not warned if you are merging hash sets of different categories),
- toggle the category of hash sets,
- verify the integrity of the hash database,
- import selected hash set text files*,
- import all the hash set text files in a certain folder and all its subfolders (ditto), optionally into a single internal hash set whose name you have to specify,
- export selected hash sets (for example if you wish to exchange individual hash sets with other examiners, not the whole database),
- and switch between the normal file hash database and the block hash database.

*NSRL RDS 2.x, HashKeeper, and ILook text files are supported, plus hash sets in the JSON/ODATA format layout as used by Project Vic (versions 1.0, 1.1, 1.2 or 2) as found in the Hubstream Inbox. NSRL RDSv3 is not directly supported, but instructions for generating a universal hash set text file from the <NSRL RDSv3> SQLite database are available. Another import and the only export format is a very simple and universal hash set text file, where the first line is simply the hash type (e.g. "MD5") and all the following lines are simply the hash values as

ASCII hex or (for SHA-1) in Base32 notation, one per line. Line break is 0x0D 0x0A.

When importing hash values from NSRL RDS 2.x, if you categorize the hash set as irrelevant, hash values marked as special or malicious will be ignored (not imported). If you categorize the hash set as notable, only hash values that are marked as malicious will be imported. If you set the hash set to the uncategorized state, only hash values that are marked as special or have an unknown flag will be imported. If you wish to import all hash values, you can import the same NSRL RDS 2.x hash set file three times, with different categorizations, and all hash values will end up in suitably categorized internal hash sets.

When importing hash values, either from an external text file with ASCII hex values or from files selected in the directory browser, you have the option to merely find out which hash values are already contained in your database and which hash values are new, without actually adding the hash values to the database. This can be used for example to find out how an import would affect your database / if there is any new material included at all etc., or if you get your hands on a list of hash values of files of interest and do not have access to the files themselves (e.g. files that once were in someone's possession) and need to find out whether they are known in your hash database.

The “Include in Hash Database” command in the directory browser's context menu allows you to create your own hash sets in any of the internal hash databases. Whenever importing/creating hash sets, duplicate hash values within the same hash set will be eliminated. When importing the NSRL RDS 2.x hash sets, X-Ways Forensics checks for records with the flags "s" (special) and "m" (malicious) so that these hash values are not erroneously included in the same internal hash set that should be categorized as irrelevant. The hash database supports up to 65,535 hash sets.

Duplicate hash values that are already contained in the hash database can optionally be either removed from a newly created or newly imported hash set or from all existing hash sets, to keep the hash database more compact/less redundant if so desired.

There is a way to efficiently delete individual hash values from an existing hash set, by importing a hash set file (simple 1-column format, 1 hash value per line), where the hash values to delete must be listed first and must be prepended with a minus sign ("-"). The file must have the same name as the existing hash set in the database that you wish to update (additional filename extension allowed).

There is an option to unload the hash database if loaded at the moment when all data windows are closed (the moment when the last open data window is closed), to save main memory or to specifically allow other concurrent users or instances to *change* the hash database.

The rather simple CRC32 algorithm is supported in ordinary hash databases. Creating a hash database based on CRC32 is useful (only) if you really only know the CRC32 values of files that you are looking for, no more advanced hash values and not the full original file contents, for example from encrypted zip archives as such archives have the CRC32 values of the unencrypted data in the metadata. If you find CRC32 matches and the file size is the same as known from the metadata in such an encrypted zip archive, then it is very likely that you have found an unencrypted copy of the very same file. If you wish to import CRC32 hash values from a text file (with "CRC32" in the first line, followed by one CRC value in hex ASCII per line), please note

that their hex ASCII values are expected in big-endian ("human-readable") byte order, as displayed in software like 7-Zip and WinZip and also X-Ways Forensics itself, which unlike MD5, SHA-1 etc. is not the byte order in which they are stored in binary, in X-Ways Forensics internally as well as in zip files themselves and presumably elsewhere.

5.19 Hash Comments

In addition to the conventional hash databases, you can maintain a database of recurring files that you have descriptions of. For example that may be useful if you are required to include descriptions of illegal photos in your case reports for the court. If the same photos occur in multiple cases, the new database can save you work and make it unnecessary to view the photos again. Whatever you enter as comments can be saved in the database along with the corresponding hash value. For that to happen you select the relevant files and invoke the command "Include in Hash Database" in the directory browser context menu. Whether hash values were already computed for the selected files is not important. They are computed on the fly if not. You can get the same comments back in another case if you match the hash values in the other case against the database as part of volume snapshot refinement.

The database is stored in the file "Hash Comments.txt". You can easily share the database by simply sharing that file with other users. The file is independent of the conventional hash databases, meaning it does not matter which user has which conventional hash database with hash sets from which source(s). You do not need a conventional hash database at all to create a "Hash Comments.txt" file or match the hash values in your cases against the "Hash Comments.txt" file of someone else. So the "Hash Comments.txt" is quite universal and suitable for inter-agency exchange.

You can merge text files of different colleagues/sources with your own database in the user interface: Open the Tools | Hash Database dialog window and click the Import button. If X-Ways Forensics detects duplicate entries (same hash value), it will either keep the previous comment or adopt the new comment, depending on the state of a checkbox in the same dialog window. Keep that in mind when importing entries from other users. The rule also has an effect if duplicate entries are found within the same text file because you have merged entries manually.

Since we are talking about a simple text file, you can merge "Hash Comments.txt" files from different sources easily in a simple text editor, or edit the descriptions as needed, get them automatically translated etc. Just keep the general layout of 1 hash value + description per line intact. The first line (header line) in "Hash Comments.txt" must contain the designation of the hash type in ASCII (e.g. "MD5" or "SHA-1"), followed by a tab and the ASCII letters "Cmt", and this is all case-sensitive. All the following lines start with a hash value in hex ASCII (both upper or lower case allowed), followed by a tab and the description in UTF-8. Both Windows and Unix/Linux line breaks are allowed.

There is an unlabeled, but tooltipped checkbox that allows you to get existing comments on files replaced when successfully matching hash values against hash comments. That means previous comments will be lost if there is a comment for the same files in the hash comment database. There is also an option to prepend comments that were automatically derived from "Hash

Comments.txt" with the initials "[HC] " to distinguish them from comments entered by the user manually.

5.20 PhotoDNA

For licensing reasons the PhotoDNA functionality is made available as a separate download, and provided by X-Ways itself **only to law enforcement agencies**, which may use it to prevent the spread of child sexual abuse content and for investigations targeted to stop its distribution and possession. For details about PhotoDNA please see this high level technical explanation and this press information.

X-Ways Forensics can apply the PhotoDNA hashing algorithm to photos. Thanks to the robustness of the hash algorithm and its specialization in photos, it usually allows to automatically recognize known photos even if they have experienced lossy compression repeatedly (e.g. JPEG), if they have been stored in a different file format, resized, partially blurred/pixelated, color-adjusted or contrast-adjusted etc. Unlike hash values computed by conventional general purpose algorithms, PhotoDNA hashes are resistant to various such image alterations or change only slightly. Optionally, known photos can be recognized even if they were mirrored (flipped horizontally). To avoid loss of time with small irrelevant pictures, PhotoDNA is not applied to pictures that are less than 50 pixels wide or tall.

If the PhotoDNA functionality is present, a database with PhotoDNA hash values of photos can be created and maintained within X-Ways Forensics, and photos may be matched against that hash database in X-Ways Forensics and X-Ways Investigator to automatically identify known incriminating content.

Law enforcement agencies may want to create and share their own collections of such hash values, based on pictures from previous cases, or import an extensive existing collection from Project Vic (JSON/ODATA format layout version 1.0, from v18.1 of X-Ways Forensics also version 1.1, from v18.2 of X-Ways Forensics also version 1.2). You can also import PhotoDNA hash databases of other X-Ways users (select the “RHDB” file!), you may delete hash categories that you don't need any more, and you may merge or rename categories in your database. When importing someone else's hash database, their categories of the same name will be merged with yours. PhotoDNA hash values may also be imported if they are stored in text files, with "PhotoDNA" in the first line, followed by 1 hash value per line in hex ASCII or Base64.

Hash values of pictures in the volume snapshot of an evidence object can be added to the PhotoDNA hash database in the same way as conventional hash sets are added to a conventional hash database, using the “Include in Hash Database” command in the directory browser context menu. The database is one of the several databases that can be managed with the Tools | Hash Database command. The PhotoDNA hash database is stored in a directory next to hash database #1.

When importing PhotoDNA hash collections or when including the PhotoDNA hash values of selected files into the database directly in X-Ways Forensics, the additional entries are checked for redundancies and conflicting categorizations among each other and with existing entries in

the database, to keep the database as small, fast and useful as possible. This is recommended, but optional, and if you skip this step and if the data set is very large, you potentially save hours of time, at the cost that matching pictures against the database during volume snapshot refinement will take more time, and that for variations of the same picture you may get different classifications returned. You may define the import strictness separately to define how similar hash values have to be to warrant a re-classification of existing values (to keep the database consistent) and to define how similar hash values have to be to overwrite (replace) an existing value with a new value (to keep the database compact and less redundant). The latter strictness must not be less than the former. A hash value can be either an existing, old value in the database, a new hash value in the database added by the current import operation, or a pending hash value that is yet to be added to the database.
1) If a pending hash Y is absolutely identical to an old or new hash X, Y will be ignored and not added to the database. If Y and X are just similar, Y will be added. If Y and X are almost identical, X is directly replaced (overwritten) with Y.
2) If Y and X are identical or similar and, but belong to different categories, and X is new, that means that the quality of the import file is low. You will see a warning. If the import is from a ProjectVic hash collection, and the two categories are the relatively similar categories "child abuse" and "child exploitation", no special action is taken. If the two categories involved are not those two: If either X or Y belongs to the category "non-pertinent" and the picture is a largely monochromatic picture, X will be assigned to the category "non-pertinent". Otherwise the categorization conflict will be resolved by assigning X to the category "uncategorized".
3) If Y and X are identical or similar, but belong to different categories, and X is old, X will be assigned to the same category as Y, assuming that the previous categorization is wrong or outdated and the import file contains correct/new information. This is beneficial for example for entries whose original categorization is from a foreign source (e.g. Project Vic) and which needs to be adjusted because of different legislation or jurisdiction in your country or simply because of categorization errors or different interpretations. What is considered child pornography in one country is not necessarily classified as such in another country (example: computer generation images, animation). Recategorization requires that you have copies of the same pictures (not necessarily the exact same files) in your collection or know which hash values belongs to which picture exactly.

Standard Project Vic categories for the USA are predefined in the user-editable text file PVicCat.txt. Law enforcement users from UK and Canada can download their definitions from the PhotoDNA download section on our web server and replace the default PVicCat.txt file in their installations. Users in other countries with differing categories can gladly share them with us.

When adding PhotoDNA hash values to the internal PhotoDNA hash database with the “Include in Hash Database” command, you have the option to store your comments about the selected files in that hash database as descriptions. These descriptions can be automatically adopted as comments again next time when the same pictures are found in another case. They can either replace existing comments in the other case or (if the corresponding check box is half checked) be appended to existing comments. This is very useful for example for police investigators who are required by the court to provide a textual description of each and every child pornography picture, to at least spare them the work of entering descriptions of the same known pictures more than once. Also useful to store information such as known identities of the persons in the photo, previous case numbers etc., for future reference if the same photos are found elsewhere. The

descriptions in the hash database can be updated with your comments by simply adding the PhotoDNA hash values of the same files to the internal database again through the “Include in Hash Database” command. When you import a colleague's internal hash database (by selecting their RHDB file), be sure to have not only the corresponding RHCN file (with the category names) present in the same directory, but also the new subdirectories that contain the descriptions, if any, if you wish to import these descriptions.

To delete all internal descriptions, you can simply delete the D* subdirectories of the PhotoDNA hash database directory. Or if you wish to share your database with other users without the descriptions, simply do not include the D* subdirectories. You may also manually delete or update any individual descriptions in the text files in the D* subdirectories at any time. Descriptions that you already have in your database will not get lost if you import hash values of the same pictures again from other sources, except they will be overwritten if that other source is a PhotoDNA hash database of X-Ways Forensics that has descriptions of the same pictures.

When creating a PhotoDNA hash set of selected pictures, you may choose to not add the hash set into the internal database, but create a separate plain text file with PhotoDNA hash values instead. For that, please check the "Save as..." box. Such files can be passed on to other users if they wish to add the specified hash values to their databases or remove them (see above).

It is possible to cleanse a PhotoDNA hash database from unwanted hash values. The hash values to remove are provided as a plain text file, with 1 hash value in hex ASCII notation per line and "PhotoDNA" in the first line. The specified hash values match exact equivalents contained in the hash database and also small variations (same deviation permitted as set for matching). It may become necessary to cleanse a PhotoDNA hash database if you have imported hash sets from a foreign source whose contents partially do not meet your requirements, which becomes apparent when you get false hits, if you do not wish to remove the entire hash set, or if you have accidentally included a wrong picture in your hash database yourself.

There is a button that allows to export selected hash collections into text files to share them with other users or to check which hash values are contained/which ones were deduplicated etc. Another function (the button with the magnifying glass) will help you to check the database for the presence of a specific hash value, specified in Hex ASCII or Base64 notation. If there is a hit, you will be shown the name of the hash collection that contains the hash value. If the matching entry in the database has a textual description, that description will be shown as well. Up to 19 matches are returned, and for each you will see how precise the match is (the higher, the more precise; same basic scale as the user-specified strictness for matching, i.e. level 1 means very rough match). You have the option to narrow down the result list to more precise matches by enforcing a higher minimum strictness level, which is useful if there are more matches than can be listed.

There is a function to mark selected PhotoDNA categories as "preferred", with a black star. That way they will get priority if for a picture in the volume snapshot matches are found with hash values in different categories. Such preferred categories will be reported as a match even if alternative matches with non-preferred categories are much closer matches. That is useful for example if you have categories in your database that you trust to be accurate and suitable and others that you trust less, for example because they are known to contain errors (e.g. the same picture classified as CP and non-pertinent at the same time) and/or because they are from a

foreign source and based on different laws and jurisdiction.

Matching is part of the "picture analysis and processing" operation in Specialist | Refine Volume Snapshot. If there matches for the same picture in different categories of the PhotoDNA hash database, you can see that in the directory browser: The name of the category with the closest match is shown, followed by a comma and an ellipsis. In rare cases where this happens it can be important to review the picture manually and make the final decision about its relevance for the case. You can also filter for pictures that were found in more than one category. Such pictures may deserve as much attention as duplicates in conventional hash databases that belong to the "irrelevant" category and "notable" category at the same time and are usually the result of an inconsistently populated database, e.g. accidental miscategorizations or correct categorizations made by users in different jurisdictions etc. If the returned best matching category for a picture is wrong in your opinion, you can fix this by adding the PhotoDNA hash value of that picture to the PhotoDNA database again, specifying the correct category.

5.21 FuZzyDoc

The so-called **FuzZyDoc™** technology can help you to identify known documents (word processing documents, presentations, spreadsheets, e-mails, plain text files, ...) with a much more robust approach than conventional hash values. Even if a document was stored in a different file format (e.g. first PPT, then PPTX, then PDF), it can still be recognized. Internal metadata changes, e.g. after a "Save as" or or after printing (which may update a "last printed" timestamp), do not prevent identification either. Very often even if text was inserted/removed/reordered/revised, a document can still be recognized. This is achieved by using fuzzy hashes.

FuzZyDoc hash values are stored in yet another hash database in X-Ways Forensics. Hash sets based on selected documents can be added to the FuzZyDoc database exactly like hash sets can be created in ordinary hash databases, and the FuzZyDoc hash database can also be managed in the same dialog window as the other hash databases. For each selected document you can create 1 separate hash set, or you can create 1 hash set for all selected documents. Up to 65,535 hash sets are supported in a FuzZyDoc hash database. You have the option to export, import and merge FuzZyDoc hash sets. The result of the export can be used with the import function or alternatively is also valid as a stand-alone database by itself.

FuzZyDoc is available to all users of X-Ways Forensics and X-Ways Investigator (i.e. not only law enforcement like PhotoDNA). FuzZyDoc should work well with documents in practically all Western and Eastern European languages, many Asian languages (e.g. Chinese, Japanese, Korean, Indonesian, Malay, Tamil, Tagalog, ..., but not Thai, Divehi, Tibetan, Punjabi, ...), and Middle Eastern languages (e.g. Arabic, Hebrew, ..., but not Pashto, ...). Note that numbers in spreadsheet cells are not exploited by the algorithm, only text. Note that only files with a confirmed or newly identified type will be matched against the FuzZyDoc hash database. For that reason, file type verification is applied automatically when FuzZyDoc matching is requested.

Documents whose contents are largely identical (e.g. invoices created by the same company with the same letterhead) are considered similar by the algorithm even if important details change

(billing address, price, product description), depending on the amount of identical text. That means that if you have 1 copy of an invoice of a company, matching against unknown documents will easily identify other invoices of the same company. For every document that is matched against the database, up to 4 matching hash sets are returned, and the 4 best matching hash sets are picked for that if more than 4 match. For every matching hash set, X-Ways Forensics also presents a percentage that roughly indicates to what degree the contents of the document match the hash set. Two different percentage types are available. A percentage based on the total text in the processed document gives you an idea of how much of the text in the document is known/was recognized, whereas a percentage based on the text represented by the hash set gives you an idea of how closely a document resembles the original document that the hash set is based on (makes sense only if you generate 1 hash set per document, i.e. do not combine multiple documents in 1 hash set). The matching percentage does not count characters one by one, and it works only on documents that actually make sense, not on small test files that only contain a few words.

Before matching files against the FuzZyDoc hash database (a new operation of Specialist | Refine Volume Snapshot), you can specify which types of files you would like to analyze, and you can unselect hash sets in the database that you are temporarily not interested in. Note that processing less files (e.g. by specifying less file types in the mask) of course will require less time, proportionally, but selecting less hash sets for matching as such does not save time. You may specify a certain minimum percentage that you require for matches (15% by default) to ignore insignificant minor similarities. That option is not meant to save time either.

In order to re-match all documents in the volume snapshot against the FuzZyDoc hash database, please remove the checkmark in the "Already done" box first. Otherwise the same files will not be matched again, for performance reasons. Re-matching the same files may become necessary not only if you add additional hash sets to your FuzZyDoc database, but also if you delete hash sets, as that invalidates some internal links (if that happens, it will be shown in the cells of the result column).

Matches with the FuzZyDoc database are presented in the same column as PhotoDNA matches and skin color percentages, called "Analysis". A filter for FuzZyDoc matches is available. FuzZyDoc should prove very useful for many kinds of white collar crime cases, most obviously (but not limited to) those involving stolen intellectual property (e.g. software source code) or leakage of classified documents.

5.22 Text Decoding and Spreadsheet Processing

Part of volume snapshot refinement and logical searches.

The text that is contained in files whose format is supported by the viewer component, e.g. PDF (Adobe), WPD (Corel WordPerfect), VSD (Visio), SWF (Shockwave Flash), can automatically be extracted/decoded/decompressed, resulting in unformatted ASCII or UTF-16 plaintext, which can be stored in the volume snapshot, indexed, previewed in Preview mode or in the Comments column, and reliably searched in addition to the original data itself. Search hits might otherwise be missed because various file types typically or at least sometimes store text in an encoded, encrypted, compressed, fragmented or otherwise garbled way. Important: In particular for

HTML, XML and RTF documents as well as e-mail messages, which may employ various methods of encoding (e.g. UTF-8) non-7-bit-ASCII characters (e.g. German umlauts or Chinese characters), decoding may be useful, depending on the language of your search terms/the characters contained in your search terms. When you specify a file mask for decoding, that mask will not only be applied to the names of searched files, but also to their true type if verified by signature (see Refined Volume Snapshots). This feature requires the separate viewer component to be active for the decoding and text extraction part.

The decoded text as well as text derived via OCR is output in Latin 1 or Unicode, and can optionally be buffered (stored in the volume snapshot) to allow for a helpful context preview for search hits in the decoded text and to accelerate future searches. Text extracted from documents or pictures in Preview mode can be optionally stored in the volume snapshot as well. This option is remembered separately just for Preview mode and disabled by default, so that you can experiment with different OCR settings and different PDF decoding settings and see fresh results instead of always the same text as stored in the volume snapshot after the first attempt. To access the Decoding/OCR settings specifically for Preview mode, you can right-click the Text/OCR submode button.

The default file mask for this option is *.pdf;*.docx;*.pptx;*.xlsx;*.odt;*.odp;*.ods;*.pages;*.key;*.numbers;*.eml;*.wpd;*.vsd;*.onepkg;*.json. It is recommended to add ;*.html;*.xml;*.rtf depending on the characters searched for, and more depending on your requirements. For example *.doc might be a good idea if you want to be very thorough because text can be fragmented or change from one character set to another abruptly in the middle of a MS Word document. Just keep in mind that the additional decoding and search result require more time and likely result in duplicated search hits (search hits found in both the original format and the result of the text extraction). E-mails will generally not be decoded by X-Ways Forensics when only 7-bit ASCII characters are search. The file mask is applied to both the filename and the detected true file type. To see what text is extracted from a document by this function, you can select the document in the directory browser in Preview mode and hold the Shift key when switching to Raw mode.

X-Ways Forensics has the ability to find *numbers* and *dates* not only if stored literally as text, but also if numbers or dates are stored in binary form in certain spreadsheet files (e.g. in OLE2 compound file format) or in some other encoded form (e.g. dates encoded as textual integer numbers in XML), if the "decode text" option is on and if "Convert binary storage of numbers/dates in spreadsheets to text" is checked. However, this is slower than the regular text decoding. Works pretty well with numbers in Excel and LibreOffice Calc spreadsheets, but can be tricky occasionally with the format of dates if the original Excel user has selected a custom date format instead of one of the standard date formats and also because of some specialties with certain Calc files where it's not 100% predictable that a date will be extractable in the expected format. This kind of search likely works with some other file types as well, e.g. older spreadsheet types like MS Works or Lotus 123. You can try and define the file types in Options | Viewer Programs if needed. To quickly see and double-check the extraction of numbers and dates from a particular file of interest, you select that file in the directory browser and switch from ordinary to raw preview mode with the Shift key pressed. Please feel encouraged to completely remove that file mask there for faster text decoding if you do not need to search for numbers and dates in

spreadsheets.

Some more details about number searches: Consider a cell in an MS Excel spreadsheet that contains the number 1234567. You can find that number with the Simultaneous Search searching simply for "1234567" (without the quotation marks). Even if you just know part of the sequence of digits and search for "34567", you will get a search hit (unless the "whole words only" option is on). If the cell has the "number" format (not "general"), with digit grouping enabled, you can optionally get the number with digit grouping when the file is searched/indexed/decoded in that volume snapshot for the first time, using the digit grouping symbol that is defined in X-Ways Forensics in Options | Notation, but that is not generally recommended because you would have to search for the same number both with and without the grouping symbol if you don't know whether the original spreadsheet cells were formatted as "number" with or without digit grouping or as "general". Anyway, to give you another example, if you enable that option for digit grouping in number cells in Options | Viewer Programs and you live in an English speaking country, using a comma as the digit grouping symbol, you would thus search for "1,234,567" to find that number in a number cell. You can also search for just ",567" to find the digit group "567" at the end or in the middle of any longer number in that notation.

If the number that you are looking for is a floating point number, the same rules apply, and you can optionally enter the number with as many decimals as you expect to be visible in the cell in the original application (or less), with the same decimal symbol as in your notation settings in X-Ways Forensics (either a point or comma). If a floating point number is stored for example as 9.876 and formatted to show 2 decimals, it will be shown rounded as 9.88 in the original application and will also be searchable like that in X-Ways Forensics. The same rules apply to currency amounts. You can append or prepend the currency symbol if you know for sure that it was shown in the original formatting, and how (e.g. with or without space between currency symbol and number), or you just omit symbol.

You can search for dates in pure date cells using the notation that is active in X-Ways Forensics as the so-called simple date format. If your simple date format is MM/dd/YY, you would search for 12/31/19 to find the date Dec 31, 2019. Partial date searches are also possible, and make sense especially if you do not use American date styles. For example in ISO notation "yyyy-MM-dd" you can search for "2019-07-". Or in German notation "dd.MM.yy" you can search for ".07.19" to find any date in July 2019. Pure time cell searches are also possible (with partial or whole time expressions). Just make sure to use the separator that is active in X-Ways Forensics for the display of times. Searches for combined date and time values are supported, however, the delimiter between date and time that you can expect is not the delimiter defined in Options | Notation, but typically a single space, or an individual delimiter defined by the user of the spreadsheet.

If an Excel worksheet is embedded in a .docx, .pptx, or .odt file and the volume snapshot has been sufficiently refined, the worksheet will be processed and searched in the same way as if it was a separate file. If embedded in a .doc file, you would get a notification in the form of a label "Contains embedded document(s)", which is often useful to check manually anyway. The number search capabilities should prove very useful especially in forensic accounting, tax fraud investigations etc. Please note that the simple search function of the viewer component (Ctrl+F) in ordinary ("pretty") Preview mode or the View command cannot find numbers or dates in spreadsheets, no matter how you type them.

By default, a newer alternative method to extract data from spreadsheets as text is active. That option improves the fidelity of the extracted text in terms of cell order and arrangement, normalizes the formatting of date cells in the decoded text to the notation that is active in X-Ways Forensics for more reliable search results, reliably includes hidden cells, better supports very wide sheets and conveniently marks the boundaries between worksheets with separator lines. If you need to preserve characters that your active Windows code page does not support (e.g. Chinese characters on a typical computer in America or Europe) because you are going to search for them, you need to check one extra box ("Must support Unicode"), and with that option this method will require usage of the Windows clipboard.

Text in PDF files from certain sources cannot easily be decoded. It may be output incompletely or garbled or as total gibberish. Whenever in a real-life scenario you come across a series of uniform PDF files with that problem (generated by the same mechanism for the same purpose, e.g. bank account statements, invoices, product specifications, ...), so that their decoded text is not legible and searchable/indexable, you can add their creator name, producer name or generator signature to a list that X-Ways Forensics checks before decoding PDF files. If is a match with either of these properties, X-Ways Forensics will apply OCR to such files rather than attempt (presumably futile) text decoding. You can find this special option in the dialog window with the decoding settings. This is a rather technical option and therefore not available in X-Ways Investigator. Without that option, the only situation in which a PDF file is OCRed is if no text can be extracted from it at all, just like in all previous versions. The list is maintained in a file named "PDF Requiring OCR.txt" and can easily be shared with other users. The format is explained in the text file itself when it is created. It is expected in the same directory where your WinHex.cfg file and various other user-editable text files are. The generator signatures, creators and producers of PDF files can be found in and copied from Details mode. For the generator signature only the 8 hexadecimal digits are required.

Concurrent scrolling through pages of multiple PDF documents for OCR in multiple threads is optional and disabled by default. Disabling it can yield more complete results for certain documents that are slow to render.

5.23 OCR (Text Recognition in Pictures)

The OCR capabilities of the software package Tesseract can be utilized from within X-Ways Forensics and X-Ways Investigator. The package can be downloaded from our web server. Updated download instructions are available from the same place as always. If Tesseract is found in the subdirectory \Tesseract of the installation directory when X-Ways Forensics is first run, Tesseract will be activated automatically. Otherwise please go to Options | Viewer Programs to indicate the path.

OCR can be applied as part of logical searches or indexing to suitable files such as document scans or digitally stored faxes in TIFF format or PDF documents that contain only graphic content. The file mask for that is matched against the filename as well as Type column (which is quite reliable and standardized after file type verification). By default, it includes even *.jpg,

however, whether applying OCR to every JPEG file in a case is a little excessive or necessary is up to you to decide, and you have full control over the scope of the search using various means anyway. Please be aware that high-resolution photos cost a lot of time to check of text. Digital photos in JPEG and HEIC format will be rotated according to the instructions in the Exif metadata to restore the correct orientation and thus hopefully allow OCR of text that was originally photographed roughly horizontally. If the ordinary text decoding is already successful for a given file of a type that is contained in both file masks (*.pdf), OCR will not be applied additionally. The option "Store decoded text for context preview and future searches" will also keep text derived from OCR stored in the volume snapshot.

All hits returned by the logical search in OCR-derived text are identified as such in the Descr. column and highlighted in a different color. The Descr. filter allows you to list only such OCR search hits or not OCR hits. Older versions of X-Ways Forensics can see OCR search hits when opening the same case, but won't know that they are OCR search hits.

You can select up to two languages for text recognition at the same time, after clicking the ... button for this in Options | Viewer Programs. However, there is a trade-off if you select Chinese/Japanese and a Western language at the same time. This will deteriorate the recognition of the Asian characters. You may want to select *only* Chinese/Japanese for much better recognition in that language. English (actually Latin) letters can still be recognized in that case, even if English is not expressly selected, at reduced quality. Select both Chinese/Japanese and a Western language at the same time only if correct recognition is more important to you in the Western language.

Preview mode now has a separate submode in addition to Raw submode, called Text mode, in which pure text from non-picture files is extracted, just like for the logical search with the decode option. That submode can also be useful to better understand how text is extracted from various document types, in particular from spreadsheets, for which different extraction options exist that may differ in output, especially in formatting.

If the ordinary text extraction/decoding in Text submode does not return any result or if the previewed file is a picture file, and if Tesseract is available and active, OCR will be applied. This allows you to better understand how well OCR will work in searches for the kind of files that you are dealing with. You can also experiment with different languages selected and compare the quality of the results. The submode button is named "Text" by default, but will change its label to "OCR" to make you aware that OCR is or was employed to retrieve the text. OCR can be time-consuming for multi-page TIFF and PDF files, but can be interrupted by the user if necessary. If a logical search or indexing has applied OCR to a file before and the result was stored in the volume snapshot, then the OCR-based preview will be available instantly and OCR will not be re-applied from scratch.

Both submodes Raw and Text in Preview mode remain active until you leave Preview mode or select a file of a different type. If you prefer to make either of these submodes more persistent, so that it remains active even when previewing files of different types, you can hold the Shift key while clicking the respective submode button.

The Tesseract package that is downloadable from our web server already has support for the following languages integrated, in alphabetic order:

ara: Arabic
chi_sim: simplified Chinese (horizontal writing only)
chi_tra: traditional Chinese (horizontal writing only)
deu: German
eng: English
fra: French
heb: Hebrew
ita: Italian
jpn: Japanese (horizontal writing only)
kor: Korean (horizontal writing only)
nld: Dutch
pol: Polish
rus: Russian
spa: Spanish
swe: Swedish
tur: Turkish
Other languages can be added if you can find .traineddata files for them at https://github.com/tesseract-ocr/tessdata_fast. Such files simply need to be put into the \tessdata subdirectory of Tesseract. Or you can visit https://github.com/tesseract-ocr/tessdata_best to download higher quality OCR engines for any of the supported languages. (Please note that OCR takes considerably more time with them.)

Supported file types are generally the following: PDF, PostScript (PS), TIFF, JPEG, HEIC, PNG, GIF, BMP, non-animated WEBP, AutoCAD DXF, Photoshop PSP, and maybe more.

5.24 Picture Content Analysis with AI

Part of volume snapshot refinement. Available in X-Ways Forensics only. Requires a 64-bit Windows 10, Windows 11, Windows Server 2016, Windows Server 2019 or Windows Server 2022. Excire Forensics provides these three distinct powerful capabilities (locally, not in the cloud!):

• It automatically analyzes photos and recognizes image content, objects like specific kinds buildings, vehicles, animals and plants, beaches, mountains, people of different ages, nudity and pornography, guns, drugs, text, ... (complete list). It also detects color themes and general photo properties. The results are output as labels or comments, so that you can focus on photos with content that is relevant in your case (combined with AND or OR) or filter out photos with irrelevant content or properties. If output as labels, videos also get labeled automatically if the stills that were extracted from them are processed.

• It allows you to find photos that are “similar” from the perspective of an artificial intelligence to a collection of typical relevant photos from earlier cases or other photos that you provide (in JPEG, PNG, Bitmap, or TIFF format, 224x224 pixels minimum).

• It allows you to find faces of particular people in photos of new cases. X-Ways Forensics will require you to mark faces of interest in JPEG, PNG, Bitmap, or TIFF pictures that you provide in

a dedicated directory. You may press Esc in the face marking process to abort processing.

X-Ways Forensics processes pictures with Excire in one of the following formats: JPEG, PNG, Bitmap, TIFF, non-animated WEBP, GIF, HEIC. You can decide which minimum resolution to require. The number of misidentifications increase in pictures with lower resolution (such as thumbnails).

Pictures can be automatically categorized as irrelevant or notable. In the extensive hierarchy of identifiable objects you can select individual objects or entire subtrees that render a picture irrelevant from your point of view with a high degree of certainty, such as any kinds of animals, plants, sports, musical instruments etc. For dominant colors the required confidence is actually the percentage of pixels with that color, so unless a picture is monochromatic, typical values are lower than the confidences of other detected picture content.

You can also define what renders a picture notable for you, such as nudity ("act"), children, vehicles, text etc. "Notable" always overrides "irrelevant" when in doubt, if for example dogs are considered as important in a particular case, but animals otherwise are still marked as irrelevant. Logical AND combinations are supported when categorizing photos as notable. Some AND combinations are predefined that are meant to assist in child pornography investigations. The computed relevance of picture files can be adjusted based on what content is detected that you consider relevant or notable, either strongly, moderately, slightly or not at all. It is also adjusted generically based on some properties detected by Excire.

You may be able to find additional use cases for Excire if you have it process known relevant pictures in previous cases. For example it may turn out that photographed documents that were crumpled or handwritten are still detected as “text” and/or “paper texture” even if OCR cannot recognize text any more and only the human eye can still read something.

Photo content descriptions are available in these languages: English, German, French, Spanish and Italian.

5.25 Time Zone Concept

The following applies to WinHex with a specialist license or higher as well as X-Ways Investigator and X-Ways Forensics.

X-Ways Forensics employs its own, not Windows' logic to convert UTC timestamps to a freely chosen time zone for display in the directory browser, in report tables and exported lists. It displays timestamps independently of the time zone selected in the examiner's system's Control Panel. The display of timestamps in X-Ways Forensics may differ from Windows because in Windows a timestamp in daylight saving time is not displayed based on daylight saving time if daylight saving time is not active when looking at that timestamp.

When working with a case, the time zone selected for that case applies globally to the entire program (selectable in the Case Properties), otherwise the one selected in the General Options dialog. When working with a case, optionally it is possible to specify different time zones per

evidence object, so that you can always see local filetimes even for media that were used in different time zones, if preferable. Note that the timestamps are converted for *display* only. That means, in a recursive view in the case root that covers multiple media, *sorting* is based on absolute UTC timestamps. Optionally, the actually used conversion bias can be displayed as well (see directory browser options).

Timestamps on FAT volumes are never converted as they are not available in UTC, but based on one or several unknown local time zones. Timestamps in file systems that store the time zone explicitly are converted to UTC internally and then for display purposes from UTC to a local time zone.

The time zone definitions can be adjusted, if necessary. Please note that changing these definitions in any dialog window affects the definition of time zones throughout the program.

The standard Windows conversion technique, which depends on the time zone selected in the user's system's Control Panel, is still employed…

- in File | Properties, where the timestamps of files on the user's own system can be accessed/changed,
- for the case logging feature,
- generally when operated without a specialist or forensic license, and
- when operated without the file “timezone.dat”.

You can tell that either of the latter two is true if the “Display time zone” button in the General Options dialog is grayed out or not visible.

5.26 Evidence File Containers

Only available with a forensic license. The Specialist menu allows to create a new file container, open an existing one, and close the active file container. The directory browser context menu allows to fill it with selected files.

When you need to pass on a collection of selected files (even from different evidence objects) that are of particular relevance to a case, to other persons involved in that case, e.g. specialized investigators, who do not need to or must not see irrelevant files, evidence file containers may come in handy. Most file-system level metadata (name, path, size, attributes/file mode, timestamps, deletion status, classification as alternate data stream or virtual file or e-mail message or attachment, …) and especially the contents of the file are fully retained in an evidence file container. Also when a conventional (physical, sector-wise) image is overkill because you need to acquire only selected files and not entire media, containers are recommended.

Evidence file containers either use a special file system (similar to NTFS) that can accomodate most metadata from conventional file systems of the Windows, Linux, and Apple world, and that is the regular and recommended choice, or they can be constructed as regular Zip archives. Many of the advanced functions, properties and settings of file system based evidence file containers as discussed below are not available with Zip archives, but using Zip archives has the following

advantages: The file contents can be encrypted immediately, which can be useful not only to prevent them from being read by unauthorized people, but also to prevent virus scanners from detecting malware that you intentionally wish to pack up/quarantine in a dedicated archive, for example with a password like "infected". This function is also called "secure export". Ordinary tools that do not understand file systems at a computer forensic level as required for regular evidence file containers may be able to read zip archives and allow to view the included files. That includes the Windows File Explorer (which, however, does not support Zip archives with AES encryption). Zip archive containers can be re-opened for further filling only if they are not encrypted.

Evidence file containers can be interpreted, added to a case and conveniently examined like other image files, and in particular also in X-Ways Investigator [CTR], the simplified version of X-Ways Forensics for investigators that are not computer forensic examiners, but specialized in other areas such as corruption, accounting, child pornography, building laws, ... The recipient of the container can add the container to his or her own case, view the files that it contains just like in a disk partition or a conventional image, can run keyword searches, comment on files, assign labels, create a report, etc. Labels can even be exported and imported back into the original case, via case tree context menu commands. This allows to split up the workload in large cases across multiple investigators who work simultaneously and to reconcile their results.

The file system in evidence file containers can be understood by certain computer forensic tools other than from X-Ways. Older versions of WinHex (with a specialist license or higher), X-Ways Forensics and X-Ways Investigator can also understand them. They can all read the contents of all files and show the most essential metadata (e.g. filename, path, many attributes, most timestamps, existing or deleted). To see the maximum amount of metadata, however, please use WinHex/XWF/XWI 16.3 and later. More information. A raw (non-.e01) evidence file container can be interpreted and mounted as a drive letter in WinHex with any license type, to render the files accessible in other tools if those other tools do not understand the container format natively. (If such a container contains no more than 1,000 objects, then even the evaluation version of WinHex can do that.)

Containers can theoretically hold around 1 billion files. X-Ways Forensics automatically prevents that the same file is copied to the container twice (a particular file at a particular location). If you wish to check the contents of an evidence file container while you are filling it, that is no problem. You can tentatively add it to the same case as an evidence object while it is open for filling. You do not need to remove it from the case or close the evidence object in order to fill the container further. After every filling step, you can take a new volume snapshot of the container to see the complete up-to-date contents. And when done filling the container, you can remove it from that case as it is probably no longer needed in there.

In order to identify/preserve the source of files that originate from different evidence objects, the names of these evidence obejcts can be included in the container as the top directory level. If the option to insert an artificial top directory level is only half selected, that means that only the the names of partition evidence objects are included that have a physical evidence object as a parent. Useful if the parent evidence object name is very long and redundant to include because you will fill your entire container only with files from that physical evidence object and will reference that object's name in the container name already.

When creating a container, you chose between a direct method and an indirect method to fill it. Indirect means via your own hard disk, i.e the contents of files are not copied directly into the container, but to your folder for temporary files first (cf. General Options), and only then from there into the container. This can be beneficial because it allows a resident antivirus software to intercept these files (check them for viruses, disinfect/disarm them, rename them, move/delete/lock them, etc.), so that it prevents viruses from making it into a container. The resulting container is free of known viruses (depending on the antivirus software in use) and can reasonably be passed on to and used in an environment with higher sensitivity, higher security requirements, and/or less sophisticated virus protection. **Important**: Please verify first, by testing with known malware, that your antivirus software works as intended in this situation.

An optional internal designation can be specified (up to 31 characters), which will become the volume label of the XWFS file system. An optional description can also be specified (up to 60,000 characters), which will be imported as the evidence object comments once the container is added to a case in X-Ways Forensics. The description stored in the container can still be added or edited later.

Files selected in the directory browser can be added to the container that is open in the background with the directory browser's context menu. Either you copy the logical contents of a file, the logical contents and the file slack separately, just the slack, only the block selected in File mode, or merely the file system level metadata of the file. You may also specify whether child objects of selected files should be copied to the container as well, even if they are not selected themselves, either child objects of any kind of child objects (if fully checked) or only e-mail attachments (if half checked).

Optionally containers can include the data/contents of directories themselves, i.e. depending on the file system, directory entries, INDX buffers, etc. Useful if the recipient of the container is technically versed and might be interested in timestamps or other metadata in these data structures. If you choose to include directory data in a container when creating it, this has a direct effect only on directories that are selected themselves. It has an effect on the respective *parent* directory of selected items only if you enable an additional option (“Include data structures/contents of direct parent items”). This additional decision is needed because otherwise the directory data might unintentionally reveal the names and other metadata of files that were intentionally omitted from the container, e.g. for reasons of confidentiality.

You can include objects in the container with or without their original path. If this option is only have selected, that means only a partial path is included, downwards from the directory from which you are copying / which you have explored, a behavior that is intuitively understandable because this is how Windows File Explorer copies selected files and directories. If in the container you have X-Ways Forensics recreate the original path of files that are child objects of other files, then those parent files will be included in the container at least as nominally, without data, so that the child object appears with the correct path and it is clear where it comes from, just by looking at the container. Examples for such parent files are the e-mail message that a selected attachment belongs to, the zip archive that contains a selected file, and the document that a selected picture is embedded in. With the option “Include data structures/contents of direct parent items”, the data of such files is also included in the container, even if these files were not selected for copying themselves. Artificial directories can be optionally created in containers to accommodate child objects of files, for compatibility with tools that do not accept files as child

objects of other files. WinHex/XWF/XWI do not need such artificial directories.

Any file that is part of a volume snapshot (e.g. even individual e-mail messages if extracted) can be added to a container. Once added, a file cannot be physically removed any more, however, its exclusion can be made permanent in the container. You have the option to automatically create labels for files that have been added to an evidence file container.

Optionally, hash values can be stored for the files that are copied into a container. This allows to verify the integrity of the files later, after having added the container to a case, by refining the volume snapshot. The hash values are computed directly for the data as read from the original source medium (unless you copy metadata to the container only) or taken from the volume snapshot, if available.

Optionally, the preparer of an evidence file container can pass on labels (either all or not those created by X-Ways Forensics internally) or comments about included files with the container. Useful to not only forward a collection of files to other investigators, but also case-specific information and preliminary findings. For example, the comment could explain the reason why a file was selected for inclusion in the container in the first place. Passing on internal file metadata in evidence file containers is a 3-state check box. If half checked, only extracted senders and recipients of e-mails will be passed on and not general metadata as known from the Metadata column. Please note that transferring extracted metadata to the container is not recommended if the recipient would like to work with an event list because events are not transferred to the container and events derived from within file contents will not be added to the event list if a file is marked as already metadata-processed.

Abort operation upon read error: This option allows to abort copying files into an evidence file container upon a read error and to not include affected files partially. Useful when acquiring files from a network location and the connection might be interrupted, if you assume that if that happens you will get the connection back and will be more successful when you try again, to avoid having incomplete files in the container, which cannot be replaced with a complete copy retroactively. Available only when not filling containers indirectly.

When closing a container that is open in the background, the user is offered to compress, encrypt, and/or split it. Splitting is useful if the container is complete and relatively huge, and e.g. should be sent to someone else on CDs or DVDs. You may also find it useful to have a verifiable overall hash value for all the data in the container, which can be computed at that occasion and embedded in the target container. You can also freeze the file system in the target container that you create in .e01 evidence file format, so that it cannot be filled further even if it is converted back later to its plain state again (to a raw image).

5.27 Related Items

Only available with a forensic license.

Files/directories that have a corresponding "related" file or directory in the volume snapshot are marked in the directory browser with a small blue arrow pointing downwards on the left-hand

side of their icon. A secondary tooltip appears for files with a "related" file when hovering the mouse cursor over the icon, which conveniently tells you the path and name of that related file, for example the target of a symbolic link. There are four different kinds of related objects:

1) When taking a volume snapshot of Unix-based file systems, symbolic links are connected to their targets in the volume snapshot as so-called related files, so that you can conveniently navigate to the target by pressing Shift+Backspace. Also one of potentially several symlinks pointing to a certain target will become the related file of the target, so that you can conveniently navigate to the symlink or quickly see that one or more symlinks exist that point to a certain target, since any file that has a "related" file in the volume snapshot is marked with a tiny blue arrow next to its icon. Also the same arrow will tell you whether the target of a symlink can actually be found in the file system. If a symlink links to other symlinks, those are not recursively linked. If resolving symlink takes to long because there are many symlinks in a volume, you may safely abort that step at any time.

2) When taking a snapshot of volumes with Windows installations, certain reparse points (a.k.a. junction points) are connected to their targets in the volume snapshot just like as symlinks in Unix-based file systems, so that you can conveniently navigate to the target by pressing Shift+Backspace. Also there will be a back-reference to one reparse point, so that you can conveniently navigate to that reparse point or quickly see that one or more reparse points exist that link to a certain directory, since any directory that has a "related" directory in the volume snapshot is marked with a tiny blue arrow next to its icon. Forensic license only. Reparse points that do not get connected with their target directories will still show a comment that advises you of the target path as in earlier versions of X-Ways Forensics.

3) Hard links in HFS+ point to their corresponding iNode* (indirect node) file. iNode* files point back to one of their hardlinked counterparts, so that it is very convenient to locate at least one of those hardlinks and see the actual use and location of the file. To find other hardlinks for the same iNode* file, you can for example sort by the column "1st sector".

4) Files found in volume shadow copies in NTFS point to their shadow copy host file. VSC host files point to their corresponding snapshot properties file.

5.28 Generator Signatures

The generator signature is a concept that identifies subtypes of common file types like JPEG, AVIF, videos, and PDF. Those subtypes can be associated with devices (scanners, cameras) or applications (e.g. Photoshop). For JPEG, the signature is based on the quantization table and some other invariant features that are shared by all JPEG files. The generator signature is provided with the metadata as a 32 bit raw hex number accompanied by a textual description derived from the file “Generator Signatures.txt”.

607AE169 (IJG Library 94 / Paint)

This example shows the signature that results from a JPEG file generated by Microsoft Paint. The number is the image quality in the range 1...100. 94 is the fixed image quality setting specific for

Microsoft Paint.

JPEG signatures can be subdivided into three groups. The first group is named Standard (identical to IJG Library). Files in this group make use of the quantization tables as defined by the JPEG standard. There are exactly 99 quality grades. The second group is named Extended. Here a particular grade is subdivided into roughly 100 additional grades by interpolating the standard quantization tables. Those signatures usually belong to entry level camera models that act according to size-priority compression methods.

D3D8AD02 (Extended 95.10 / 10 MP camera)

The image quality is presented with two fractional digits within the metadata column as well as with the DQT-marker in the details pane. Whether a camera operates with the size-priority scheme can be judged by the Exif field CompressedBitsPerPixel.

The third group is called Custom. Files in this group make use of proprietary quantization tables that are specific to certain devices or applications. Here too the image quality is shown in the range 0…100 with two fractional digits. Exceptions are Photoshop with 13 grades in the range 0...12, Apple Quicktime with grades in the range 1…1024, and LEAD Technologies with the range 2…255.

53631B67 (LEAD Technologies 2 / Scan)

The second part of the description, Scan, can also have the values Facebook, WhatsApp or MsPhoto. MsPhoto means that this file has been edited by Micosoft Photo Gallery.

Generator signatures form the basis of the calculation of the generic relevance. In addition, Generator signatures are used in X-Ways Forensics during the file header signature search to name carved JPEG files if no “better” metadata is available (e.g. camera model and timestamp from the Exif data). If the metadata extraction cannot find any “better” metadata, the generator signature can still be output, and that signature at least allows you to identify groups of files that likely have the same origin. Verifying whether the generator signature and available Exif metadata are consistent with each other may tell you whether a picture was edited and saved again.

In particular the generator signature allows to identify files that were produced by scanners, as there are only a handful of generators commonly used in scanners. That allows to reliably identify scanned images even if they are not black and white or not 100% using gray scale colors only. PDF files produced by scanners can also be identified by generator signatures.

PDF generator signatures are available even if there are no metadata or no metadata could be extracted. With 4,700 signatures (as of v19.0), more than 99% of all PDF files are covered. One particularly notable PDF generator signature category in the file „Generator Signatures.txt“ is “Reporting/Records”, which identifies documents like bank account statements and invoices. This identification also improves the automatic relevance judgment.

The file "Generator Signatures.txt" is similar to the other text files that ship with X-Ways Forensics and like those can be edited it to adjust the relevance estimation that is part of metadata

extraction. If for example knowing that a JPEG file was generated by a scanner is important for you (because you are a tax fraud or other white collar crime investigator interested in scanned documents), you would make sure that the “JPEG/Scan” group has a high weight (e.g. 9). That's the number after the tab in the line with the *** group definition. If such a file is of less importance to you (e.g. because the pictures that you have to look for are CP photos), then you reduce the weight of that group (setting it e.g. to 1). You can also edit the individual relevance of each generator in a group. A weight of a particular signature has to be in the range 0…9, default being 5. There is no such range restriction for the weight of a group.

The model designations of known scanning devices can be manually extended in the section "KnownScanner" of "Generator Signatures.txt". Identification by model name can help to identify scanned images if they contain Exif data or were edited. Generally the detection as scanned images is based on 1) generator signature, 2) generic properties of the Exif metadata (FileSource, Density, ...) and 3) the KnownScanner list.

The prefix "Reporting::" in generator signature definitions allows for easier filtering for the category reporting/records.

The structure of the separate file "Video Signatures.txt" is the same as that of "Generator Signatures.txt", but it deals with signatures of video files of the QuickTime format family only. It currently consists of two subcategories: Original and Generic. You may insert newly found signatures (as shown in Details mode) in the Original section if you are certain that the video has not been edited, otherwise in the Generic section.

5.29 Picture Metadata Details

In Details mode for picture files like JPEG among other things you can find the so-called Summary table. This table contains for example the generator signature (refer to the description of the column of the same name).

The picture size+ information offers textual descriptions of the resolution, the aspect ratio if worth pointing out, and potentially a known previous resolution if a picture was resized. An arrow up indicates an unexpectedly high propensity score. An arrow down indicates an unexpectedly low propensity score, which is correlated with reduced-resolution copies for dissemination and a lower generic relevance. The tiny + symbol after "Picture size" sets this row apart from the directory browser column of the same name. The internal prospensity score can assume values from 1 to 99. The relevance computation is mainly based on that value. That value is an objective statistical probability for the picture having additional, removable, relevant metadata. It could also be also be designated as documentality, the quality of being able to serve as a document. One can check this additional information for its consistency. The propensity score generally exists for raster images (that are measured in pixes), in particular also for the formats PNG and WEBP.

The "processing state" depends on the detected generator, where each generator is now assigned to one of three generator classes D (device), E (editor), or C (content management system). JPEG files produced by generator class D are absolute originals. Their processing state is always

"original". JPEG files produced by the generator class E are relative originals. Their processing state is always "Edited normally". Examples are photos published by news agencies like Reuters. The detected processing state of the third generator class (CMS like WordPress, Drupal, TYPO3, Joomla etc.) can assume different values. They are usually irregularly edited, i.e. their edited status is not officially indicated. The state can be deducted indirectly based on filename, generator signature, pixel dimension. The state "irregularly edited" can also result from picture manipulations. The state "EXIF stripped" refers to JPEG pictures, whose device origin was detected although no EXIF metadata is present. The device can potentially be detected based on generator signature, filename or a characteristic pixel dimension. The state "social media" is indicated separately because such pictures often have a higher intelligence value. Unlike news agency pictures they are rather semi-public in nature. The state "scaled" is new and refers to classical content management systems. It can be said with a high probability that such pictures have been released to the public. They were automatically and individually adapted to the respective output display in order to optimize the loading time of the web page. The state "minimized" is also new and indicates that the JPEG quality was reduced or that the file size was reduced by optimized recompression (jpeg-recompress, JPEGMini). The state "undefined" is a category for everything that remains. Such pictures are usually also the output of content management systems, those that do not identify themselves and whose format is not yet identified (which may change in future versions).

A processing state and other values (size, bits per pixel, filename analysis) are also output for PNG files. The same processing states as for JPEG are used, except "Irregularly edited" and "EXIF stripped" are not possible. The value "Original" is used only for screenshots, if they have passed a special test. A processing state is also presented for WEBP files, similar to PNG. The processing state "Original" of videos of the QuickTime format family is also brought t to your attention, if applicable. However, this statement is not as strong as for JPEG pictures. The contents of such a video may have been changed in some irregular ways without a way of detecting it (e.g. exchange of individual frames). The statement refers to the format structure. Conventional editing tools practically always alter this structure, so "normal" editing will be detected.

Special conditions are reported where detected, for example "incomplete" (if the file was truncated) or "rotated". The condition "embedded" identifies pictures that were not generated as stand-alone files, but embedded in larger files, as thumbnails or reduced resolution alternates. That condition may also occur if JPEG metadata was retroactively removed with a tool.

The compression quality is quantified in a linear scale from 0 to 100. This number is not to be confused with the nominal/official JPEG quality, which does not take the actually achieved compression into account. The average number of bits per pixels in a JPEG picture is related to this and compared to the median value for that particular generator signature to put it into perspective.

If there is trailing data after the expected end of the file, the offset is reported and the amount of data.

The amount of slack (zero-value bytes) at the end of an EXIF segment is presented in Details mode if such slack is present. For example, iPhone 4 and iPhone 5 usually produce such an area of a variable length, but iPhone 7 does not. If the slack remains present after a rotation, that

means the rotation was minimally invasive, without recompression (no loss of quality). If however a photo editing program rewrites the JPEG file, the slack will disappear.

"EXIF compliance" is another aggregated single value, a score that allows to see whether a low quality photo editor was used to edit a photo. A good rating that JPEG pictures produced by Nikon or Canon cameras usually have is retained only by high quality photo editing programs. A bad rating for such pictures indicates editing by a low quality program. Irregularly coded fields in the EXIF data are marked with a star. Irregular might mean that a wrong data type was used or the permitted value range was violated or there are duplicate tags or a character string is not null-terminated or contains slack. Some tags must not appear at the same time, some tags must be stored in a designated directory. Generally the EXIF presentation is not a simple unstructured output of all EXIF values, but it aims to provide background information and highlights certain parameters within their context to make examiners aware of irregularities. Already in their original files digital cameras produce characteric EXIF metadata errors. By editing a photo additional errors may be produced, or others may be fixed.

If the IFD GPS field in Exif metadata is available, but empty, or if it contains unvalid coordinates, this is an irregular situation, different from the IFD GPS not being present at all, and often means that the GPS data have been removed retroactively. It is reflected as "GPS format: NaN", where NaN means "not a number".

The DHT marker in JPEG files is evaluated. If the marker has the values as defined by the JPEG standard, it will be marked as "Standard", otherwise the number of table entries will be output. Practically all digital cameras use standard tables, but JPEGs encoded by social networks don't. They use optimized tables and achieve a file size reduction by around 5%.

Values in the internal metadata of JPEG files that are not default values of the respective camera/device model, but that were assigned by the user or changed by a photo editing program, or that are simply not considered original/normal for that particular device model that generated the file, are highlighted in blue color. If there is something unusual about the presence of GPS coordinates in JPEG files, those GPS coordinates are also highlighted in blue color. For example if the GPS coordinates are present and a GPS timestamp is absent, for a mobile device type that is known to always include both at the same time (sometimes depending on whether the front or back camera is used), or for a camera type that is known to not have GPS, it could mean that the coordinates have been retroactively embedded. GPS timestamps that are different from the time when the photo was taken are also highlighted in blue color. The GPS processing mode, if available, is listed in Details mode. This mode allows to estimate the reliability/precision of the coordinates. It is used by various manufacturers, and it can be one of the following values: unknown, GPS, Network, Hybrid, Fused, or CELLID. "Geolocation" shows the GPS coordinates in a notation as accepted by Google Maps, OpenStreetMap or Bing Maps. Three additional fields for Exif GPS data are output in Details mode where available: Altitude, Image direction, and GPS Error. Altitude might be helpful to judge the reliability of the geo coordinates. Image direction is a feature of high-end smartphones. For JPEG files created by many Samsung phones Details mode also shows firmware date and region, which can help to validate other metadata. The Summary part of the internal metadata of JPEG files has a field named "Light value". That value is derived from the well-known photography formula Ev=log2(N**2/t)+log2(100/ISO). The value range ends at around 16, which means full sunshine. This aggregated value can be interesting to some examiners because it allows to distinguish indoor and outdoor photos and

because it allows to check whether the local time of a photo is plausible.

"Software class" aggregates various information and at the time of writing in v21.1 can be one of the following: 28 software classes are currently supported for JPEG and WEBP pictures: AI generated, Adobe, Amazon (for photos from their shopping web site), Android, Apple, Beautifier, Bing, Camera, ContentGeneral, Editor, Facebook/Instagram, Firmware, General, Google/Picasa, LinkedIn, MSN, PHP, Pinterest, Scanner, Screenshot, Other social media, Stock (in the sense of stock photos), Twitter, Video still, Website builder, WhatsApp, Windows, Wordpress.

"Device class/type" is related to the column of the same name, and additionally reveals the confidence of that information.

“Media design” is meant to aid the assessment of a picture's aspect ratio. There are about 128 aspect ratios that represent a statistically significant variant. All other aspect ratios are labeled "Random". Particularly common aspect ratios, like e.g. 4:3, which are used by camera sensors, are labeled "Native". The group of "Framed" media designs are further distinguished as "Framed", "Square", "Scaled", "Social media" or "Featured". The latter refers to the "Open Graph" standard introduced by Google, which identifies pictures that are meant to represent a website as a whole. Media design information can be used to assess the overall consistency: A picture with a processing state labeled "Original" should always have a media design labeled "Native". A modified picture would expect a "Framed" variety, while "Featured" or "Social media" correlates with the processing state of "Disseminated". “Swipe” indicates that a picture was resized to match the screen width and enable vertical swiping on the screen. If no other tangible context exists, the media design could still be used for a general assessment.

Various special properties that are detected in pictures are referenced in Details mode with remark numbers. The text file "Annotations.txt" in the installation directory documents those numbers. Annotations are about special format peculiarities. The way they are brought to your attention is the most suitable option as they are are distributed according to a power law distribution and exhibit a long tail. In this context, the phrase ‘the exception is the rule’ holds true. The number of special features that maybe identified in a picture varies considerably. Pictures may have any number, from zero to beyond twenty-one, of special features. On average there are five special features per picture file. Reviewing the annotations ensures that you do not miss any peculiarities that could be relevant to the investigation. When dealing with evidence, it could become important to be able to provide reasons for a classification (device class, software class, processing status, sensor size, special circumstances). With help of the annotations, it is also possible to assess the confidence of a classification. You can also find several reasons that speak for a certain classification (corroboration). But it may also be that there are contradictory reasons. It is recommended that you take a closer look at the list of annotations if you have important evidence.

A file in the installation directory named "Phone Alias Table.txt" contains a translation from internal device designations to human-readable marketing names. In particular device designations used by Samsung, Motorola, LG and Huawei are rather cryptic and better understood if translated. This table can also contain the device's release date and region. Its format is explained in the header so that users can help to complete it. The table must be alphabetically sorted as that allows for enhanced performance. Note that this is just an auxiliary

table. Corresponding entries in "Generator Signatures.txt" are essential for detection and for categorization into device classes.

--

The Data Quantization Table (DQT) is a data structure in JPEG files. This table is not optional and cannot be removed from a JPEG file. Despite that, this data structure is not commonly evaluated. The general categorization of the generator signature (in the parenthesis after the hex value) is called tagging. There is always just one type tag which refers to the usually two DQTs for luminance and chrominance. At the moment, this is followed by the categorization of the JFIF marker, if present: Disseminated, Edited or Embedded. These are actually part of judging the editing state, however.

Tag "Standard" refers to the standard ISO/IEC 10918-1 introduced in 1992. In this standard, 100 DQT tables for quality parameters 01 to 100 were defined. Standard 75 is the most common, but the median is Standard 90. This tag mostly excludes the alternatives, more than a positive identification. Premium OEMs (Apple, Samsung, Adobe, Canon, Nikon) do not use that standard. A representative sample would contain about 50% Standard tags. Any deviation of that already suggests a tendency.

Tag "Extended": This tag refers to a further subdivision of the original "Standard". It is basically an interpolation of the Standard. This is mostly employed by compact cameras, to achieve a predefined file size for the JPEG pictures. This is no longer considered a relevant goal, and thus pictures with this tag are considered outdated. There are a number of distinctions within. The most significant one is "Extended Quicktime". Quicktime uses 1024 fixed subdivision levels. Quicktime is not used any more. It is a characteristic of older Apple software. "Extended MAVICA" is used by very early Sony cameras.

Tag "Photoshop": Photoshop uses quality levels 0 to 12. A Photoshop DQT is very characteristically plane and symmetrical, and thus deviates noticeably from the Standard, which uses asymmetrical DQTs. By contrast, the asymmetry of the Standard DQT defines an artificial horizon. If that horizon is oriented vertically, the tag "Portrait" is assigned.

Tag "Photoshop Web": Uses levels 00 to 100, where 00 is the lowest quality.

The group "Minimized" comprises tags jpeg-recompress, JPEGMini and Minimized. Almost all of them have the context Dissemination. Minimized means there is no static Standard Huffman table in use, but instead one that is optimized for the picture in question. This requires significant computational effort and a recompression of the picture.

The group "Google" comprises "Google adaptive" and "Google Pixel". Google has a patent for "Adaptive quantization". With this approach, a DQT is created precisely for each picture. Previously, recognition of such pictures posed a problem. There are also "Adobe adaptive" and "Galaxy S6", "Galaxy S10" and "Galaxy S23" are also created adaptively.

Age references the median age 2017 of the sample. Positive values are new pictures.

Group Share Age Relevance

Standard	54	0.7	-0.09			
Extended	9	-3.4	0.56			
Photoshop	7	-2.2	-0.07			
Photoshop Web		6	-3.8	-0.31		
Canon	5	-3.1	0.51			
Minimized	3	4.3	-0.12			
Aperture,Adobe Disseminated				2	2.7	0.08
Extended ImageIO		1.8	-2.8	-0.06		
Google	1.4	2.0	0.04			
iPhone	1.3	0.7	0.6			
PENTAX/OLYMPUS		1.3	-6.4	0.6		
Adobe (Fireworks..)		1.2	0.0	-0.06		
Nikon	0.9	-3.1	0.62			
KODAK	0.8	-4.8	0.52			
FinePix	0.7	-3.5	0.56			
Panasonic	0.7	-4.2	0.59			
Quicktime	0.06	-6.0	0.28			
ACDSee	0.6	-1.5	0.04			
CASIO	0.5	-4.8	0.54			
Mediatek	0.4	4.0	0.67			
Galaxy	0.4	4.2	0.71			
Sony	0.3	-0.5	0.59			
Lavc	0.3	-3.9	-0.05			
AccuSoft	0.3	-5.5	-0.12			
HUAWEI	0.2	4.5	0.67			
Spreadtrum	0.1	3.7	0.66			
HP	0.1	-6.1	0.54			
Amazon	0.1	2.7	-0.38			
Nokia,Broadcom		0.1	-6.2	0.32		
TOSHIBA	0.05	-9.5	0.51			
LEAD IMatch	0.1	-3.0	-0.1			
Extended MAVICA		0	-13.2	0.16		
Extended Blackberry		0.15	-5.3	0.47		
jpeg-recompress		2.3	4.6	-0.13		
Galaxy S23	0	7.6	0.67			
Minimized Twitter		0.09	5.7	-0.03		
Standard 94 Paint		1.1	3.0	-0.04		
Standard 80 PHP		1.2	-2.0	-0.13		
Google Pixel	0	6.1	0.66			

5.30 External Analysis Interface

Via the menu command "Export Files for Analysis" in the CaseData window, you can send files (for example all files in the case that belong to a certain category) to an external program for further analysis. This external program must comply with the interface described below. Requires

X-Ways Forensics or X-Ways Investigator or WinHex with a forensic license.

The analysis result can be imported back into X-Ways Forensics with the Import Labels menu command in the Case Data window. (For example, right-click the case title where it is printed in bold.) That will label files classified by the external software (and may create new label names), which allows you to filter for such files or create a report about them.

Technical description of the interface

All files or files in a certain category or all tagged files or all non-excluded files are copied into a subfolder of the output folder specified by you. The subfolder is named with a CRC in hexadecimal characters that is unique for the active case. The files are named with unique IDs (64-bit integer numbers). One additional file named "Checksum" is created that contains 4 bytes with the same CRC, 4 bytes with the handle of the main window of X-Ways Forensics (or X-Ways Investigator, for that matter), 8 reserved bytes, and 128 bytes with the case title in UTF-16. When the files have been copied, X-Ways Forensics executes the external analysis program and specifies the complete path of the subfolder in quotation marks as a parameter.

The external program can now perform the analysis. It can classify files by creating one .rtd file for each classification.

When finished, the program can optionally check whether the X-Ways Forensics main window still exists and, if so, make X-Ways Forensics aware of the availability of the results, by sending a WM_SETTEXT messages to the main window, where the text starts with "Import: ", followed by the path of the directory where to find the .rtd files, without quotation marks. This will trigger the import automatically. Alternatively, the user can import the result as described above.

The names of the .rtd files (report table definition files) will be used as the label names. An .rtd file starts with a 4-byte signature (0x52, 0x54, 0xDE, 0xF0), the 4 byte checksum (see above), followed by the 64-bit file IDs (integer numbers) that indicate the files that should be associated with that label.

6 Volume Snapshots and their Refinement

6.1 Introduction

A volume snapshot is a database of the contents of a volume or physical medium (files, directories, …) at a given point of time. The directory tree and the directory browser present views into this database. Based on the underlying file system's data structures, it consists of one record per file or directory, and remembers practically all metadata (name, path, size, timestamps, attributes, ...), but not the *contents* of files or data of directories.

A volume snapshot usually references both existing and previously existing (e.g. deleted) files, also virtual (artificially defined) files if they are useful for a computer forensic examination (e.g.

so that even unused parts of a disk or volume are covered). Operations such as logical searches, indexing, and all commands in the directory browser context menu are applied to the files and directories as they are referenced in the volume snapshot. Because of compressed files and because deleted files and the virtual "Free space" file may be associated with the same clusters of a volume multiple times, the sum of all files and directories in a volume snapshot can easily exceed the total physical size of a volume.

A volume snapshot is stored on the disk either as a set of files named Volume*.dir in the folder for temporary files or (if associated with a case) as files named “Main 1”, “Main 2”, “Main 3”, “Names”, …, in the evidence object's metadata directory.

6.2 Refinement at the Volume/Sector Level

The Specialist menu allows to *expand/refine* the standard volume snapshot in various ways, such that they contain more than referenced by the regular file system. Requires a specialist or forensic license. Full functionality only with a forensic license.

6.2.1 Run X-Tensions

X-Tensions are DLLs, which you can program yourself, to extend the functionality of X-Ways Forensics or use it automatically for your own purposes. More information.

6.2.2 Particularly thorough file system data structure search

Running a particularly thorough file system data structure search is possibly a lengthy operation, depending on the size of the volume, and for that reason not done automatically when taking the volume snapshot.

FAT12/FAT16/FAT32: Searches for orphaned subdirectories (subdirectories that are no longer referenced by any other directory).

Ext3/Ext4: Similar to the procedure for FAT. Checks the entire volume for previously existing directory structures whose contents are no longer known from corresponding inodes (these would have been looked at as part of the regular volume snapshot already). Such directories are listed with a generic name, usually in "Path unknown", but potentially in the root directory, if that is where they existed previously (the root directory is special in this situation, as it has an unchangeable ID). Optionally, certain previously existing files that otherwise would be presented only with file system metadata and no contents can be associated with data using the Ext3/Ext4 journal.

ReiserFS, Reiser4: Searches for deleted files (which are not included in the standard volume snapshot at all).

UDF: While the first and the last session of multi-session UDF CDs/DVDs will be listed automatically, additional sessions in the middle can be found only with this option.

CDFS: Usually all sessions on a multi-session CD/DVDs are detected automatically. In cases where they are not (e.g. when CDFS co-exists with UDF or if the gaps between the sessions are unusually large), this will detect sessions beyond the first one.

RAM (main memory): May find terminated processes and rootkits.

NTFS: Volume shadow copies can be parsed optionally, with a forensic license. Existing and previously existing volume shadow copy host files are checked for valuable information that would not be available otherwise, such as files that cannot be found in the current $MFT any more or previous versions of files whose contents have changed. Those files will be reconstructed up to 1 GB in length according to the shadow copy. Processing of volume shadow copies, if any, occurs before all the other operations that are part of the particularly thorough file system data structure search (parsing $LogFile, optionally searching for FILE record outside of $MFT and outside of VSC, searching for index records in the slack of INDX buffers). If there are volume shadow copies, the caption of the small progress indicator window will tell you when they are being parsed. Volume shadow copy host files that you exclude before processing will be omitted.

Files found in volume shadow copies are specially marked with "SC #" in the Attr. column, or "SC #, prev. version" if they are previous versions of files that were known to the volume snapshot already before the thorough file system data structure search, so that it is easy to filter them in or out. # stands for the sequential number of the snapshot in which these files were found. Remember you can sort by ID to see the files they are a previous version of next to them. You can also easily navigate to the VSC host by using the command Navigation | Find related file in the directory browser context menu, for example so that in Details mode learn more about that particular snapshot. You could then invoke the same command once more to navigate to the corresponding snapshot properties file, where in Details mode you learn even more, e.g. description and official creation date.

Optionally avoid that previous versions of files in volume shadow copies are added to the volume snapshot if they are exact duplicates (identical file contents) so that it is much easier to focus on files for which actually previous data is still available. Even if modification dates are different, the file contents are often the same for files installed by the operation system. If fully selected, X-Ways Forensics will compare files up to 128 MB, if half selected, only up to 16 MB, as to not waste too much time on this feature.

NTFS: FILE records can be optionally searched everywhere, in sectors that neither belong to the current MFT nor to a volume shadow copy (VSC) processed by the above-mentioned option. Such FILE records can be found e.g. in free space after a partition has been recreated, reformatted, moved, resized, or defragmented. Time consuming on very large partitions. Clusters belonging to certain virtual machine disk image types are skipped to avoid the inclusion of files in the file system of the virtual machine in the volume snapshot of the host machine's volume.

NTFS: With a forensic license, the current $LogFile as well as old versions of $LogFile found in processed volume shadow copies can be exploited. The contents of deleted files can often be reconstructed thanks to $LogFile. Index records remnants in $LogFile as well as in the slack of

INDX buffers can be exploited that either reveal previous names or paths of renamed/moved files/directories that were known to the volume snapshot before or deleted files that the volume snapshot was not aware of before (without file contents, though). You can indicate whether you are interested in earlier names and paths of renamed/moved files and directories or not. If the checkbox for earlier names/paths is half checked, you may find earlier names/paths of renamed/moved files in the Metadata column and don't get additional files in the volume snapshot for each earlier name/path. You can also indicate whether you are interested including traces of files in the volume snapshot whose clusters are unknown and for which only name, size, timestamps and attributes are available.

During all the suboperations for NTFS, the inclusion of redundant (identical) files in the volume snapshot is avoided as much as possible. If the only new information gained from old versions of FILE records or index records is previously valid timestamps, no earlier names/paths/contents of files, or if you have indicated that you are not interested in earlier names/paths, then these timestamps are only output as events, depending on the volume snapshot refinement option "Provide by-catch timestamps from various sources as events".

NTFS: You can indicate whether you are interested in getting files included in the volume snapshot whose clusters (and therefore data) are totally unknown, with only metadata (e.g. filename, path, size, attributes, and timestamps), as may be found in index records in INDX buffers or in $LogFile. If checked, all previously existing files of which metadata only is known will be included in a volume snapshot. If not checked, those files will be ignored.

Other file systems: no action taken

6.2.3 File Header Signature Search

The “**File header signature search**” operation helps to include files in the volume snapshot that can still be found in free or used drive space based on their file header signature and are no longer referenced by file system data structures. You are asked to select certain file types for detection, specify a default file size, an optional filename prefix etc. Please see “File Recovery by Type” and the file type definitions for details. Files found with this method will be included in the volume snapshot only if there is no other file in the volume snapshot with the same start sector number yet (overwritten files don't count), to avoid duplicates. However, files that are not aligned at sector boundaries will always be included for performance reasons. Files found with this method are listed with a generic filename and size as detected by the “File Recovery by Type” mechanism. If applied to a physical, partitioned evidence object, only unpartitioned space and partition gaps will be searched for file headers, because the partitions are treated as separate, additional evidence objects.

Usually, results of the file header signature search are output in a special virtual directory for carved files, which is a subdirectory of "Path unknown". However, there is an option to show resulting files as child objects of existing files, if the carved files were found within these other files.

On BitLocker volumes that it can decrypt, X-Ways Forensics tries to automatically detect

unencrypted areas. Such areas can be present if only in-use drive space was encrypted and rewritten when the BitLocker volume was created, for example for performance reasons or because the security implications of this were not understood. If this situation is detected, X-Ways Forensics will recommend running your analyses also on the undecrypted volume, bypassing BitLocker decryption. The file header signature search can additionally and automatically perform a second run on the data directly as stored in a partition that is protected with BitLocker, bypassing the decryption algorithm. Either only if the presence of unencrypted areas was detected by X-Ways Forensics in the BitLocker volume (potentially just seconds before during the first, regular run of the file header signature search!) or, if fully checked, on any BitLocker volume that is processed in its decrypted form. X-Ways Forensics will specifically remember which files were carved (automatically or manually) while BitLocker decryption was bypassed so that those files in future can be read correctly even when BitLocker decryption is otherwise active. The Description column will identify such files. When working with the decrypted BitLocker volume, switching between Volume/Partition and File mode for such files will show the obvious difference between the data that are either passed through the decryption algorithm in the former modes (falsely, because it was never encrypted in the first place) or not in File mode.

6.2.4 Block-wise Hashing and Matching

Available with a forensic license. Block-wise hashing may allow to identify complete or incomplete remnants of known notable files that are still floating around in free drive space even if they were fragmented and the location of the fragments is unknown, to show with some or very high certainty that these files once existed on that medium. The hash values are computed when reading from the evidence object sector-wise, and that happens at the same time when running a file header signature search if selected, to avoid unnecessary duplicated I/O, with the same sector scope. Matches are returned as a special kind of search hits. Multiple matches for contiguous blocks are more meaningful than isolated individual matches, as they are even less likely the result of some coincidence, and they are usually combined in a single hit. The size of all such hits is shown when listing search hits. The larger the size, the higher the evidentiary value of the match. Please note that X-Ways Forensics does not verify itself that contiguous matching blocks are in the same order as in the original file(s), but that can be verified manually and for data that is as unique as compressed data that is most likely the case.

Most suitable for selected notable files larger than a few sectors, files that are ideally compressed or at least not only sparsely populated with non-zero data and do not contain otherwise trivial combinations of bytes values that occur frequently. Good examples are zip-styled Office documents, pictures and video files. Very trivial blocks within a file that consist of mostly just 1 byte value are ignored and not hashed (the same already when creating the hash set). For quicker matching, ideally work with a small hash database and do not select a hash type stronger than MD5. The length of block hash matches is shown in the Size column. This is useful so that you can sort them by the lengths and review more important (larger) matches first.

Hash sets of block hashes can be created or imported in the same way as ordinary hash sets, i.e. for selected files using the directory browser context menu, but they are handled by a separate hash database for block hashes (as opposed to file hashes). You can create hash sets consisting of

the block hashes of 1 file at a time, or combined hash sets of multiple selected files.

6.3 Refinement at the File Level

The below operations are applied *after* the aforementioned operations, to files that are already contained in the volume snapshot, and they are all applied together and file-wise (i.e. first all operations to one file, then all operations to the next file, and so on), to process files in the order of ascending internal IDs. Some of these operations may produce additional files, which will get the next higher available internal ID. Previously existing files whose first cluster is known to have been overwritten or whose first cluster is unknown are not processed except if you specifically target them via tagging or selection.

Files that are considered irrelevant based on hash matching can be automatically omitted from all further operations to save time and avoid potentially even more irrelevant files that might otherwise be extracting from them. It is also possible to omit not only known irrelevant files, but also known relevant files from further processing. Useful for example if in large cases you have or expect really many such files and having proof of their presence is sufficient for you and you don't need to extract their internal metadata, don't need to compute their skin tone percentages or PhotoDNA hashes, and don't need to check them for embedded data etc. There is also an option to omit files that are filtered out. All of these options are particular powerful in that they can target even files in advance that are not yet part of the volume snapshot when the refinement starts. For example when additional files are added to the snapshot by the file header signature search, depending on the file type these files can be further processed (e.g. hashed) or not, if the Type filter is active during the later stages of the volume snapshot refinement.

There is an option to omit additional hard links for the same file in NTFS/HFS+ from volume snapshot refinement just as from logical searches, to save time and reduce the number of redundant identical child objects etc. This can make a big difference on partitions with Windows installations that have a lot of hard links and HFS+ partitions with Mac OS X Time Machine. Which hard links are considered the "additional" hard links internally can be seen in the "Link count" column (gray number means to be omitted) and also in the Description column, which identifies all hard links (i.e. files with a hard link count larger than 2) and the additional ones in particular textually. The hard link that is not marked as "optionally omitted" in the Description column is considered the "main" hard link internally.

Previously existing files whose first clusters were reallocated according to the file system (shown as "1st cluster not available") are usually omitted from volume snapshot refinement. If you choose to not omit them, that means that unrelated random data that does not belong to such a file can be hashed and presented as that file's hash value although it most certainly was *not* that file's hash value. If X-Ways Forensics was forced to compute them, hashes of nonsensical data will be displayed in the directory browser in gray color, to remind the user that they should not be overinterpreted or expected to be found on other storage devices.

6.3.1 Hash Value Computation and Matching

Hash values can be computed for files in the volume snapshot. They are not recomputed if you apply this operation again to the same files. In addition to the mere hash computation, a forensic license allows to **match** the hash values against individually selected (or simply all) hash sets in an internal hash database. The filter can then later be used to hide known irrelevant files. Files recognized as irrelevant with the help of the hash database can be optionally excluded from further volume snapshot refinement operations, which among other benefits saves time. The hash values will not be updated in the volume snapshot once computed. However, the *matching* process (looking up the hash values of files in the volume snapshot) can be repeated for the same files at any time. This will first discard previous hash set matches from all files in the volume snapshot, except if you apply the matching only to tagged files (in which case only those files will lose their previous hash set matches). The hash category field will be updated only, but not emptied.

It is possible to compute hash values of two different hash types at the same time when refining the volume snapshot, for general purposes or to match them against two hash databases with different hash types. If matching is selected, all hash values will be matched against any of the two hash databases whose hash type fits. That means even if the primary hash type in the volume snapshot is MD5 and the secondary is SHA-1, and hash database #1 is based on SHA-1 and #2 based on MD5, X-Ways Forensics will match the hash values accordingly. The hash types in the volume snapshot and in the hash databases do not have to be in the same order.

A forensic license allows to verify hash values that were computed at an earlier point of time, or imported from an evidence file container. The result will be output to the Messages window. Any file whose current hash value does not match the originally recorded one will be associated with a special label for convenient review. Running the hashing volume snapshot refinement step a second time never updates the hash values that were already computed for files in the volume snapshot.

Child objects of files inherit the hash category "irrelevant" from their parents. That is possible because if an entire file is irrelevant, everything that can be extracted from that file must also be irrelevant. However, what is extracted from a "notable" file is not necessarily also notable, because perhaps only some parts or aspects of the parent file are notable. Of course, child objects of irrelevant parents will only be output if the user chooses to not omit irrelevant files from further processing in the first place.

When matching hash values against hash databases (ordinary hashes like MD5, SHA-1, SHA-256, ...), there is an option to make a local copy of the database and work with that copy. This can be helpful if you share the database with your colleagues and your colleagues want to update the database (e.g. add additional hash sets) while it's in use for matching, which otherwise would not be possible for the whole duration of volume snapshot refinement. It could also enhance performance if the database is large and does not fit into main memory and is stored on a remote network drive. The local copy is created in the directory for temporary files if it does not exist yet, and updated only if the master copy of the hash database has changed (all users should have v19.8 or newer to avoid unnecessary copying of an unchanged database).

In addition to conventional hash databases you can also match the hash values of files against the Hash Comments database to retrieve comments/descriptions that were previously attached to known relevant files by you or colleagues/peers in other cases.

MD5/2 is one of the hash types offered, and that option refers to MD5 hash values of half the regular length ("folded", i.e. first half xor'ed with the second half, yielding 64 bits), as an economical compromise between CRC32 (32 bit) and regular MD5 hashes (128 bits), to bridge the gap between the two and save memory and/or drive space, for example for deduplication purposes.

Another special kind of hash value is EDRM MIH, an eDiscovery standard. Such hash values can be computed for extracted e-mail messages and original .eml and .emlx files (if the contain complete headers), to search for e-mails with a known MIH value, for database matching, or for deduplication purposes. If an MIH is assigned to an .eml file in a volume snapshot and the .eml file was extracted from an .msg file, the same MIH will automatically be assigned to the parent .msg file as well. Two copies of the same e-mail message may have different regular hash values, but the same MIH, for example if the file format is different (raw .eml file vs. OLE2 MSG file) or if body format and/or content are stored differently. If EDRM MIH is selected as the hash type, but no MIH can be computed because the targeted file is not an e-mail message of a supported type, the hash value cell remains blank. As a compromise, you can choose MD5/MIH as the hash type, where an MIH is computed if possible, or an MD5 hash value if not, so that all hashable files get a hash value usable for deduplication or matching.

6.3.2 File Type Verification

A forensic license allows you to **verify file types based on signatures and** various **algorithms**, i.e. detect filename/file type mismatches in all files in the volume snapshot except those whose original first cluster is known to be no longer available. For example, if someone has concealed an incriminating JPEG picture by naming it "invoice.xls" (wrong filename extension), the recognized file type "jpg" is stated in the Type column of the directory browser. For more information see the description of the columns Type and Status. The file signatures and extensions used for mismatch detection are defined in the accompanying file type definition files, which you may fully customize. It it the same database also used for file header signature searches. Please note that the link between the current data in a free cluster and a deleted file that previously was stored in that cluster and its filename is weak, so that a discrepancy between filename extension and detected type can simply be the natural result of a reallocation of this cluster to a totally different file in the meantime. If you wish to repeat the file type verification, e.g. after editing the file type signature database, be sure to check the Again option. For the status of the Type column of the directory browser, see the “Type status” column.

Most self-extracting .exe archives are internally detected by the file signature check, too. They are classified as the file type “sfx” and assigned to the category “Archives” so that they can be specifically targeted. This prevents that compressed files in such archives go totally unnoticed in an investigation. .exe archives with Zip compression can be viewed in Preview mode, other self-extracting archives need to be copied off the image and opened with an appropriate tool like WinRAR or 7-Zip.

The file signature check also reveals hybrid MS Office files, i.e. merged MS Word and MS Excel documents that can be opened in both applications, showing different contents. A notice in the messages window will be displayed, and any detected files will be associated with a special label. Hybrid MS Office files are a clever attempt to conceal the contents of one of the merged documents.

6.3.3 Extraction of Internal Metadata

Requires a forensic license.

a) Can check the file format consistency of EXE, ZIP, RAR, JPEG, GIF, PNG, RIFF, BMP, and PDF files. The Type Status column will show the result, either "OK" or "corrupt".

b) Allows to extract internally stored creation times from OLE2 compound files (e.g. pre-2007 MS Office documents), EDB, PDF, MS Office HTML, EML, MDI, ASF, WMV, WMA, MOV, JPEG, THM, TIFF, PNG, GZ, GHO, PGP pubring.pkr keyring, ETL, SQM, IE Cookies, CAT, CER, CTL, SHD printer spool, PF prefetch, LNK shortcut, and DocumentSummary alternate data streams. This timestamps will be shown in the Int. Creation column of the directory browser. In some cases the earliest timestamp will be extracted, which approximates the real, original creation date best.

c) Allows to copy certain file metadata to the Metadata column, which will allow you to filter by this metadata, to export the metadata with the Export List command, and to output it in a report table in a case report. Metadata can be extracted from all the file types specifically supported in Details mode plus Windows shortcut files (.lnk) and prefetch files (.pf). Only a subset of the metadata that you see in Details mode is extracted. You have the option to strip certain lines off the extracted metadata in order to not see them in the Metadata column, for example to keep the case report or the output of the Export List command more compact for printing or viewing on the screen, or just because certain metadata fields are not relevant to you. You can identify unwanted metadata fields by a substring. That substring can either match the field name (e.g. "Focal Length") or the value of the field (for example if you know in advance that you are not interested in the Author field if the name of the author of a document is "Joe Huber"). 1 substring is entered per line. Substrings may contain spaces. You can share your definitions by sharing the file "Unwanted Metadata.txt".

Jump list hash values are translated to application names in the presented metadata of customDestionations-ms and automaticDestinations-ms jump list files, based on a new user-editable text file named "Jump List Names.txt". The translation table currently consists of around 500 entries. If you add entries, please make sure to insert them at the correct place such that all entries remain sorted by the CRC in ascending order. Leading zeroes in the CRC obviously must be preserved. There is a tab character between the CRC and the application name.

d) Allows to restore original file system metadata (such as filename, timestamps) when found in certain file types such as $I* recycle bin files and iPhone mobile sync backup indexes (Manifest.mbdx). Original filenames are typically much more meaningful than random names

that are assigned just to guarantee uniqueness in a single directory for backup purposes. Examples of such random names are 3a1c41282f45f5f1d1f27a1d14328c0ac49ad5ae (for a file in an iPhone backup) or $RAE2PBF.jpg (Windows recycle bin). The current filename according to the file system can still be seen in square brackets in the Name column, as well as in Details mode, and the Name filter will find both the original and the current name, so that current filename is not completely lost.

Alternative names and timestamps are also extracted from Linux PNG thumbnails as known from Ubuntu and Kubuntu distributions, desktop manager MATE and GNOME ThumbnailFactory. The name of the original file is shown in square brackets in the Name column and the recorded timestamp of the original file is shown as a "Content created" timestamp. The complete path of the original file can be seen in the Metadata column.

e) Populates the Sender and Recipients columns for original single e-mail files (.eml, .emlx, .olk14msgsource). Extract the subject of such e-mail messages and shows it in the Name column if different from the name of the file, and unless the file is a carved file (i.e. a file with an artificially generated filename), the original filename will be preserved and shown as an alternative name in the same column.

f) Creates previews of Internet browser SQLite databases, which may require that the files have been checked for their true file type. Supports Firefox history, Firefox downloads, Firefox form history, Firefox sign-ons, Chrome cookies, Chrome archived history, Chrome history, Chrome log-in data, Chrome web data, Chrome sync, Safari cache, Safari feeds, and Skype's main.db database about contacts and file transfers. The Google Chrome history also displays the transition for each visited web site, making it easier to ascertain whether the visit was triggered by the user or by some other action like redirect. The duration of each visit is listed as well. Internet searches run from the address bar of Chrome are listed in a separate table and also added to the event list. Parses Google Chrome SNSS session files (Current/Last Session and Current/Last Tabs). The resulting session overview lists all open tabs and their browsing history. Creates previews also of Internet Explorer index.dat files (including artificial index.dat files compiled from individual records from various locations during the file header signature search), Internet Explorer 10's WebCacheV*.dat files, the Edge browser's spartan.edb file (all favorites and ReadingList entries will be added to the event list), $UsnJrnl:$J, Windows Event Logs (.evt and .evtx), Apple FSEvent logs. From iOS's sms.db all recorded conversations via SMS are extracted to individual chat files, and all messages are added to the event list, where they can be filtered based on phone number or email address. Also extracts browsing history information from Safari's icon database. This alternative source is very interesting because it records browsing history even when Safari is in private browsing mode.

X-Ways Forensics can extract specific data from the event payload in .evtx event logs and list them directly in the event list. This makes working with event logs much more powerful, as it allows to quickly filter for usernames, IP addresses from log-in or RDP events, task or service names, PowerShell commands, etc. There is a tab-separated definition file "Event Log Events.txt" in the installation directory that contains a list of event IDs, (optional) log provider, the list of individual data fields to extract (separated by commas) and (optionally) a text comment that will be added to the event's description field. The definition file can be adjusted to your own requirements, including commenting out individual lines by placing a semicolon in the first column.The events in an .evtx file are output in a TSV table. That table contains the complete

payload of each event. It is ideally viewed in MS Excel or similar applications.

HTML previews and views of index.dat Internet Explorer browser cache/history files contain a column with the offset of the record within the file where the data of each row has been found. This offset is presented as a link. If you click it, you will automatically navigate to that offset in the corresponding index.dat file in File mode so that it is convenient to verify the information that X-Ways Forensics has extracted from the record at that location. (Note that this works correctly only if the link is not broken into 2 lines, which may happen in v8.4 of the viewer component, but not in v8.3.7. Anyway you can still navigate to that offset manually.)

Metadata and events are extracted from SRUDB.dat, i.e. the activity captured by the system resource usage monitor (SRUM). You can see the processes started over time, listed with their owners, and a lot of statistics. Network usage activity by each process is extracted as well. The extracted information can be useful to pinpoint the moment of a possible intrusion or the process that caused an intrusion. The information is presented in detailed HTML child object files and as events in the event list. Support is also included for iOS netusage.sqlite files, which record the data usage of apps. Besides the amount of data flowing in and out, they also provides approximate timestamps when apps were used for the first and last times. Appropriate events are extracted and an HTML preview is created containing all relevant information.

The HTML child objects that will be generated can not only be used internally by X-Ways Forensics for previews of the parent file. You can also view all of these tables in an external program such as your preferred browser or in MS Excel, by sending these child object to the program of your choice (directory browser context menu). You may have X-Ways Forenscis split HTML tables after an arbitrary number of rows. You can set this number much higher if you do view the HTML previews externally with your preferred Internet browser and not with the viewer component, which cannot deal with very large tables. The existence of HTML child object with searchable text for browser data, event logs and more data sources also improves effectiveness of searches and indexing.

g) Extracts tables from various other SQLite databases in TSV format and uses the first one as a preview of the SQLite database file itself.

h) Extracts the original revision of PDF documents that were edited, if available, as a child object.

i) Provides timestamps from the file system as events to analyze in an event list.

j) Provides internal timestamps in files as events.

k) A generic relevance of files can be estimated, and you can check out files in the order of relevance by sorting by the Relevance column. The weight with which the currentness and the size of a file affect its computed generic relevance is user-definable. 100% means default weight. 50% means half of that. 0% means the factor has no effect at all. The maximum is 255%. The weight of the device type for the generic relevance judgement can be defined in the file Generator Signatures.txt. The weight factor can be found at the end of the *** line. It may be between 0 and 50. For pictures in the formats JPEG, PNG, GIF and WEBP the algorithm tries to put more emphasis on intelligence value rather than news value, and to weigh evidential value

higher than informational value. A relevance value of 3.0 is the base value defined for JPEG files in File Type Categories.txt. This value is also what you can expect from pictures that are just advertising. 3.2 = typical browser cache picture. 3.5 = typical for a picture from the system partition. 3.9 = social media. 4.1 = webcam. 4.2 = backup. 4.7 = photo as originally taken by a digital camera. Sorting picture by relevance achieves a grouping effect in the gallery because pictures from a similar context are sorted next to each other.

l) The "structure type" column can be populated. The structure type is an improvement on the generator signature concept with the idea of a scalable typology, filling the gap between file type and hash value. The structure type is presented as a 32-bit integer number in hexadecimal notation. Identical numbers typically identify pictures/videos/documents/fles that belong to the same sequence (for example photos that were likely taken during the same photoshoot). The structure type is determined for JPEG, PNG, GIF, WEBP, BMP, DOC, XLS, WAV, EML, MSG, GZIP, ordinary ZIP, TAR, MP3, HTML, PDF, Quicktime videos (MP4, MOV, 3GP, ...), and for DOCX, PPTX, XLSX. Please validate any insights gained with this column using timestamps and additional metadata. You can copy the structure type for a file of interest and use the column's filter to search for files with the same structure type (or use the "Filter for similar files" context menu command for that). The structure type may also be useful as a criterion in the "Find duplicates in list" command.

m) When X-Ways Forensics determines the generating device type for files of certain supported file types, it will output the result in the "Device type" column if the confidence of this finding exceeds the minimum percentage that you specify.

HTML report files can be generated automatically for the Windows Registry hive files NTUSER.DAT, SYSTEM, SOFTWARE, SECURITY, and SAM as part of metadata extraction, based on the “Reg Report *.txt” definition files that you have in your installation directory (a number of which are preinstalled). The HTML files are added to the volume snapshot as child objects. The benefit is that they can serve as human-readable previews of selected interesting values, and they contain some encoded text in plain text such as UserAssist entries, so that the logical search can find them. Lots of timestamps from the processed registry hives will be added to the event list at the same time. This all happens if the user also chose to generate HTML previews for browser databases etc. and/or to populate the event list with internal timestamps in files.

6.3.4 Archive Exploration

A forensic license allows to include the contents of **ZIP, RAR**, ARJ, GZ, TAR, 7Z, CAB, and BZIP **archives** in the volume snapshot, so that files in such archives can be separately listed, examined, searched, etc., in their decompressed state, as long as the archives are not encrypted. Theoretically, there is no limit to the number of nested levels that can be processed (i.e. archives within archives within archives…). If the files are encrypted in the archive, they are marked with “e” in the attribute column and the archive itself with “e!”. This allows to easily focus on such files using the attribute filter.

Document files of MS Office 2007/2010/2013, LibreOffice, OpenOffice, and iWork are typically

Zip archives, too, technically, and if so are processed in the same way by default. You can choose to not process those files if you or the recipients of evidence file containers that you prepare only wish to see the documents as a whole, no embedded pictures or XML files separately, and don't need to extract metadata from these XML files and can recognize nested documents (documents embedded in other documents) themselves if necessary. There are many, many other file types that are technically subtypes of Zip that are processed optionally. Zip subtypes whose contents are usually irrelevant are for example .jar, .apk and .ipa, though special interest groups like malware investigators might think otherwise, so the choice is yours.

For each "family" of file archives (general purpose, Office, special interest, ...) you can decide whether such archives should be presented in the directory tree in the Case Data window once their contents have been included in the volume snapshot, just like directories are.

X-Ways Forensics tries to detect and protect itself against of zip bombs as well as recursive zip and gz archives and possibly other recursive archive types. Protection means that processing will stop at a certain level once the malicious nature of the archive is detected. Archives identified in this fashion will be marked as already processed and added to a special internal label. Please note that if afterwards you wish to manually dig deeper than the level at which the recursive automatic exploration stops, you can do so by marking the inner-most archive reached as still to be processed (by pressing Ctrl+Del) and then applying the Explore command in the context menu to it manually.

Note that for Zip archives with non-ASCII characters in filenames to be processed correctly, you need to pick the correct code page in the case properties first. E.g. for Zip archives created under Linux, that's likely UTF-8. For Zip archives created under Windows with WinZip, that's likely a regional code page. Spanned Zip archives in PKZIP/WinZip and 7-Zip styles are supported, plus spanned 7z archives, but no other split/spanned/segmented archive types.

Extended timestamps from the extra field in zip records are extracted and presented in the timestamp columns based on Apple specifications, which is not always how these timestamps were meant. An alternative interpretation can be seen for each zip record in Details mode when selecting the zip archive. The latter interpretation shows these timestamps with the "UT" prefix and tries to recognize the actual format variant, for example that used in GrayKey collections, and from GrayKey collection also extracts an an additional type of timestamp (a record change timestamp). The alternative interpretation of extended timestamps can also be made available in the directory browser. This is an option in Options | Volume Snapshot. That kind of processing needs some more time.

Encrypted ZIP, RAR, and 7Z file archives can also be processed, provided that the password is known or can be guessed. X-Ways Forensics will try any password listed in the password collection of the current case. The case-specific password collection can be edited from within the case properties, and it is stored in a UTF-16 encoded text in the case directory, named "Passwords.txt". Almost all Unicode characters are supported, including space characters and Chinese characters etc. Passwords are usually case-sensitive. If the collection contains the right password for a particular file archive, that password will be remembered in that file's extracted metadata and taken directly from there instead of the password collection if needed again later to read files in the archive. Alternatively, you can provide a specific password for a particular file archive manually and directly by editing that file's metadata, you just need to know that the

password must be prepended with "Password: ". (Note to French users: No space before the colon.) Files within encrypted file archives are not treated and shown as encrypted ("e" attribute) if the right password was available at the moment when the files were added to the volume snapshot. The archives themselves are still shown with the "e!" attribute. RAR archives and 7Z archives in which not only the file contents, but also the names are encrypted are not currently supported.

6.3.5 E-mail Extraction

A forensic license allows to separately list and examine **e-mail messages** and e-mail **attachments** stored in the following e-mail archive file formats: Outlook Personal Storage (.pst), Offline Storage (.ost), Exchange (.edb, Exchange 2010 and earlier supported, 2010 still in a testing stage), Outlook Message (.msg), Outlook Template (.oft) certain databases of Microsoft Outlook for Mac (.olm), Outlook Express (.dbx), Outlook for Mac, Kerio Connect (store.fdb files that can be processed like ordinary PST/OST files), AOL PFC files, Mozilla mailbox (including Netscape and Thunderbird), generic mailbox (mbox, Unix mail format), MHT Web Archive (.mht), winmail.dat = TNEF files. By default, X-Ways Forensics tries to extract from these file types:
pst,ost,edb,dbx,pfc,mbox,eml,emlx,mht,mim,msg,olk14msgsource,olk14message,olk14msgattach,olk15msgattach,olk15msgsource,olk15message,oft,mbs,tnef,olm

E-mail messages are usually output as .eml files. To conveniently focus on all extracted e-mail messages from all e-mail archives (and even processed original .eml files) it is recommended to explore recursively and use the Attribute filter (not the Type or Category filter). Deeply nested e-mails (that have been forwarded as an attachment and whose parent was forwarded again as an attachment) extracted from MBOX archives are presented as direct child objects of the main parent e-mail message for technical reasons. There is an unlabelled, but tooltipped checkbox that will make X-Ways Forensics name MSG files after the e-mail subject when extracting e-mail messages and attachments from them. That could be useful when dealing with generically named MSG files.

For extracted e-mails and their attachments, sender and recipient will be displayed in the corresponding columns in the directory browser. You may filter by creation and modification dates as well as sender and recipient.

If e-mail messages have a Sender: line in addition to a From: line, then the sender according to the Sender: line is shown in the Sender column of the directory browser additionally, after the From: sender, if actually different. They are delimited by spaces and a pipe (|). For example, an English language MS Outlook shows such e-mails as having been sent "on behalf of" someone else (by the Sender: sender on behalf of the From: sender). You can filter for such e-mails by entering a pipe as a substring for the Sender column. Analogously, different kinds of recipients (To:, Cc:, and Bcc:) are delimited by pipes in the Recipient column.

Attachments and embedded files are extracted, too, if found in the e-mail archive (exception e.g. AOL PFC) and usually become child objects of their respective containing e-mail messages in the volume snapshot. All extracted e-mails and attachments actually reside in the evidence

object's metadata subdirectory and may utilize a lot of drive space.

E-mail extraction from PST can process password-protected PST archives without the password! It supports the following code pages for encoded PST files: ISO8859-1, ISO8859-2, ISO8859-3, ISO8859-4, ISO8859-5, ISO8859-6, ISO8859-7, ISO8859-8, ISO8859-9, ISO8859-10, ISO8859-11, ISO8859-13, ISO8859-14, ISO8859-15, ISO8859-16, koi8-r, koi8-u, 1250, 1251, 1252, 1253, 1254, 1255, 1256, 1257, 1258, 874, UTF16, UTF32, UTF8

In certain old AOL PFC files, pictures may be embedded in e-mail messages in a special way. In that case, such an e-mail message will be marked with a paperclip icon, but the picture will not be separately extracted. The picture, if JPEG or PNG, can be found, however, when extracting JPEG and PNG files from *.pfc.

Some advantages of the .eml format for output: E-mail messages output as .eml files are represented as simple and as authentic and universal as it gets. They are easy to understand, clearly structured into header and body, and extremely easy to completely view in a variety of simple programs (e.g. text editor, word processing, Internet browser, free e-mail clients like Thunderbird and Windows Mail). No commercial software like MS Outlook needed is needed to view .eml files. .eml is the "natural" format of e-mail, just like a raw image is the natural format of a disk image, if you even want to call it a "format" (actually it has no additional format specifications, it's just a plain representation of the data that it should represent). An .eml file contains the complete original metadata of the e-mail message, fully intact, exactly as it was sent and delivered. You have complete control over the file if you copy it out for someone else, can see all data, can verify that no unintended data made it into the file. You can easily redact any text in the body manually with a simple text editor, redact any metadata in the header, easily retroactively remove any attachment using a simple text editor if needed, all of which is impossible to do with a complex proprietary binary file format such as MSG. The general format of .eml files can be understood by anyone, and it is simply a text file. The format of MSG files can be understood only with a computer science or programming background, and learning it takes a lot of time. Redacting e-mail data hidden in MSG files is difficult.

A side task of e-mail processing is to extracted files from e-mail related MIM archives and make them accessible as child objects in the volume snapshot in plain binary form.

6.3.6 Uncovering Embedded Data

Forensic license only. Allows to carve files of various types that are embedded in files of other various types, through a byte-level file header signature search within certain files. This is successful if the outer file (host file) is intact and the embedded file is not stored in the host file in a fragmented manner. Otherwise the embedded files may appear as corrupt. Notably this function searches for JPEG and PNG pictures, even JPEG pictures in other JPEG files (those that contain thumbnails of themselves). The files found this way will be generically named as "Embedded 1....jpg", "Embedded 2....png", etc. Thumbnails in HEIC files are output in JPEG format.

This function also extracts .emf files embedded in multi-page printouts (.spl spooler files). .spl

files that contain a single .emf file only can be viewed directly with the viewer component. Also extracted this way are .lnk shortcut files from .customdestinations-ms jumplists.

Special internal algorithms exist that properly extract, by following the data structures in the respective file format, even if fragmented, .lnk shortcut files from .automaticdestinations-ms jump lists, files of various types from OLE2 compound files (e.g. MS Word .doc, MS PowerPoint .ppt), Firefox browser caches (based on "_CACHE_MAP_" files), Safari browser caches, Norton Backup files (N360 backup, .nb20) and Windows.edb databases (from the latter even e-mail messages), and pictures that are embedded as Base64 in VCF files (electronic business cards).

Chrome browser caches are processed based on "index" files, with support for multiple streams of the same cache entry: The HTTP response (named .chrome1) is output as well as, if present, as are compiled JavaScript entries (.js1). If a no-cache directive was sent by the web server, at least the HTTP response is still cached. In Preview mode you can see a special representation of HTTP responses. Chrome caches can now also be processed if their index is not available, for example if cache fragments have been carved or if the cache was partially deleted or corrupted. It may be possible in some cases that a better extraction result can be achieved without the index, even if it is present. To try that, if the index has not been processed before, you can have the uncover function process "data_4" files and omit the index. data_4 is part of the optional "special interest" group.

Also extracted are thumbnails from thumb*.db files, from Google's Picasa 3 image organizer and viewer software (thumbindex.db and related files), Photoshop thumbnail caches (Adobe Bridge Cache.bc), Canon ZoomBrowser thumbnail collections (.info), and Paint Shop Pro caches (.jbf). Thumbnails in certain very old "thumbs.db" files cannot be displayed correctly. Such thumbs.db files will get the label "Unsupported thumbs.db" and can be viewed e.g. with the freely available program "DM Thumbs" by GreenSpot Technologies Ltd. Thumbcache*.db files of Windows Vista and later are targeted indirectly if thumbcache_idx.db is in the mask and if that file is available in the same directory. That speeds up the extraction and avoids the output of numerous duplicate thumbnails (only the highest available resolution is output). If thumbcache_idx.db is in the mask, that also means that thumbcache*.db files that are specifically selected or tagged for processing are not processed unless the thumbcache_idx.db file is also selected/tagged.

Also, from PDF documents it extracts any kinds of files that are marked as embedded plus JPEG and JPEG 2000 plus Acrobat form files in XML format plus JavaScript objects (the latter may make it easier to determine whether a PDF file should be considered malware). Extracts individual cookie files from Firefox and Chrome SQLite databases, also data blocks embedded as Base64 in XML-formatted PLists (.plist) and raw data blocks embedded in binary PLists (.bplist). It is recommended to verify file types at the same time so X-Ways Forensics can distinguish between traditional (XML-formatted) PLists and binary PLists (BPLists). Many PLists do not have a .plist extension and need to be identified as PLists first. Since the type of the embedded data is not identified by the PList as such, the output also benefits from a simultaneous file type verification. Nested PLists (PLists embedded in PLists) will also be identified and processed recursively. Another child object created for PLists represents parsed text in a human-readable way and serves as a preview of the PList itself.

Also reconstructs e-mail messages and extracts contact and account information from the Livecomm.edb database, which is used by the Windows Mail client (Windows 7 and newer), and

contacts from Windows Live Mail contacts.edb database, also contacts from Windows Live Messenger's contacts.edb database..

You can also uncover various potentially relevant resources in 32-bit and 64-bit Windows PE executables (programms and libraries) as child objects, in particular RCDATA, named objects, bitmaps, icons and manifests. Useful for example for malware analysis. This does not happen automatically, only if you specifically target executable files via a suitable series of file masks.

Fully Base64-encoded files in the volume snapshot, provided that they have "b64" in the Type column can be automatically decoded, and the result is output in binary as (surprise) a child object.

Last not least this function can decompress hiberfil.sys files from Windows XP, Vista and 7 (32 and 64 bit) and automatically add the result to the case as raw memory dumps. hiberfil.sys slack (compressed data from previous usage of a hiberfil.sys file, as found near the end, if the last usage achieved stronger compression than previous usages) is provided as a child object in its decompressed form.

Generally all files produced by this function are added to the volume snapshot as child objects of their respective host files in which they were found. Files smaller than 65 bytes are not touched, for performance reasons.

Two separate file masks are maintained for uncovering embedded data in various file types. The second mask is optional and labelled as "special interest". For example malware investigators may choose to also process executable files that way when needed. You may prepend any element of a mask with a colon to temporarily exclude it, but keep it in the list for future reference. E.g. :*.jpg means *not* files with jpg as the extension or type.

In files of a type for which no internal extraction algorithm is built in, X-Ways Forensics tries to carve embedded data using those file header signatures that are marked in “File Header Signatures Search.txt” with the “e” flag. That means you can have X-Ways Forensics uncover embedded data in many more file types than it does by default if you like!

Extra precautions are taken to not produce duplicates of files that were already carved by the file header signature search. More precisely, the output of this function will replace corresponding carved files in the volume snapshot. The internal IDs of the carved files will remain the same, but additional metadata may become available (such as path/representation as a child object of the parent file, presumed original filename, more correct file size etc.). With the usual settings, this affects a considerable number of sector-aligned files.

File header signature search in all files not processed above

A separate sub-operation optional allows you to freely carve any kind of file within any file that is not processed by the first sub-operation. By default, file types with the "e" flag are selected for that. Use great caution to avoid delays and copious amounts of garbage files (false positives) and duplicates. Please apply this new function very carefully and only with a good reason to specifically targeted files only, such as swap files or storage files in which backup application concatenate other files without compression, not blindly to all files or random files. Remember

with great power comes great responsibility.

Signatures marked with the "E" flag (upper case) are never carved within other files, to prevent the worst effects, for example MPEG frames carved within MPEG videos, zip records carved within zip archives, .eml, .html and .mbox files carved within e-mail archives, .hbin registry fragments carved within registry hives. If you know what you are doing, of course you could remove the E flag.

There is an option to apply the carving procedure recursively, that is to also carve in files that were already carved within other files themselves. This can lead to many duplicates if the outer file at level 1 is carved too big so that files can be carved in it that were also carved at level 0 (the original file).

For situations were you want to carve embedded files that are not aligned at 512-byte boundaries in the original file, you may make use of the extensive byte-level option. Files are never carved in $MFT.

The default settings will make X-Ways Forensics conduct a file header signature searches at the byte level within pagefile.sys files, to find e-mail fragments, .lnk shortcut files, pictures, etc.

An unlabeled checkbox at the bottom allows you to turn on the verbose report mode that makes you aware of files that were previously carved at the general partition/volume level (i.e. by the File Header Signature Search) and that were output in the virtual directory for carved files, that have since been turned into child objects of other files because they seem to logically belong to them and are contained in them.

6.3.7 Capture Still Images from Videos

A forensic license allows to **sporadically capture still images from video** files in JPEG format. This happens either in a user-defined interval (e.g. every 20 seconds) that can be dynamically based on the play length of the video, or you can opt for a fixed number of video stills per video (1-255), no matter the play length. While fixed-length intervals result in number of stills that grows proportionally with the play length, the fixed absolute number limits your workload if you are going to look at all stills in the gallery, and also decreases the time to process long videos, but of course at the cost of being less thorough and an increased risk of missing something should any suspect hide relevant content somewhere within an innocuous video. X-Ways Forensics tries to extract a fixed number of stills evenly from all over the video to give a representative impression of it.

This functionality is applied to files whose type matches the specified file mask series. Requires an external program (MPlayer) and requires that the volume is associated with the active case. Pictures can be extracted from all the video formats and codecs supported by MPlayer. Useful if you have to systematically check many videos for inappropriate, illegal, or otherwise relevant content (e.g. child pornography or terrorist training camp instructions). The use of intervals ensures that you won't miss important parts of videos that are hidden in the middle of a harmless vacation or birthday party video.

Extracting pictures considerably reduces the amount of data, and looking at stills in the gallery is much faster, efficient and more comfortable than having to watch all videos one after the other. The potentially time-consuming extraction process can be run unattended e.g. over night beforehand.

Also useful if you need to include extracted pictures in a printed report. The first extracted picture at the same time optionally can serve as a preview picture for the video file in Preview and Gallery mode. ASF/WMV videos protected with DRM cannot be processed and are consequentially marked with e! in the Attr. column. Note that you may hear occasional sound from the videos. Please turn off sound on your computer if you wish to avoid this. Note also that if you select a small interval, you may not necessarily get additional pictures. This depends on how the video was encoded/compressed. If for example one particular video was encoded in such a way that it contains an I-frame (full image) only every 8 seconds and all the frames in between just describe the changes to reduce data, then you can expect no smaller still intervals than 8 seconds. That's because this function is meant to work fast and does not reconstruct exact frames at exact time indexes. If you need more detail, you can export all frames using the directory browser context menu. Duplicate stills are omitted when extracting pictures with MPlayer.

Once JPEG pictures have been exported from videos, the videos can optionally be dynamically represented in the gallery, with all extracted stills, showing them stills in a loop, to give a much more complete impression of the contents of videos without further user interaction (without having to explore them). Thus an alternative efficient way to review a large number of videos is this: Explore recursively, filter for videos, sort in descending order by number of child objects (so that videos with a similar number of stills are shown together), and activate Gallery mode. Watch the various video stills for each video. Proceed to the next gallery page when you are confident that no incriminating videos are represented on the current page, for example when all stills have been shown, which you will know is the case when the gallery has rotated back to the first still for each video.

A small amount of metadata is extracted from videos when exporting stills, usually coding/compression format, resolution, bits per pixel, frames per second, data rate per second for video data. That is in addition to the metadata that is provided by the regular metadata extraction.

6.3.8 Pictures Analysis and Processing

A forensic license additionally allows to compute the percentage of **skin colors** in pictures and to detect **black & white pictures**. This can be done for the file types JPEG, PNG, GIF, TIFF, BMP, PSD, HDR, PSP, SGI, PCX, CUT, PNM/PBM/PGM/PPM, ICO. The detection of black & white or gray-scale pictures is useful when looking for documents that were scanned and faxes that were stored electronically. A forensic examiner who has to look for traces of child pornography can sort pictures by skin color percentage in descending order to immensely accelerate the job. Checking the mass of 0%..9% skin color percentage pictures (e.g. thousands of browser cache garbage files) may not be necessary any more as the most likely incriminating files will be sorted near the top of the list. Please note that there may be false positives, i.e. skin-like colors of a non-skin surface. Pictures that cannot be correctly scanned for their color contents, e.g. because they

are too large or corrupt, will be listed with a question mark instead of the skin color percentage. Pictures with very small dimensions (width or height no more than 8 pixels, or width *times* height no more than you indicate) will be marked as irrelevant with the assumption that they cannot contain incriminating pornography or documents.

For large JPEG, PNG, GIF and TIFF files, at the same time when analyzing the colors in the pictures during volume snapshot refinement, X-Ways Forensics can optionally also create thumbnails in advance for much quicker display updates in Gallery mode later. Internal thumbnails are only created if no original thumbnails are embedded in the files and extracted at the same time, and they are actually utilized for the gallery only if auxiliary thumbnails are enabled (see Options | General). It is possible to specify your preferred resolution (maximum width or height in pixels) and quality (JPEG compression factor) of the thumbnails. However, the maximum amount of data that can be stored in the volume snapshot for a thumbnail is limited, to 64 KB, so if a generated thumbnail gets larger than that, X-Ways Forensics will automatically reduce the user-defined resolution accordingly. To discard all internal thumbnails, but keep the computed skin color percentages, you may delete the file "Secondary 1" in the "_" subdirectory of an evidence object behind X-Ways Forensics' back, i.e. when the evidence object is not currently open.

You can process picture files of certain types and resolutions with OCR. The resulting text can be stored in the volume snapshot for subsequent indexing and logical searches. OCR can optionally be restricted to picture files produced by/for certain device types, e.g. produced by a scanner, produced as a screenshots, or generated for printing, because such pictures are more likely to contain relevant text and because omitting other pictures can save time. Picture files for which device type recognition was unsuccessful ("unknown") or to which it was not applied because metadata extraction was not yet run or because device type recognition is not supported for the respective file type (resulting in a blank device type cell) can optionally be OCRed, too. OCR can optionally also be applied to pictures if the regular conditions (file type, resolution and device type) are not met, but if text is detected by the picture content analysis.

Excire: Photo analysis with artificial intelligence (separate chapter)

There is a small button on the right-hand side showing a pointing finger. Clicking that button will show user interface controls for usage of PhotoDNA, even if the functionality is unavailable, to give you an idea of how this module can be used. PhotoDNA is provided for free to users in law enforcement agencies.

If you have an internal PhotoDNA hash database, known photos can be recognized automatically even if visually altered. If you select more strict matching (allow less variation in a picture), the process can be noticeably faster in huge databases. Resulting matches can be seen and filtered in the combined Analysis column (only 1 match per file). Multiple matches for the same picture can optionally be output as labels. This is useful if you need to see *all* matches and/or if you wish to see PhotoDNA matches in the same place as ordinary hash database matches, which can also be output as labels. Please note that photos that are recognized via PhotoDNA already are not additionally checked for the amount of skin tone. PhotoDNA hash values are computed and matched only if the picture contains a total number of pixels that is larger than a user-defined minimum (width times height). This avoids database look-ups that can be time-consuming in very large PhotoDNA hash databases and typically have no benefit for small garbage pictures.

The minimum dimensions allowed as a condition are 50x50 pixels. The PhotoDNA algorithm intrinsically requires a certain minimum number of pixels to provide meaningful results. If you select the lowest possible strictness level for matching (level 1), you will be asked whether you are really certain, as that level is known to occasionally deliver false matches. That level is offered in X-Ways Forensics only because it is provisionally suggested by the original developers of PhotoDNA. The recommended and default level in X-Ways Forensics is level 3.

It is possible to more conveniently match pictures against the PhotoDNA hash database again, for example after having added some hash values to the database or after having assigned hash values to different categories, thanks to a checkbox simply labelled "Again". You can still uncheck the "Already done?" check box for the whole picture analysis and processing operation to also discard the results of the skin color computation and precomputed thumbnails and regenerate both plus the PhotoDNA matches from scratch. Please note that with the "Again" option when re-using previously computed PhotoDNA hashes, changes to the state of the check box "Recognize pictures even if mirrored" have no effect. That means if previously unchecked when hash values were computed for the first and stored in the volume snapshot, checking it later when re-using the stored hash values won't do any good.

Matching pictures against the PhotoDNA hash database another time is much faster if during a previous run you have X-Ways Forensics store the computed PhotoDNA hashes in the volume snapshot. Saves the time to read the files from the disk/image again and to decode/decompress the JPEG data or other formats again (time-consuming for high-resolution photos) and to recompute the hash values. Please note that PhotoDNA hashes require considerably more drive space than ordinary hashes. Also, more than one PhotoDNA hash may be required for just one picture. It is recommended to store the hash values in the volume snapshot for future fast re-matching only if you expect your PhotoDNA hash database to change during processing of a case, for example if it is likely that you or your colleagues discover further relevant pictures in that case, forcing you to search for other copies of these pictures. Files that got their PhotoDNA hash value(s) stored in the volume snapshot have more information in Details mode: The actual hash value(s) can be seen, as well as all the matches along with their respective accuracies.

When computing PhotoDNA hash values and storing the hashes for deduplication and fast re-matching, X-Ways Forensics now also automatically compares embedded thumbnails to their parent files. If the difference is noticable, that will be brought to the user's attention with two labels, "Thumbnail discrepancy" and "Thumbnail notable (data corrupt/incomplete)", where the latter means that there is a difference most likely just because the parent file is corrupt or incomplete. (The thumbnail, which requires little storage space and is located near the start of the file, could be unaffected and therefore helpful.) The former could indicate that someone has retroactively altered /redacted the full resolution picture and left the embedded thumbnail as it was.

To discard stored hash values you can either take a new volume snapshot, or alternatively you may delete the file "PDNA" in the "_" subdirectory of the evidence object, where the volume snapshot is internally stored.

If matches are returned from regular hash databases as well as the PhotoDNA hash database at the same time with conflicting categorizations, the "more severe" category prevails: unknown < known good < known, but uncategorized < known bad. The option to mark a file as already

viewed when it gets categorized as irrelevant is now applied to the combined result of ordinary hash database and PhotoDNA hash database matching.

6.3.9 Detection of Encryption

A forensic license allows to optionally perform **file format specific and statistical encryption tests**. With an entropy test, each existing file larger than 255 bytes is checked whether it is fully encrypted, i.e. from the very first to the very last byte. If the test is positive (the entropy exceeds a certain threshold), the file is flagged with "e?" in the attribute column, to indicate that it might deserve special attention. Typical example: Encrypted container files, which can be mounted by encryption programs like TrueCrypt, PGP Desktop, BestCrypt, or DriveCrypt as drive letters. The entropy test is not applied to ZIP, RAR, TAR, GZ, BZ, 7Z, ARJ, CAB, JPG, PNG, GIF, TIF, MPG, and SWF files, which are well-known to be compressed internally and therefore almost indistinguishable from random or encrypted data. This test is not needed to detect that files are encrypted at the NTFS file system level or inside archives.

Secondly, documents with the extensions/types .doc (MS Word 4...2003), .xls (MS Excel 2...2003), .ppt, .pps (MS PowerPoint 97-2003), .mpp (MS Project 98-2003), .pst (MS Outlook), .docx (MS Word 2007...2010), .xlsx (MS Excel 2007...2010), .pptx, .ppsx (MS PowerPointer 2007-2010), .odt (OpenOffice2 Writer), .ods (OpenOffice2 Calc) and .pdf (Adobe Acrobat) are checked for file format specific encryption; MS Office documents also for digital rights management (DRM) protection. If positive, these files are flagged with "e!" in the attribute column. This check requires that the separate viewer component is active. X-Ways Forensics also automatically tries the passwords in the current case's password collection with such files and remembers the matching password, if any, in the file's Metadata cell, for future use when viewing/previewing the file and for the user's information.

Additionally, the encryption test can detect eCryptfs-encrypted files (files stored by the Enterprise Cryptographic File System for Linux), with a test that is based on eCryptfs implementations for Ubuntu 8.10, 9.04, 9.10 and 10.04. Such files will by marked with "E" in the Attributes column, just like EFS-encrypted files in NTFS.

6.3.10 Indexing

Available only with a forensic license. Reads the data with the same logic as a logical search, with the same advantages (see that topic).

Creates indexes of all words in all or certain files in the volume snapshot, based on characters you provide, based on the Unicode character set and/or up to two code pages that you select. It is possible to have up to three such indexes per evidence object (e.g. Cyrillic characters indexed in Unicode and two Cyrillic code pages). X-Ways Forensics allows you to conveniently select characters from more than 22 languages for indexing. Currently, most European and many Asian languages are predefined, e.g. German, Spanish, French, Portuguese, Italian, Scandinavian languages, Russian, South Slavic languages, Eastern European languages, Greek, Turkish, Hebrew, Arabic, Thai, Vietnamese. You may specify each and every character explicitly, or

specify ranges of characters that can optionally be followed by additional single characters (e.g. a-zA-Zäöü) if the edit box for the character pool starts with "range:". To index the dash itself (not recommended), specify it as the last character in the edit box.

Indexing is a potentially time-consuming process and may require a large amount of drive space (rule of thumb for default settings and average data: 5-25% of the original amount of data). However, the index will allow you to conduct further searches very quickly and spontaneously. The index files are saved in the subdirectories of the metadata folder of the corresponding evidence object. The scope of the index, i.e. which files are to be indexed, can be fine-tuned. Note that the index of partitioned media such as physical hard disks solely covers unpartitioned areas. That's because each partition can have its own index.

Words shorter than a lower limit you specify are ignored. The longer the minimum length in characters, the smaller the index and the faster the indexing procedure. The default lower limit is 4 characters. Frequent irrelevant words can be excluded from the index in the exception list with a minus prefix (e.g. -and, if 3-letter words are already accepted), which reduces the size of the index and the time needed to create it. The larger the range of accepted word lengths, the larger the index becomes and the more time indexing takes. Important 3-letter words can be added to the exclusion list with a plus prefix (e.g. +xtc), which overrides the default lower limit of 4 characters. The exception list does not have to be sorted alphabetically. Words in the exception list longer than the upper limit you specify are truncated in the index. Words in the exception list are bound by the character pool and cannot contain different characters.

X-Ways Forensics can optionally distinguish between uppercase and lowercase letters, i.e. create a case-sensitive index. This can be useful e.g. if you create the index for the purpose of later exporting a word list for a customized dictionary attack.

If you have X-Ways Forensics include substrings in the index, this will further slow down index creation (by a factor of 3 to 5) and inflate the index, however, you will later be able to find e.g. "wife" in "housewife" and "solve" in "resolve". If you do not include substrings in the index, it will still be possible to search the index for substrings later, but the result will be incomplete, and the search speed much slower. Please note that it is the responsibility of the user to enable substring indexing if the words in the language to index are not delimited with spaces (e.g. in Chinese, Japanese or Thai).

Indexing will be unnecessarily slow if the data to be indexed resides on the same disk with the case file and directory, where the index is created. Try to avoid indexing with an active Internet connection if your Windows system is configured to download updates and reboot automatically upon installation.

Optionally, text in certain file types can be decoded for indexing (cf. Logical Search), and it is possible to create indexes for multiple selected computer media/images associated with a case in a single step. You can index in up to six different code pages simultaneously.

It is possible to define a character substitution list in Unicode that causes certain letters to be indexed as other letters (e.g. "é" as just "e"), *if* they are indexed (that still depends on whether they are listed in the character set above). This will allow you to find certain spelling variations with a single search term, e.g. both the name "René" with an accented e at the end and "Rene"

without, with either spelling. This list must have the structure
é>e
è>e
à>a
...
(i.e. 1 substition per line) and needs to be present as a Unicode text file named "Character Adjustment.txt" that starts with the LE Unicode indicator 0xFF 0xFE. "Character Adjustment.txt" is an optional file and expected in the X-Ways Forensics installation directory.

You will be warned if you define a space character as part of words. That is because space characters are meant to delimit words, they are not part of the words themselves. If a space character is defined to be part of words, that means a whole sentence like "Mike Smith lost his credit card yesterday." is considered just a single word.

You can delete all indexes for an evidence object by removing the "Already done" check mark in the Refine Volume Snapshot dialog. This will also clear the "i" flag from all indexed files in the volume snapshot.

Search in Index: After indexing files, you may search the index for keywords very quickly, using the Simultaneous Search function. Select "Search in Index" from the drop-down box at the bottom. Anything in excess of the maximum word length used for indexing is ignored (so that "ridiculous" is found in the index even if in the index that word was truncated to "ridicul" based on a maximum word length of 7 letters). X-Ways Forensics does not distinguish between uppercase and lowercase letters except if a case-sensitive index was created. In a search hit list populated by an index search, physical offsets are not available.

You may conveniently run non-RegEx index searches for search terms that contain space characters, just like in conventional searches. This is very important for names (e.g. "John Doe" or "XYZ Technology Ltd") and spaced compound words (e.g. "bank account" or "credit card limit"). This works even if the individual components of the compound already exceed the maximum word length that was indexed (by default 7 characters), so that you will have no trouble finding "basketball positions" (10+9 letters) or "skyscraper architecture" (10+12 letters). Just as always the components are only matched up to the length that was indexed, which is not a big problem because there are not many words other than "basketball" and "skyscraper" that start with "basketb" or "skyscra", respectively. In fact the spaces in the search terms match unindexed word delimiters other than spaces as well, such as hyphens, so you will also find "Spider-Man" and "freeze-dried" when searching for "spider man" and "freeze dried", or underscores as in "bank_account" (think of a filename like "bank_account.html"), or plus signs as in "credit+card" (e.g. common in Google search URLs when searching for more than 1 word), or periods as in "interview.pdf". So in that respect index searches are even more powerful than conventional searches. Defining spaces as being part of words is a big no-no.

6.4 More Information about Volume Snapshot Refinement

Should processing freeze on a certain file, note that the internal ID and the name of the currently processed file are displayed in the small progress indicator window. If the volume snapshot

refinement is applied to an evidence object and the refinement crashes when processing a single file at a time, X-Ways Forensics will tell you which file when you restart the program and label it with “Reason for crash?” (depends on the Security Options). All that happens so that you can exclude and omit the file when trying again. It does no harm (does not create duplications and does not cost much time again) if you restart snapshot refinement for that volume from scratch, as already processed files will quickly be skipped, up to the point where the refinement progress was last saved, which depends on the auto-save interval of the case. The volume snapshot remembers for each file separately which operations of the volume snapshot refinement have been applied to it already, so the same operations will usually not be applied again to the same file.

If the hash value for a problematic (crashing) file was computed, that file and identical files are skipped automatically if you (continue to) refine the volume snapshot and compute hash values (at least if the protection against identical crasher files is active in the properties of the case). To make the case forget previous crasher files, click the Delete button in the case properties. Skipped files also automatically get the aforementioned label.

The file processing part of volume snapshot refinements supports multiple threads (only if not applied to a selection). Depending on the selected suboperations and the types of the files in the volume, and depending on I/O speed, this can double, triplicate or even quadruplicate the performance. The faster your mass storage solution (HDD, SSD, RAID) in terms of seek times and data transfer speed, the more time you save percentage-wise. This parallelization feature is still considered experimental and not complete yet, but the potential time saving in one of the most important and most time-consuming functions of the program is enormous. Selecting multiple extra threads has an effect only when searching in evidence objects that are images or directories, not disks. If you select 0 extra threads, it will work as in X-Ways Forensics versions before 19.0. If you select 1 or more extra threads, processing is done in additional worker threads (as many as you select), and the main thread of the process will be idle, which means the GUI will remain highly responsive. In X-Ways Investigator up to 3 worker threads may be used, in X-Ways Forensics up to 16, if your CPU supports that. If multi-threaded processing crashes, next time when you restart the program it probably cannot tell you which file exactly presumably caused the crash. File-wise processing conducted by X-Tensions (through calls of XT_ProcessItem or XT_ProcessItemEx) are also parallelized if the X-Tensions identifies itself as thread-safe. Processing of files in file archives is currently excluded from parallelisation internally.

You can make a backup of the volume snapshot automatically once refinement has completed, so that you can quickly return to that state if necessary instead of taking a new volume snapshot and refining it again. Useful for example if you make some mistake in your manual review of files or if the volume snapshot gets corrupted somehow. If the checkbox for this is fully checked instead of only half checked, an intermediate additional backup if made once the operations of step 1 (at the disk/partition level) have completed. The menu command to restore volume snapshot backups can be found in the context menu of the evidence object in the Case Data window.

You may schedule a simultaneous search in advance for the time after the volume snapshot refinement.

6.4.1 Interdependencies

There are various interdependencies between all these operations. For example, if the contents of archives are included in the volume snapshot, among these files there could be pictures that are to be checked for skin colors, or documents that are to be checked for encryption. You can work under the premise that if an additional file is added to the volume snapshot or if the true type of a file is detected as part of Refine Volume Snapshot, all the appropriate other operations are applied to that file, *if they are all selected*. The output of one operation automatically becomes the input of all other operations (or even the same operation again), where suitable.

Imagine someone tries to conceal an incriminating JPEG picture by embedding it in a MS Word document, misnaming that .doc file to .dll, compressing that file in a Zip archive, misnaming the .zip file to .dll, compressing that .dll in another Zip archive, misnaming that .zip file again to .dll, and then sends this .dll file by e-mail as an attachment using MS Outlook. If all the respective options are selected, Refine Volume Snapshot does the following: It extracts the e-mail attachment from the PST e-mail archive. It detects that the .dll attachment is actually a Zip archive. Then it includes the contents of it in the volume snapshot, namely a file with the .dll extension. That file is found to be actually another Zip archive. Consequently that archive will be explored, and the .dll file inside will be detected as a .doc file. Searching for embedded pictures, X-Ways Forensics finds the JPEG file in the .doc file and can immediately check it for skin colors if desired. All of this happens in a *single* step.

6.4.2 Notes

X-Ways Forensics conveniently remembers for each and every file in the volume snapshot which refinement operations have already been applied to it, so that the file will not unnecessarily be processed again, which would lead to undesirable duplication of child objects, waste of time etc. X-Ways Forensics does not remember the individual suboptions of each operation (e.g. whether "Create previews of browser databases" was selected for the metadata extraction) and cannot catch up on these suboptions individually. The only operations that will always be applied repeatedly are indexing and matching of hash values against conventional hash databases. If for any reason you wish to apply certain other operations again to the same file (e.g. then with different suboptions or after having updated the signature database for file type verification), you may find an "Again" check box or completely reset a file to the state of "still to be processed" by volume snapshot refinement, by selecting it and pressing Ctrl+Del. This will also clear any computed skin color percentages, extracted metadata, hash values, hash matches, etc. This will also discard extracted text from the volume snapshot, so that running the text decoding + OCR operations again (e.g. after adjusting "PDF Requiring OCR.txt") will make another attempt. However, this function does not remove any child objects from the volume snapshot. That would have to be done by the user separately, if desired, by excluding and removing them. Neither does this function delete any events that were created during prior refinement operations. Another keyboard shortcut, Ctrl+Shift+Del, allows to remove matches with ordinary hash sets, FuzZyDoc hash sets, and PhotoDNA categories from selected files in the volume snapshot, which even if the hash sets are deleted from the hash database are not discarded otherwise, plus removes the "Duplicates found" marker in the Description column.

Whether a file should be processed by volume snapshot refinement or not is decided only at the time when it is that file's turn, not when you start the operation. That means if you continue to work in the program while a volume snapshot refinement is ongoing, and alter or activate or deactivate filters or tag or untag files or exclude or include files, that may still affect the scope of the operation, depending on the chosen options and depending on whether the files that you tag/untag/exclude/include/... still have to be processed or not. So if for example you find out that the operation takes too much time, you can still make the filter more strict or untag certain very large files etc., without interrupting the process.

When volume snapshot refinement is in the stage of processing individual files, then the progress percentage is simply the internal ID of the currently processed file divided by the total number of items in the volume snapshot. X-Ways Forensics doesn't know beforehand which files need a lot of time to process, only when actually reading from the file it will be decided what should be done with the file and discovered how much data is embedded etc. File type verification and potentially hash database matching may change the decision about what to do with the file, if anything at all. If an entire evidence object consists of just 1 file, e.g. if you added a single files to the case, then the progress percentage will not advance. The progress is 0% initially and 100% for a fraction of a second when done. The displayed percentage does not reflect the sub-progress within a given large file.

An unlabelled (but tooltipped) check box in the volume snapshot refinement dialog window can now make X-Ways Forensics reveal which suboperation is currently applied to the currently processed file. A 3-digit abbreviation will be displayed with the following meaning:
Sig: file type verification
Hsh: hashing
Vid: capture sporadic still images from videos
Idx: preprocessing original file contents for indexing
Dec: text decoding for indexing
IdX: preprocessing decoded text for indexing
Emb: search for embedded data
PDN: PhotoDNA database matching
Pic: other picture analysis steps
Eml: e-mail extraction
Fuz: FuzZyDoc database matching
Met: metadata extraction
Enc: file format specific encryption test
Ent: entropy check
Arc: inclusion of files in archives into the volume snapshot
This may be helpful for educational reasons, to give users a better idea of how computationally expensive certain suboperations are and how much time could be saved by not selecting them if not absolutely necessary. It may also prove useful for debugging purposes. Whether this option may slow down processing on certain computers has not been tested.

Certain previously valid timestamps of files are output as events during various suboperations of the particularly thorough file system data structure search on NTFS, depending on the refinement option "Provide by-catch timestamps from various sources as events", which may also effect other operations whose primary purpose is not the retrieval of timestamps/events.

7 Some Basic Concepts

7.1 Edit Modes

The info pane displays for each file/disk, in which mode it was opened in the program. The info pane's context menu allows to selectively change the edit mode of the active window.

Read-only/View mode: Recommended for computer forensic examinations. Files or disks that are opened in view mode cannot be (intentionally or accidentally) edited/altered, only viewed. In other words, they are opened write-protected = read-only. In order to enforce strict forensic procedures, this is the only mode available in X-Ways Forensics for disks or images interpreted as disks, also for files that you open from drive letters that are protected (see Security Options), meaning you could still process (decode, decrypt, convert, ...) files that are stored the current case's directory or in the general folder for temporary files etc.

Default edit mode: Modifications to files or disks opened in default edit mode are stored in temporary files. Those temporary files are created and maintained dynamically when needed. Only when you close the edit window or use the Save menu command the File Menu, the modifications are flushed and the original file or disk is updated, after prompting the user.

In-place edit mode: Please use caution when opening files or disks in in-place edit mode. All kinds of modifications (keyboard input, filling/removing the block, writing clipboard data, replacements, ...) are written *to the original file or disk* ("in-place") *without prompting*! It is not necessary to save the file manually after having modified it. Instead, the modifications are saved lazily and automatically, at latest when closing the edit window. However, you may use the Save command to ensure the buffer is flushed at a given time.

The in-place edit mode is preferable if the data transfer from the original to the temporary file and vice-versa, which is obligatory in default edit mode for certain operations, consumed too much time or disk space. This may be the case when opening very large files or when modifying huge amounts of data. Since usually no temporary files are needed in in-place edit mode, this edit mode is generally faster than the default edit mode. The in-place edit mode is the only mode available when using the memory editor. Hint: Even in in-place edit mode the creation of a temporary file is unavoidable when altering the file size.

If you open files using the operating system (e.g. via File | Open, from any drive letter currently available in Windows), then operating system file write commands will be used to change a file on the disk. However, in WinHex it is even possible to edit files without using operating system file write commands, directly on a disk/in a raw disk image in any file system supported, even if that file system is not known to Windows, even files not seen by Windows (e.g. deleted files), even in partitions not seen by Windows (e.g. by damaged or deleted), without changing any timestamps or attributes, in in-place mode only. For this editing capability, the file must been opened from within the already opened volume that contains it, either via the Open command in the directory browser context menu or in File mode (forensic license only). Compressed files or generally files within other files (e.g. e-mails and attachments in e-mail archives) cannot be

edited, except in an evidence file container if they have been copied there from the original disk/image. Note that files cannot be shortened or expanded that way, only the data in already allocated areas can be modified. Editing files opened directly from within disks/raw images as described above is possible in WinHex only, not in X-Ways Forensics or X-Ways Investigator, where sector level write access (to which file editing is internally translated) is disabled and where the only mode available for disks and interpreted images and files opened from within volumes is read-only mode. X-Ways Forensics can be easily ran as WinHex if preferred (simply rename the .exe file).

In forensic computing, electronic discovery and IT security, this editing capability can be helpful to manually redact (e.g. overtype) specific data that should not be examined/disclosed/seen or to securely erase specific areas within files (e.g. define as a block and fill the block). Note that evidence file containers are raw images if they have not been converted to the .e01 evidence file format and thus allow for retroactive file editing, which, however will invalidate any accompanying hash values. It is even possible to edit directories, i.e. the clusters with directory data, e.g. INDX buffers in NTFS, for example if you need to redact the names of certain files.

7.2 Scripts

Some of the functionality of WinHex can be used in an automated way, e.g. to speed up recurring routine tasks or to perform certain tasks on unattended remote computers. The ability to execute scripts other than the supplied sample scripts is limited to owners of professional licenses or higher. Scripts can be run from the Start Center or the command line. While a script is executed, you may press Esc to abort.

WinHex scripts are text files with the filename extension ".whs". They can be edited using any text editor and simply consist of a sequence of commands. It is recommended to enter one command per line only, for reasons of visual clarity. Depending on the command, you may need to specify parameters next to a command. Most commands affect the file or disk presented in the currently active window.

See Appendix B for a description of currently supported script commands.

7.3 X-Tensions API

Automate investigative tasks and extend the functionality of X-Ways Forensics with *X-Tensions*: The X-Ways Forensics X-Tension API (application programming interface) allows you to use many of the advanced capabilities of the X-Ways Forensics computer software programmatically and extend them with your own functionality. For example, you could implement some specialized file carving for certain file types, automated triage functionality, alternative report generation, or automatically filter out unwanted search hits depending on your requirements etc.

Among other things, X-Tensions allow you to:
- read from a disk/partition/volume/image
- retrieve abundant information about each file and directory in the volume snapshot

- read from any file
- create new objects in the volume snapshot
- label files
- add comments to files
- process, validate and delete search hits
- and do practically *everything else that is possible with a Windows program*! (thanks to the Windows API)

You can use your programming language of choice, e.g. C++, Delphi, or Visual Basic, and do not have to learn any new programming language. You can use your compiler of choice, for example Visual Studio Express (freeware).

Since an extension is not an interpreted script, but regular compiled executable code that is running in the address space of the application itself, you can expect highest performance, the same as with internally implemented functionality. X-Tensions give you easy and direct access to crucial and powerful functions deep inside X-Ways Forensics.

When X-Tensions functions can get called:
- when refining the volume snapshot
- when running a simultaneous search
- via the directory browser context menu
- via the search hit list context menu

The X-Tension API also allows the development and use of so-called Disk I/O X-Tensions. These are snap-ins that sit between all analysis functionality and the user interface of X-Ways Forensics on the one hand and a disk/image/RAID/partition/volume from which sectors are read on the other hand. They can for example deal with full disk encryption and decrypt the data in all sectors read by X-Ways Forensics on the fly when needed, so that all relevant functions only get to see the decrypted data and can deal with it as if it was a normal disk/volume.

The user may open a selected evidence object through such a Disk I/O X-Tension using a command in the context menu of the Case Data window. After selecting the intended X-Tension DLL, if the DLL signals that it can successfully deal with the data in that evidence object, the case will remember which DLL that was chosen and automatically apply it next time when opening the same evidence object. Note that as always partitions count as evidence objects themselves. That way full disk encryption can be tackled as well as volume level encryption.

Once completely run, the user is prompted whether or not stubborn C# X-Tension DLLs should be completely unloaded after execution. Programmers may prefer to do that when debugging their own X-Tensions, but apparently this can prevent usage the same DLL a second time in the same session of X-Ways Forensics, so ordinary users better choose No.

You may distribute your X-Tension DLLs that you compile and/or your source code free of charge or even for a fee, under whatever license terms you see fit.

For more information please see http://www.x-ways.net/forensics/x-tensions/api.html.

7.4 Sector-wise Access to Storage Devices / Disk Editor

The Open Disk command in the Tools menu and the Add Disk command in the Case Data window allow you to open locally attached physical storage storage devices as well as volumes represented by a drive letter with sector level access. Note that if you open a physical, partitioned disk, afterwards you can open a partition on that disk by double-clicking that partition. That representation of a partition includes volume slack (surplus sectors that don't add to another full cluster), if any is present, which a logical drive letter does not.

The list of logical volumes can optionally include volumes that are active in Windows, but not currently associated with any drive letter. Active volumes that are not ordinary volumes are displayed with a special icon and a special description, e.g. "TrueCryptVolumeX". Useful so that on a live system that you wish to preview, examine or acquire you can quickly see which volumes may need to be addressed separately (in additional to physical storage devices) because it would be difficult to reconstruct or unlock them later based on the data on the physical storage device. If volumes without connected drive letter are listed, that also includes volumes that have been mounted within Windows as a junction point in another volume. Such volumes are listed with a special link icon, and the junction point is displayed between volume label and volume size. The list of volumes that do not have drive letters may now also include volumes that were previously active in Windows. Those are marked with a crossed out red circle icon. For example a previously mounted TrueCrypt volume that was dismounted might be shown in this fashion. Such volumes cannot be opened any more, they are just listed for informational purposes, which is useful when working on a live system that needs to be examined.

Usually it is preferable to work with a volume instead of the physical storage device as a whole because more features are provided in this case. For example, "clusters" are defined by the file system, the allocation of clusters to files (and vice versa) is known to WinHex, "free space" and "slack space" have a meaning. If you need to edit sectors outside a logical drive (e.g. the master boot record), if you wish to search something on several partitions of a hard disk at the same time, or if a partition is damaged or formatted with a file system unknown to Windows, so Windows is unable to make it accessible as a drive letter, you would open the physical disk instead. From the window that represents a physical medium you can usually also open individual partitions, by double-clicking them in the directory browser of that window. WinHex understands conventional MBR partitioning, GPT (GUID partition type), Apple partitioning, superfloppy format, Windows dynamic disks as organized by the LDM (Logical Disk Manager, MBR and GPT style), LVM2 (MBR and GPT style) and PC-compatible BSD disklabel. All dynamic volume types are supported: simple, spanned, striped, and RAID 5. Holding the Ctrl key when opening hard disks disables detection and special handling of dynamic volumes and ensures the hard disk is treated like it has been partitioned in the conventional way. Some of the aforementioned partitioning types are supported with specialist and forensic licenses only.

A function to decrypt BitLocker volumes without the help of the Windows operating system is available in WinHex Lab Edition, X-Ways Imager, X-Ways Investigator and X-Ways Forensics. This requires that you have and enter (e.g. copy & paste) the right password or recovery key or startup key, if one of those is actually required to decrypt the volume (not in case of clearkey encryption). The option to enter a password or key is not given in X-Ways Investigator. However, X-Ways Investigator can use a password or recovery key that was already entered for a

particular evidence object in a case by someone using X-Ways Forensics, so users of X-Ways Investigator can work on a case that includes BitLocker volumes if that case was properly set up for them by a colleague. X-Ways Forensics can try the passwort list of the active case when prompted for the password of a BitLocker volume. Before trying the password collection in Passwords.txt on a BitLocker volume, X-Ways Forensics first attempts decryption with the passwords of other BitLocker volumes in the same case, if there are any, for which a password is known. For that reason it can be beneficial to unlock BitLocker volumes a known password instead of a known recovery key when prompted if you have a choice. The verified password or recovery key of a BitLocker volume is saved in the Description box of the evidence object properties for future reference. Startup keys are stored in .BEK files, which in turn are usually stored on removable USB storage devices. Whenever X-Ways Forensics encounters a .BEK or recovery key file in any evidence object while taking the volume snapshot, it copies the file to the case directory and keeps it there. (The case directory, not the directory for cases.) In that directory .BEK and recovery key files are automatically found whenever a BitLocker volume is opened to see if any of them fits. You can also manually copy such files that you have found into the same directory to get X-Ways Forensics to try them. A security option controls whether BitLockers passwords and keys that you enter manually or that are found automatically (BEK and recovery key files) or that match when trying out passwords from a list are centrally stored in the case (on disk). That is convenient and active by default, but perhaps not desirable for internal investigations.

There is an optional raw mode for optical discs that allows to read from audio CDs and also the complete 2352-byte sectors on data CDs (CD-ROM and Video CDs) that contain error correction codes. If a physical storage device is treated as offline or read-only in Windows Disk Management, that information is displayed in all disk selection dialog windows. Offline disks can be opened for reading/imaging/analysis, but they are write-protected.

Please note the following limitations:

- Administrator rights are needed to access sectors on any kind of media. Under Windows Vista and later you need to run the program as administrator specifically, just being logged on as administrator is *not* sufficient.
- Remote (network) drives cannot be accessed sector-wise.
- X-Ways Forensics cannot edit disk sectors or sectors in interpreted images at all, only WinHex can.
- WinHex cannot write to CD-ROM or DVD.
- Under Windows Vista and later, WinHex cannot write sectors on the partition with the active Windows installation and on the partition where WinHex is running from.

The appendix C of this manual provides you with specifications of the master boot record, which can be edited using the disk editor.

Save Sectors: To be used analogously to the Save command for files. Part of the File menu. Writes all modifications to the disk. Please note that, depending on your changes, this may severely damage the integrity of the disk data. If the corresponding undo option is enabled, a backup of the concerned sectors is created, before they are overwritten. *This command is only available in the full version.*

7.5 Memory Editor/Analysis

The memory editor allows to examine the physical RAM/main memory and the logical memory of a process (i.e. a program that is being executed) in a live system. All memory pages committed by a process are presented in a continuous block. Unused (free or reserved) pages are ignored by default, but optionally included and displayed with "?" characters. With no such gaps, you may compare memory dumps to files exactly with one another (absolute and virtual addresses are identical), e.g. to examine stack and heap states or observe virusses.

If you expand one of the listed processes in the list, you may open either the so-called primary memory or the entire memory of this process or one of the loaded modules (DLLs). The primary memory is the lower part of the address range, below the area where system DLLs are loaded. Usually it also contains the main module of a process (the EXE file), the stack, and the heap. The "entire memory" contains all the allocated pages in the entire logical memory address space of a process.

With the 64-bit edition of WinHex/X-Ways Forensics you can get loaded modules above the 4 GB barrier in 64-bit processes listed, and read and edit memory in such address ranges. Unicode is supported for process and module names and paths in the memory editor. Page boundaries are represented by horizontal lines. Boundaries that represent gaps between contiguous allocated regions are represented by darker horizontal lines. The Info Pane shows information such as the maximum address represented and the number of allocation gaps (=number of contiguous allocated page ranges -1) as well as protection status and type of the currently displayed page.

Please note the following limitations:

- Access to *physical* RAM under Windows XP (32-bit) only, no more than 4 GB, and with administrator rights only
- Caution: Only keyboard input can be undone!
- Editing is possible in in-place mode only.
- The evaluation version only supports view mode.

The options relevant for the memory editor are "Check for virtual memory changes" (Options | Security) and "Logical addresses in memory editor" (Options | General).

Main Memory Analysis

Requires a forensic license. When you open the local physical RAM (via Tools | Open RAM, only under Windows XP) or a main memory dump as a file (and interpret that file exactly like you would a disk image) or add a memory dump to a case, processes will be listed in the directory browser, even hidden processes, with their timestamps and process IDs, and their own respective memory address spaces can be individually viewed in "Process" mode, with pages concatenated in correct logical order as seen by each process. The "particularly thorough data structure search" is signature-based, will take a little longer than taking a standard volume snapshot and may turn up traces of additional processes including rootkits. Memory can be acquired remotely with the help of F-Response (Tools | Open Disk). The analysis is supported for most (but not all) variants (service packs) of Windows 2000, Windows XP, Windows 2003 Server, Windows Vista, Windows 2008 Server, and Windows 7, 32 bit and (less complete) 64 bit.

Only complete memory dumps are supported, those which include regions in RAM that are utilized by the BIOS and by PCI devices.

Windows kernel data structures and named objects are conveniently listed in a tree in the volume snapshot under “Objects”. Loaded modules are listed under “Modules”. That enables X-Ways Forensics to allocate the memory pages in RAM mode that they occupy to them, and to compute hashes for them so that they can be identified via special hash sets. For hashing purposes it is recommended to list the invariant headers of loaded modules only (see Volume Snapshot Options).

The technical details report informs you of important system-wide parameters as well as of the current addresses of important kernel data structures and loaded kernel modules. In Details mode you can find the addresses of process-related data structures for each process and the ID of its parent process. In RAM mode, the Info Pane shows for each memory page a process to which it is allocated (if any) and its memory management status.

With the appropriate background knowledge, this functionality can be used learn more about the current state of the machine and its processes, sockets, open files, loaded drivers, and attached media, to identify malware, to find the decrypted version of encrypted data, to analyze network traces in incident response, and to do further research in the field of memory forensics.

7.6 Template Editing

A template is a dialog box that provides means for editing custom data structures in a more comfortable and error-preventing way than raw hex editing does. Editing is done is separate edit boxes. Changes take effect when pressing the **ENTER** key or when quitting the template after being prompted. The data may originate from a file, from disk sectors, or from virtual memory. Especially when editing databases, you may prefer to define a custom template for ease of access to the records. You will find the command to print a template in the system menu.

A *template definition* is stored in a text file with the extension .tpl. The *template editor* enables you to write template definitions and offers syntax checking. A template definition mainly contains variable declarations, that are similar to those in source code of programming languages. The syntax is explained in detail in Appendix A. The supported data types include all the common integer, floating-point and boolean variants, date types, hex values, binary, characters, and strings type. Arrays of both single variables and groups of variables can be used.

The ability to move freely forwards and backwards within the data makes using templates particularly flexible:

- The same variable may be interpreted and manipulated in several ways.
- Irrelevant data sections can be skipped.

The *template manager* lists all text files in the WinHex directory that contain template definitions. The title of the template along with a description, the filename, and the date and time of the last modification are shown. Click the Apply button to display a template using the selected template definition for the data in the current editor window at the current position. You

may also create a new template definition, delete or edit an existing one.

WinHex comes with several sample templates.

8 Data Recovery

8.1 File Recovery with the Directory Browser

Most obviously, deleted files and directories that are listed in the directory browser can be recovered easily and selectively with the directory browser's context menu. You navigate to a directory (or explore the root directory recursively), select the files to recover, and use the Recover/Copy command in the context menu. See chapter "directory browser". Ideally you refine the volume snapshot first, so that more previously existing files are found and listed in the directory browser.

8.2 File Recovery by Type/File Header Signature Search

Data recovery function in the Disk Tools menu, and also a strategy to find previously existing files as part of the Refine Volume Snapshot command. This recovery method is also referred to as "file carving". It searches for files that can be recognized by a characteristic file header signature (a certain sequence of byte values). Because of this approach, file carving does not depend on the existence of functional file system structures.

File Recovery by Type: Files found based on a file header signature are carved and stored in the output folder that you specify on one of your own drives. Optionally, recovered files of each type are put into their own subfolder (...\JPEG, ...\HTML, etc.). The presumed contents of the files are actually copied.
File header signature search: Files found based on a file header signature are not stored anywhere, but merely *listed* in a dedicated virtual directory of the volume snapshot. Only a reference to the file is stored, with an artificially generated name (based on either an incrementing number or the start sector number), presumed size, start offset. The file contents are read from the original disk/image on the fly when needed to view/copy the file. Optionally, you can output files from separate file header signature search operations into separate subdirectories, so that it's easier to distinguish between them if needed.

Note that file carving generally assumes contiguous file clusters, so it produces corrupt files in case the files were originally stored in a fragmented way. The following exception exists: If the file header signature search in volumes with a supported file system other than Ext2/Ext3 finds the start of a file in free space, at a cluster boundary, the data is by default assumed to flow around potentially following clusters that are marked by the file system as in use. This will correctly reconstruct files that were created after and stored around other files and then deleted, as long as the released clusters were not re-used and overwritten afterwards. To prevent file

carving purely in free space this way, i.e. assume contiguous clusters, you can unselect the option “Carve files in free clusters around used clusters”. As an example, consider 9 consecutive clusters in a partition: f1 f2 f3 u1 u2 u3 f4 f5 f6. These are 3 free clusters, followed by 3 used clusters, followed by 3 free clusters. Carving in purely free space "around" used clusters can give you a file that consists of f1 f2 f3 f4 f5 f6. Without the "around" option you can get a carved file that consists of f1 f2 f3 u1 u2 u3.

The option “Ext2/Ext3 block logic” causes this recovery method to deviate from the standard assumption of no fragmention as well, in that it will follow the typical Ext block pattern, where e.g. the 13th block from the header of the file is considered an indirect block that references the following data blocks. This option has no effect when applied to partitions that WinHex knows have a file system other than Ext2 and Ext3 or when a header is found that is not block-aligned.

A log file "File Recovery by Type.log" about the selected parameters and the recovery results is written to the output folder for verification purposes.

You can expand or collapse the entire file type tree in this dialog window with a single mouse click on the appropriate button. That is useful because when expanded you only need to type the first few characters of the file type description to automatically jump to the first matching item in the tree.

Since no use is made of a possible presence of a (consistent or damaged) file system, the original *file sizes* are principally *unknown* to this recovery method, and so are the original *filenames*. That is why the resulting files are mostly named generically according to the following pattern: Prefix#####.ext. "Prefix" is an optional prefix you provide. #####" is an incrementing number per evidence object. "ext" is the filename extension that corresponds to the file header signature according to the file type definition. The output filename prefix may optionally contain a placeholder "%d", which will be replaced by the drive name. This is useful if you apply File Recovery by Type to multiple drives at a time and wish to be able to easily distinguish files from different drives.

With a specialist license or higher, the "intelligent naming" option will cause Exif JPEG files to be named after the digital camera model that created them and their internal time stamp, if available. Many Windows Registry hive files are given their original names, also some JPEG files in whose metadata Photoshop has embedded a name. JPEG files without known name and no Exif metadata that however have been created by a known library receive some additional information in their artificial names in parentheses (see generator signature). Thumbs.db files are always named thumbs.db, index.dat always index.dat. The aforementioned prefix is not used in conjunction with original filenames.

Various algorithms are at work internally that try to determine the original sizes of files of many different types (among others, JPEG, GIF, PNG, BMP, TIFF, Nikon NEF, Canon CR2 raw, PSD, CDR, AVI, WAV, MOV, MPEG, MP3, MP4, 3GP, M4V, M4A, ASF, WMV, WMA, ZIP, GZIP, RAR, 7Z, TAR, MS Word, MS Excel, MS PowerPoint, RTF, PDF, HTML, XML, XSD, DTD, PST, DBX, AOL PFC, Windows Registry, index.dat, Prefetch, SPL, EVTX, EML) by examining their data structure. This applies to entries in the file type definition database that have a "~" in the Footer column. These entries should not be altered in order for the size and type detection to work for these file types. Alternatively, a footer signature can also help to find the

end of a file. Files for which neither an internal algorithm nor a footer signature definition exists or file about whose original size the available internal algorithm has no idea and for which no footer signature is actually found, are recovered at the default size specified in the file type definition database in bytes. Be generous when specifying such a size because whereas files recovered "too large" can still be opened by their associated applications, prematurely truncated files often can't be as they are incomplete. The attempt to detect the original size of files of certain types by searching for a footer is limited by a *size detection limit*, which is optionally specified in the database as well, after the default size and a forward slash. Such a limit is necessary to avoid that a footer for a given file is searched within the whole volume, which would be very time-consuming if the volume is large. Also, it becomes increasingly unlikely to find the right footer if not in the immediate vicinity of the header, and even if found very far apart, such a file is likely fragmented or partially overwritten etc. The standard default size (if not specified) is 1 MB. The standard maximum size (if not specified) is 64 times the default file size.

File headers are usually found at cluster boundaries because that is where file systems mostly put the start of a file. However, it is more thorough (and not slower) to search for *sector*-aligned file headers because that allows to also find files from previously existing partitions with a different cluster layout, so searching at sector boundaries is the default behavior. If performed on a physical medium or raw file with no cluster layout defined, WinHex has to search at sector boundaries anyway. There is yet another possibility, a thorough *byte*-level search. This is required when you are trying to find files that are not reliably aligned at any sector boundaries (e.g. files in backup files or tape images or embedded in other files) or when trying to find entries/records/micro-formats/memory artifacts etc., i.e. not complete ordinary files. This comes at the cost of a possibly increased number of false positives, though, misidentified file signatures occurring randomly on a media, not indicating the beginning of a file. Individual flags in the file type definition database can help on a per file type basis to decide which files to search for a cluster, sector or byte boundaries.

Start sectors of files that are already known to the volume snapshot are mostly excluded from file carving, to prevent duplicate files. This can be more or less strict. If less strict, and if the file header signature definition or the internal file size detection is strong enough to suggest that a known deleted file was overwritten with a new file, then that new file will be carved although it has the same start sector as the known file. A general exception is that the first sector of a completely uninitialized file (valid data length = 0) is never omitted from the file header signature search. The first sector of a known file whose file contents are not aligned at sector boundaries does not participate in the duplicate prevention effort either. (The start sectors of such file are no proper way of identification anyway.)

If you intentionally abort the file header signature search or if the file header signature search causes X-Ways Forensics to crash, next time when you start a file header signature search in the same evidence object, you will find an option to resume it right where it was interrupted, or where it was when the volume snapshot was last saved before the crash occurred (depends on the auto-save interval of the case).

You may limit the scope of the recovery to a currently selected block if necessary and/or to allocated or unallocated space (option available on a logical drive or volume). E.g. in order to recover files that were deleted, you select to recover from unallocated space only. Files that are not accessible any more because of file system errors may still be stored in clusters that are

considered as in use.

The effects of NTFS compression on file data can optionally be compensated for in a file header signature search (forensic license only), in many cases successfully. If the signature of an NTFS-compressed file is found, the file will be marked as compressed, and an attempt will be made to decompress the file “on the fly” when needed with a sophisticated algorithm that can even decompress files that consist of multiple compression units.

8.3 File Type Definitions

“File Type Signatures *.txt” are tab-delimited text files that serves as a file type definition database for refining volume snapshots and for the File Recovery by Type command.

WinHex comes with various preset file type signatures. You may fully customize the file type definitions and add your own ones, either in "File Type Signatures Search.txt" or in any additional such files of the same format named “File Type Signatures *.txt”, which will be loaded as well and may have the benefit that they will not be overwritten when you install the next update if they don't have the same name as one of the default files. Only if the filename contains the word “search”, the file types will be available for file header signature searches. Otherwise they are used for file type verification only of files that are already part of the volume snapshot (forensic license only). Up to 4096 entries are supported altogether (1024 for searching).

When you click the Customize button to edit the file "File Type Signatures Search.txt", by default WinHex opens the file in MS Excel. This is convenient because the file consists of columns separated by tabs. If you edit the file with a text editor, be sure to retain these tabs, as WinHex relies on their presence to properly interpret the file type definitions. MS Excel retains them automatically. After editing the file type definitions, you need to exit the dialog window and invoke the File Recovery by Type or Refine Volume Snapshot menu command again to see the changes in the file type list.

1st column: File Type

A human-readable designation of the file type, e.g. "JPEG". Everything beyond the first 19 characters is ignored.

2nd column: Extensions

One or more file type extensions typically used for this file type. E.g. "jpg;jpeg;jpe". Specify the most common extension first because that one will be used by default for naming recovered files. If that first extension is specified in upper-case characters, it will be used by the file type verification to fill the Type column for a file even if the file has one of the alternative plausible filename extensions. More than 255 characters supported.

3rd column: Header

A unique header signature by which files of this file type can be recognized. It is specified as a

regular expression (see Search Options for an explanation), so that it's possible to match variable byte values (e.g. [\xE1\xE2] mean "the byte value could be 0xE1 or 0xE2") or undefined areas (.).The maximum length of the represented signature is 48 bytes. To find out characteristic file header signatures in the first place, open several existing files of a certain type in WinHex and look for common byte values near the beginning of the file at identical offsets.

4th column: Offset

The relative offset within a file at which the signature occurs. Often simply 0. The signature must be contained in the first 512 bytes.

5th column: Footer

Optional. A signature (byte sequence) that reliably indicates the end of a file, specified as a regular expression. Regular expressions that represent variably-sized data may not work as expected. A footer signature may help to achieve a recovery with the correct file size. The recovery algorithm does not search for the footer further than the number of bytes specified as the maximum file size, starting from the header.

Even better than a footer is the potential availability of an internally implemented algorithm in X-Ways Forensics that knows the file format well and can usually find out the correct file size if a file is not fragmented, incomplete or corrupt. Such an algorithm is indicated in the Footer column with a tilde (~) and an algorithm ID number.

6th column: Default size

Optional. 1 or 2 values. If 2 values, the second one is a file type specific size detection limit and delimited from the default size by a forward slash.

7th column: Flags

Optional. Can further tailor file carving for certain file types and are yet another indicator of how sophisticated and powerful file carving is in X-Ways Forensics.

A: Means that a definition heavily depends on the associated algorithm (the one defined with the ~ character) and is too generic for identification without it.

b (lower case): The signature is searched at the byte level when given the choice. Useful especially for entries/record/micro-formats/memory artifacts (i.e. not complete ordinary files) that are not typically aligned at any sector or cluster boundaries.

B (upper case): Prevents a byte-level search for that particular signature, for performance reasons.

c (lower case): If taken into account (depends on user interface settings), ignores header signatures that are not aligned at cluster boundaries. Can be useful for some file types to avoid to many false positives.

C (upper case): Denotes file type signatures that should not be used to search for NTFS-compressed files if compensation for NTFS compression is active, because they are too weak and would yield too many false positives or would not be actually stored as compressed anyway.

d (lower case, for "direct"): The signature will be interpreted literally, not as a regular expression, character by character, with byte values according to the active code page in your Windows system. Useful for example if you are not very familiar with regular expressions or don't need them and just want to get all characters interpreted literally according to the code page that is active in your Windows system, without thinking much about whether the characters are considered special characters in regular expressions. For example, *<?xml version="1* is a valid signature for certain XML files, but it works only with the direct flag because the question mark has a special meaning, which results in a different byte value sequence for the signature internally if the entire expression is interpreted as a regular expresion, and would not yield any matches if regular expression interpretation is active.

e: Stands for "embedded". If a file type has a tilde (~) algorithm in the Footer column and is marked with this flag, it will be preselected for a search of embedded data in certain other files during volume snapshot refinements, in the "File header signature search in all files not processed above" section. The "e" flag merely helps to initialize the tick marks for this option. Ultimately the user can change the selected file types for that operation in the user interface. Also, the types marked with the "e" flag will be searched embedded in files of types for which no internal extraction algorithm exists.

E: Never carved as an embedded file within other files.

f (lower case): Indicates that the specified footer signature is used to find data that is not part of the file any more and should excluded. Ordinary footers are included in the carved file. Useful for file formats that do not have a well defined footer, where the end of the file can be detected by the occurrence of data that does not belong to the file any more. That could be the same signature as the header (if files of that type occur typically in groups, back to back) or just \x00 (for file formats such as text files that do not contain zero-value bytes, where however \x00 can be expected with a high likelihood in the RAM slack). Such footer signatures should be marked as exclusive because the data matched by it is not part of the file itself.

F (upper case): Makes X-Ways Forensics discard hits of the file header signature search if no corresponding footer can be found, provided that a footer signature is specified in the definition. Can be useful to reduce the number of or totally avoid false positives.

G: Stands for “greedy”. Greedily allocates all the sectors exclusively. The file type signature search continues its search for further file headers only after the presumed end of such files. Can be useful if an internally implemented algorithm is available that is sure that the carved file contains all valid data, so that it is not necessary to search for other files within the previously carved file's boundaries. If a file in free space is carved around allocated clusters, only the first fragment of the file is skipped when searching for further file header signatures.

g (lower case): Weaker version of the same flag. Only if an internal file size detection algorithm exists for a file type and if a file with the same start sector number exists already with the same file size as detected, the "g" flag will cause X-Ways Forensics to skip the affected sectors. This

can help to prevent overlapping zip files and thereby avoid potentially many contained duplicate files.

h: Indicates that the specified *h*eader signature is used to find data that is not part of the file itself. That means that the header signature will be excluded from the carved file. The carved file will start after the header signature. Additionally, this flag prevents file carving in free space around allocated clusters for files of this type.

H: The definition is meant only for the signature *h*ighlighting feature, not for regular file header signature searches or for file type verification. Such definitions only require three pieces of information: The keyword or regular expression, the relative offset (typically 0) and the flag "H". The description at the start of the line is optional, but recommended because the color depends on the description, and for different descriptions you will likely see different colors. You could even create a dedicated text file, for example named "File Type Signatures Search Highlighting.txt", that defines various keywords or regular expressions that you are always interested in and would like to get highlighted immediately in every case even before running appropriate searches. Also useful if you analyze or reverse-engineer file formats, where for example records do not have a fixed length (so that the record presentation option in WinHex is not applicable), but are identifiable by signatures.

L: Identifies links that merely link to other definitions. Useful for example to have an entry for OpenOffice files, which was missed by some users and whose absence could lead to the misconception that it is not possible to carve OpenOffice files. If the entry for OpenOffice is selected for carving, this internally automatically selects zip archives for carving, which makes sense because OpenOffice files technically are zip files and can be carved as such. The disadvantage is just that other zip archives that are not OpenOffice files are also carved. However, those files will be distinguishable thanks on the internal file type detection, for example based on the automatically assigned filename extension.

S: Marks signatures that are good enough for the file header signature search (probably in conjunction with a carving algorithm), but not for file type verification because of occasional misidentifications. This flag should be very rarely needed.

t: Prevents X-Ways Forensics from presenting the type of carved files immediately as confirmed. Useful for example for file format families such as XML, to determine the exact subtype later during file type verification.

u (lower case): Stands for "unused". Allows to carve files only in clusters that are free according to the file system.

U (upper case): Allows to carve files only in clusters that are free according to the file system and not known to be used by any previously existing file as per the volume snapshot either. This requires "net free space" computation. If that volume snapshot option is disabled, an upper-case U will work like a lower-case u.

W (upper case): Identifies header signatures that are too weak to newly detect the type of a file and are merely used to confirm the type suggested by the name extension of the file.

x: Identifies file types for which it is relatively normal that the actual filename extension is not the standard extension for that file type, so that files of these types will not be highlighted as "mismatch detected" after file type verification, but just presented as "newly identified", as to not draw more attention to these files than they deserve.

y: Identifies file types that are known to use encryption internally, which allows to mark carved files of these types in the Attr. column immediately with "e!".

8.4 Manual Data Recovery

It is possible to restore lost or logically deleted files (or more general: data) that are merely marked as deleted in the file system, but have not been *physically* erased (or overwritten).

Open the logical drive where the deleted file resided on using the disk editor. Principally you can recreate such a file by selecting the disk sectors, that were allocated to the file, as the current block and saving them using the menu command Edit | Copy Block | Into New File. But it may prove difficult to *find* the sectors where the file is still stored. There are two general ways to accomplish this:

1. In case you know a snippet of the file you are looking for (e.g. the characteristic signature in the header of a JPEG file or the words "Dear Mr. Smith" in a MS Word document), search it on the disk using one of the search commands (e.g. "Find Text" or "Find Hex Values"). This is a very simple and reliable way.

2. In case you only know the filename, you will need some knowledge about the filesystem on the disk (FAT16, FAT32, NTFS, ...) to find traces of former directory entries or other data structures that define the file and thereby determine the number of the first cluster that was allocated to the file.

You may encounter the problem that the file to recover is fragmented, that is, not stored in subsequent contiguous clusters. In FAT file systems, the next cluster of a file can be looked up in the file allocation table at the beginning of the drive, but this information is erased when a file is deleted.

9 Options

9.1 General Options

1st column:

- Under Windows Vista and later it may be recommendable to **always run** WinHex/X-Ways Forensics **as administrator** if you need sector-level access to media. This can be remembered

by Windows in the registry hive HKEY_CURRENT_USER under \Software\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Layers, but has no effect on installations on removable media.

- The option **Allow multiple program instances** allows you to execute WinHex more than once at a time. If not checked, WinHex makes the main window of the previous instance the foreground window instead of creating a new program instance. By default, this option is half selected. At that time you may also try to recover a previous instance if caught in an infinite loop. For example, should X-Ways Forensics get into an infinite loop when processing a certain file during volume snapshot refinement, this can potentially help the already running instance break out of that loop and proceed with the next file. The second instance also shows some technical information about what the already running instance is doing at the moment, and can do so even without recovering a supposedly hanging previous instance. Terminating a previous instance is another option, but of course should be avoided, as data loss may occur.

- Instances (sessions, processes) of WinHex/X-Ways Forensics are numbered starting with 1. The instance number can not only be seen in the so-called About box (the box that pops up when clicking the version number in the upper right corner), but for additional instances optionally also in the caption of the main window so that it's easier to tell them apart in the Windows Task Bar and the Windows Task Switcher. If the checkbox for that is fully checked, different instances have their own background colors. Note that if you end earlier instances and then start new ones, the new instances will re-use slots with instance numbers starting with 1.

- At startup, WinHex can optionally **show** the **Start Center** or **restore** the **last window arrangement** (all windows with their sizes and the positions as you left them in the precedent WinHex session).

- By default, **edit windows** are not **opened** in a **maximized** state.

- Specify the number of **recent**ly opened **documents** to remember and to **list** in the Start Center (255 at max.). Up to 9 of them are also listed at the end of the File menu.

- **Do not update file time** means that WinHex will preserve the last modification time when a modified file is saved with File | Save or Save As.

- **More context menus**: If fully checked or if the Shift key is pressed while right-clicking a directory in the Case Data window, a context menu appears that allows to recursively explore the right-clicked directory (just like when no context menu is shown), allows to tag the directory recursively (just like when pressing the Space bar), to expand the directory recursively (just like when pressing the multiply key of the numeric keypad), to collapse all, export a subtree into an ASCII text file, or copy the entire path of that directory into the clipboard. If at least half checked or if the Shift key is pressed while right-clicking the hex editor display, a suitable context menu will appear there as well.

- You may have **WinHex** appear in the Windows **context menu**. The shell displays the context menu when the user clicks an object with the right mouse button. WinHex provides menu

items for files, folders and disks. If this option is not fully selected, there is no menu item for files.

- A 3-state check box can optionally prevent Windows **screensavers** from starting and potentially requiring to re-enter the current user's password, either only during operations that show a progress indicator window (if half checked) or generally while the program is running (if fully checked). This option has an effect no matter whether the main window is visible or whether the program is running in the background. Useful for example when acquiring a live system of which you don't want to lose control during imaging, or if you wish to keep an eye on the progress indicator on your own machine from another corner in your office.

- Users can define their own tooltips for four types of control items in dialog windows: Check boxes, radio buttons, drop-down boxes/combo boxes, and ordinary push buttons (except for "OK", "Cancel", and "Help"). This is done by clicking such items with the Shift key pressed and can be useful for personal notes and ideas, so that you can describe and better remember your preferred settings for different situations and their meaning. The tooltip texts will be stored in a file named **Tooltips.txt** and can be shared with other users, for example within an organization to remind your colleagues of which settings should be used according to your defined standards. Tooltip texts are stored in Unicode, may be up to 510 characters long, and may contain line breaks for formatting purposes. You can tell that a user-defined tooltip is available for a control item if it has a gray asterisk on its left. An English language Tooltips.txt file is included. If you wish to load tooltips from that file, please make sure that the "Tooltips.txt" box is checked.

- **Save program settings in .cfg file:** If half checked, the settings are saved whenever the program terminates (cleanly). If fully checked, then every time when you click OK in any dialog window (could be useful if the program does not terminate cleanly, to avoid that you lose your latest settings). If totally unchecked, the program settings will not be saved at all, except if you hold the Shift key when exiting the program, which is necessary once if you would like to save in the .cfg file the setting that from then on the settings should not be saved again.

- By default WinHex **numbers** disk **partitions** in the order of their physical **location**.

- If **Auto-detect deleted partitions is enabled**, WinHex tries to identify obvious deleted partitions automatically in gaps between existing partitions and in unpartitioned space directly following the last partition, when opening physical hard disks. Such additionally detected partitions will be listed in the Access button menu and marked as deleted. Please note that deleted partitions detected in gaps between existing partitions cause the partition numbering to be changed. E.g. an existing partition #3 might become partition #4 if a deleted partition is detected on the disk before it.

- You can control whether opening volumes should include **volume slack**, i.e. those remaining sectors in a partition that don't add to another cluster. The data in that area, aside from a potential NTFS backup boot sector, does not belong to that volume logically and was stored there before the volume was created. It is not needed to parse the file system or to mount the volume (though some tools may output an error message if it's not included). Including such

data in a volume image can be an IT security leak if only the regularly accessible part of the volume had been sanitized (wiped) before usage, not the entire partition or physical storage device.

- If **Check for surplus sectors** is disabled, WinHex will not try to access surplus sectors when a physical hard disk is opened. When additional sectors are detected, WinHex will remember them the next time you open the disk. You may enforce a new check by holding the Shift key while opening the disk. Checking for surplus sectors may cause very long delays, strange behavior or even damage to the Windows installation on *some very few* systems.

- The **alternative access method 1** for physical hard disks may allow to access hard disks formatted with an unconventional sector size or other media that cannot be accessed otherwise. Note that it may be slower than the regular access method. If considerably slower, WinHex will notify you of this and recommend to revert to the standard access method. **Access method 2** affects physical hard disks only as well. Both alternative methods allow you to specify a timeout in milliseconds after which read attempts will be aborted. This can be useful on disks with bad sectors, where an attempted read access to a single sector could otherwise cause a delay of many seconds or minutes.

- The **Sector reading cache** accelerates sequential disk access by the disk editor. This option is recommended particularly when scrolling through CD-ROM and floppy disk sectors, since the number of necessary physical accesses is significantly reduced.

- Another option is to **always request user input for raw images** to confirm the kind of the image (volume or disk), the sector size to assume and the path for potentially existing additional image file segments. Exactly what happens if you hold the Shift key while invoking image interpretation or while adding the image to a case. Usually not necessary if the image was created by X-Ways Forensics itself, but still some removable media (USB sticks and memory cards) may have been used and formatted as both volume and partitioned medium at different times. In such a situation, interpretation as a volume and as a partitioned medium may reveal different file systems that overlap each other.

- The **surrogate pattern for unreadable sectors** is described in a dedicated chapter.

2nd column:

- Specify the **folder** in which to create **temporary files**. By default that is the directory indicated by the TEMP variable in your Windows system. Instead of an absolute path you may also specify a dot (.) as a placeholder for the directory from where WinHex/X-Ways Forensics is executed. Or .. for the parent directory of that directory. Or partial path relative to either the . or .. directory (e.g. .\temp or ..\temp). This concept applies also to the next folders.

- Specify the **folder** in which to create and expect **images and backup files** (.whx).

- Specify the **folder** in which **cases and projects** are created and expected.

- Specify the **folder** in which **templates and scripts** are stored.

- Specify the **folder** in which to maintain the **internal hash databases** and the PhotoDNA hash database. The hash database of block hash values, if used at all, is stored in a directory at the same level as the first internal hash database, with the same base name plus " [block hash values]" appended.

In all of these standard paths you may use system and user environment variables, where the variable name has to be enclosed in percentage signs, e.g. %TEMP%.

- **X-Ways Investigator [CTR]/X-Ways Imager GUI:** Available when operated with a forensic license. Allows to activate the considerably reduced user interface of X-Ways Investigator [CTR], which is meant for investigators
  - who are specialized in a certain area e.g. of white-collar crime
  - who do not need profound knowledge of computer forensics
  - who do not need technical insights that WinHex and XWF are well-known to offer
  - who receive e.g. convenient-to-handle X-Ways evidence file containers from well-versed computer forensics examiners with only selected files from various sources (e.g. "all documents that contain the keywords x and y"), with obviously irrelevant stuff already filtered out
  - who need to review hundreds of electronic documents, identify relevant ones, add comments to them, identify logical structures and connections between them with the help of their comments, and print documents, all within the same environment with a few mouse clicks, which saves the time to extract and load each document in its associated application
  - who may or may not need to work in an environment severely restricted by the system administrator anyway

  The X-Ways Investigator interface lacks many advanced technical options, to allow for easier access to non-technical personnel. X-Ways Investigator licenses that only allow to use this GUI are available at 50% the regular rate on request. An optional file "investigator.ini" controls additional simplifications and administrative security precautions, e.g. to allow users to open evidence file containers only, and only such containers that have been classified as secure.

- Windows settings for window text and background colors are adopted in WinHex/X-Ways Forensics since v20.0. Those settings can be reached with a few mouse clicks in the Control Panel in Windows XP, in Windows 7 they can still be found via Personalization | Window Color | Advanced appearance settings, and which in Windows 10 they can still be edited as raw RGB value with the Registry Editor in this key: HKEY_CURRENT_USER | Control Panel | Colors (followed by logging in and out). Black backgrounds for almost all parts of the user interface (main window, data window, Case Data window, ...) in particular are supported, which can be helpful when working in an environment with little ambient light, which generally benefits users who think they can work longer with a less bright screen, and which in general should reduce the disruption of melatonin production and the circadian rhythm among people who face screens emitting unnatural light. The viewer component respects those settings for most document types (but it does not or cannot respect them for PDF files for example). For the most complete dark screen experience you would change your entire Windows system to a dark theme. The easiest way to achieve that not only for "apps", but also real desktop applications, is to activate the black high contrast theme. In Windows 10 you

would go to PC Settings | Personalization | Settings for high contrast | Activate high contrast | Contrast black. There is also an *internal* **dark mode**, which is readily available even without any of the above procedures or settings, which you can activate when needed for night time or generally, for health reasons or to attract less attention during secretive work in a dark adversary environment. It is not 100% complete, as for example it does not affect user interface elements such as window captions, pop-up menus, scrollbars, standard file selection windows or date selection boxes. For those dark mode support from Windows is needed (see above).

Various meaningful colors in the graphical user interface have to be adjusted in X-Ways Forensics' own dark mode or when a black background color in Windows settings is detected and adopted, for example the color of file types depending on the type status. In the calendar, the grayscale coding of days with lots of activities is inversed if the background color is black. Color preferences for block selections, tag marks, "already viewed", modified bytes, and positions/search hits highlighting are remembered separately for normal mode and dark mode.

- A separate option useful in conjunction with dark mode is the ability to render pictures with the internal graphics viewing library as well as all thumbnails in the gallery darker. If that check box, which can be found next to the check box for dark mode, is half checked, that means the pixels will be darkened a little less.

- If you select **Show file icons**, the icons stored in a file are shown in the info pane. If a file contains no icons, the icon of the file *type* is shown if this option is “fully” selected. Only for files opened with the File | Open menu command.

- Last not least, you may select one of several different dialog window and button styles.

- In the "Sleep(0) Frequency" child dialog window you may specify how cooperative X-Ways Forensics behaves during long operations (e.g. hashing, searching) when competing with other processes for CPU time, by pressing Shift+Ctrl+F5. 0 is the default setting (not specially cooperative). You could try values like 10, 25, 50, or 100 (maximum willingness to share CPU time) e.g. if X-Ways Forensics is executed simultaneously by different users on the same server, for a fairer distribution of CPU time.

- With a forensic license, you may monitor lengthy operations from other computers in the same network, i.e. see whether they are still ongoing or completed. You can enable **progress notifications** via text files (that can be created in a directory on a network drive) and via e-mail, in user-defined intervals. Multiple recipient e-mail addresses can be specified as well if delimited by commas. The correct SMTP port is often 25, sometimes 587. The correct settings are provided by your administrator or Internet provider.

3rd column:

- The **ENTER** key can be used to enter up to four two-digit hex values. A useful example is **0x0D 0x0A**, which is interpreted as an end-of-line marker in the Windows world (Unix: 0x0D). The Start Center could then still be opened using **SHIFT**+**ENTER**.

- Decide whether you want to use the **TAB** key to switch from text to hexadecimal mode and vice versa or to enter the TAB character (0x09). In any case, **TAB**+**SHIFT** can be pressed to switch the current mode.

- Non-printable **characters** with a character set value smaller than **0x20** can be represented by a user-defined other character such as a space or a period. That substitute character can also be used for high Unicode values. It's easier on the eye if characters in languages other than your own are not actually displayed, and you can probably afford to not see them if you are not looking for foreign language text (e.g. Chinese, Japanese, Korean) anyway. To see only pure 7-bit ASCII characters (sufficient for English), in ANSI ASCII and all UTF-16 variants, you can apply the substitute character to above 0x0080. To see letters at least from other Western European languages like Spanish, French, German you can apply it to > 0x00FF. To see Eastern European languages, apply it only to > 0x04FF.

- The **bytes** in the **display** can be represented **as** characters in the **text** column **one by one**, or WinHex can try to combine them, which if the active code page in Windows is a double-byte character set *may* be desirable to get the characters right (if 2 bytes = 1 character), or undesirable because of the variable row length. This has an effect only if View | Character Set | * ASCII is selected, as only then the code page active in Windows can make a difference for the display.

- When using the **memory editor**, it may be useful to have WinHex display **logical** memory **addresses** for processes instead of zero-based, linear, contiguously counted offsets. This is always done in hexadecimal notation. The dialog window of the Goto Offset command will also prompt for logical addresses.

- **Page** and sector **separators** may be **displayed**. If this option is enabled partially, only sector separators are displayed.

- Specify the number of **bytes per line** in an edit window. Common values are 16 or 32 (depending on the screen resolution).

- Choose how many **bytes** shall be displayed in a **group**. Powers of 2 serve best for most purposes.

- There is an option to define the size of the extra gap between rows in the hex editor display in pixels, which together with the official height of the selected font defined the distance between the rows. The default value has always been 3 before v17.2, but now it can be decreased, to display more rows at the same time and see more data. For example with the Courier font the display still looks fine with an extra gap of 1, but you see 15% more data (based on font size 10). Even negative values are possible. With -1 you may see 35% more data than before.

- **Search hit highlighting in File mode**: Option to get all search hits in a file highlighted in File mode at the same time, either only when a search hit list is displayed (if half checked) or permanently once search hits have been loaded for an evidence object, i.e. even when working with the normal directory browser (if fully checked). Search hits are loaded after an evidence

object has been opened as soon as search hits are listed. This feature also applies to user search hits. Requires forensic license.

- **Automatic coloring:** 1) Highlights the various elements in FILE records of the NTFS file system, when the cursor is located within such a record, to facilitate navigation and understanding. Requires a specialist or forensic license. If half checked, highlighting is attemped only on NTFS-formatted volumes, not in other file systems and not on physical, partitioned disks.

  2) Highlights FILETIME values in Disk/Partition/Volume and File mode. Useful when manually inspecting files of various Microsoft formats which may contain more timestamps than can be automatically extracted (try e.g. with index.dat, registry hives, .lnk shortcut files etc.). If the lower half of a data window has the focus and FILETIME values are highlighted, you may also hover the mouse cursor over such a value to get a human readable interpretation of the timestamp. Alternatively, of course, you could get it from the data interpreter if you click the first byte of the value. If half checked, only FILETIME values that are aligned at 4-byte offsets are highlighted.

  3) X-Ways Forensics and WinHex Lab Edition: Highlights file header signatures, right in the hex display (X-Ways Forensics: Disk/Partition/Volume and File mode). The identification is done by matching the signature definitions in "File Header Signatures Search *.txt" to every single offset in the currenly visible page. The enhancing effect of the "~" algorithms, which often can identify false positives or further distinguish between different subtypes during file header signature searches, is not applied, though. This kind of highlighting will help you spot start positions of well known data/file types, even if embedded within one another, immediately, for example thumbnails in JPEG files, individual records in zip archives, TIFF signatures in Exif metadata, certificates in Windows Registry hives, etc. etc. For more information, refer to the documentation of the H flag of the signature definition file. If half checked, only signatures aligned at 512-byte boundaries are highlighted.

- **Highlight free space/slack space:** Displays offsets and data in softer colors (light blue and gray, respectively). Helps to easily identify these special drive areas. Requires a specialist license at least.

- Select a **color** used as the **background** of the current **block**.

- Select a **color** used as the **background** of every other fixed-length **record**, if record presentation is enabled (see Position menu).

- Select the default **color** for newly created **annotations**/positions.

- You may want WinHex to **highlight modified bytes**, i.e. display altered parts of a file, disk, or memory in a different color, so you can distinguish between original data and changes you have made so far. You may select the hilite color.

- Select the **color** for slack space and uninitialized space.

- You may choose a **font** for the hex editor display, and decide whether the standard Windows GUI font should be used for the other parts of the WinHex/X-Ways Forensics GUI (for example in the directory browser and in the Case Data window).

- You may adjust the size of the standard GUI font. A positive number of pixels increases the size, a negative number decreases it. Restarting the application is recommended after making any adjustments. Generally it is much better to adjust the DPI scaling settings in Windows instead because that has a more consistent effect on all elements of the GUI, including clickable controls etc., not just on the font size in certain areas. However, there are situations in which it is more practical to control the font sizes in X-Ways Forensics specifically, for example if your eyesight is above or below average and you frequently use a portable installation of X-Ways Forensics on computers other than your own.

Factory settings of *all* options (application wide) can be restored using the Initialize command of the Help menu.

9.2 Directory Browser

- **Grouping files and directories** in the directory browser is optional. X-Ways Forensics remembers the sort criteria and this option separately 1) for the normal directory browser of a volume, 2) for the normal directory browser of a partitioned disk, 3) for search hit lists and 4) for event lists. By default this box is half checked, which means grouping occurs only when not exploring recursively, i.e. only when directories are needed for navigation and thus helpful/expected at the top of the list.

- **Grouping existing and deleted items** in the directory browser is optional. There are two possibilities how to use this feature. Either previously existing files that potentially recoverable (question mark icon) and known unrecoverable (red X icon) are internally grouped as well (so that in total there will be three groups) or not (only 2 groups). A small symbol with either one or two horizontal dividers indicates whether the list is split up into two or three groups, also in the header of the column that is the primary sort criterion, as a small reminder that when scrolling in the directory browser and watching out for a certain file for example based on its name, you need to check in every group, because the sorting takes place within each group and does not span the groups.

- **Double-clicking** a directory will **explore** it. Double-clicking an ordinary file will **view** it. This option controls whether files with child objects will be typically viewed or explored on a double-click. If the checkbox is half-checked, you will be prompted.

- Files can optionally be **opened and searched** including their **slack**. The middle state of this checkbox makes a difference only for logical searches (cf. that topic).

- A "**..**" item can be optionally listed at the top of the directory browser when navigating within a volume from one directory to another. If displayed, it is frozen at the top and does not scroll along with all the other items. It shows all the information on the directory that it represents (the one that you would navigate to if you double-click it), just like with all the other items in

the directory browser. A "**.**" item is also displayed optionally, representing the currently explored directory. Useful if for example you wish to see certain metadata (e.g. timestamps) of the parent object at the same time as metadata of its child objects. And if the . or .. item is a file and you select it, then you can see that particular file in File, Preview or Details mode. And it is represented in Gallery mode.

- Listing the root directory of a volume in the directory browser, in the root directory itself, actually, is kind of illogical, but can be very helpful to see that directory's timestamp (if any, depends on the file system) or to quickly navigate to its clusters (if any, also depends on the file system) or as another place where to quickly tag or untag all items in a volume.

- Listing the internal files of the file system is optional in the normal directory browser. This affects for example the various $* files in NTFS. Specifically in X-Ways Investigator those files are no longer listed as they are irrelevant to non-technical examiners (the target group of X-Ways Investigator) and might confuse them because they are not familiar with them from using ordinary high-level computer software.

- If multiple filters are active, they are usually ANDed, meaning each file has to pass the first active filter AND also all other active filters to be listed in the directory browser. However, you can also filter files with a logical OR, meaning any file that passes the first active filter OR any other active filter will be listed. If active filters are combined with a logical OR, that is shown in the directory browser caption line next to the active filter count. A click on the filter count or the word OR toggles between AND and OR combination. If multiple filters are combined with OR, the Description filter can still be optionally ANDed and is ANDed by default, as you can tell from an additional checkbox labeled AND in the Description filter dialog window, visible in such a situation, and that filter is then counted and treated separately. Please note that complex nested filter settings with OR and AND can be realized using multiple .settings files.

- That filters are applied to directories, too, is optional. If active, that concerns only suitable filters. Filters that do not make sense to apply to directories (Type, Type Status, Hash, Hash Set, Author, ...) are not applied.

- Listing sub**dir**ectorie**s** when **exploring recursively** is optional. They are not needed for navigation if already all files from all subdirectories are listed and may distract you when you are merely interested in viewing files. By default this option is half-checked. That means that directory are not needed for navigation, and they are listed only if they match any active filters that is actually applicable to directories (Name, timestamp filters, Owner, Int. ID, Attributes, ...). If for example both the Name filter and the Type filter are active at the same time, directories will not be listed, because even if they satisfy the Name filter, they cannot possibly satisfy the Type filter (directories do not have a file type). But if the Name filter is on and the filter for timestamps, then directories are listed if they match both filter conditions.

- The **selection statistics** are displayed below the directory browser (with a forensic license only). If computed in a **recursive** way, they reveal how many files and how much data are contained in a selected directory and all its subdirectories (or in a file with child objects) when selected it in the directory browser, taking any active filters into account. If this option is not

enabled, the statistics only tell you about what is directly selected, not about the child objects that may indirectly be covered by that. The middle state lists the direct selection and the total of direct and indirect selection separately. If you select multiple video files whose play durations are known in the Metadata column, the total play duration of all these videos combined is computed and shown. This enables you and others (e.g. lawyers) to better understand the amount of video data, for example to assess how complete the coverage of surveillance videos is or to judge the amount of illegal videos found, in a more meaningful way than measuring it in megabytes, gigabytes or terabytes, especially for a computer layman.

- **Tagging** or **excluding** items in the directory browser can occur **recursively** or non-recursively. Non-recursively means that tagging/untagging/excluding/including a file or directory in the directory browser has no effect on parent or child objects or parent directories or subdirectories. Useful for example if all child objects of a file should be processed in volume snapshot refinement or searched, but not the parent object. If it works recursively, then it is not possible to have an untagged parent object whose child objects are all tagged. If the recursive tagging option is in its middle state, that means that child objects still inherit the tagged state from their parent at the moment when they are newly added to the volume snapshot, e.g. when you extract e-mail and attachment from a tagged e-mail archive. Whether tagging and excluding work recursively or not can also be controlled by holding the Shift key. Tagging or untagging recursively can be *very* slow in large volume snapshots.

- **Advanced sorting**: Takes 4 to 6 times more time than the highly optimized standard Unicode sorting (noticeable when sorting millions of files), but has several useful settings and characteristics:
  - Language-specific character equivalence rules (treat ß like ss, treat é similar to e, ü similar to u etc.)
  - Linguistically improved case insensitivity
  - Special treatment of hyphens and apostrophes (they are treated differently from other non-alphanumeric characters to ensure that words such as "coop" and "co-op" stay together in a sorted list).
  - Treat decimal digits as numbers, e.g. sort "2" before "10" (not useful for hexadecimal notation, available under Windows 7 and later only)
  - Treat half-width and full-width characters the same (full-width characters are sometimes used by East Asians when writing English language letters)
  - Ignore kana type (treat corresponding Japanese hiragana and katakana characters the same)

  Advanced sorting depends on the regional settings of the currently logged on user. For example, if regional settings of a Nordic country are active, Å comes after Z, as defined in the alphabets of that region, otherwise near A, as perhaps expected by non-locals. Advanced sorting rules are also applied when sorting the search hits by the Search Hit column.

  There is an option to sort search hits by their data and context instead of just by the search terms to which they belong. Helpful for keyword searches (not technical, e.g. hex value, searches). Indeed slower since the data and context of all search hits to sort have to be read and converted to a comparable code page. Sorting by the data in search hits helps for RegEx searches. It makes a difference only for regular expressions that match variable data because for constant search terms the search terms and the data in their corresponding search hits are identical. For example, after searching for e-mail addresses with the expression [a-zA-Z0-9_\-

\+\.]{1,20}@[a-zA-Z0-9\-\.]{2,20}\.[a-zA-Z]{2,7}, sorting by the data allows you to quickly identify and visually skip groups of identical e-mail addresses or see similar e-mail addresses (starting with the same characters) next to each other. Continuing sorting by the text that follows the actual search hit if the search hit data is the same will show identical or similar text passages next to each other and allow you to more quickly review the search hit list. You can specify how many characters of data and context to take into account for sorting. The more characters, the more memory is needed for sorting, which can make a difference when listing a huge number of search hits.

- Optionally, after **start-up**, the directory browser can be not **sort**ed at all, for performance reasons. That means the program will forget the last sort criteria in use last time. If selected, there will now also be no sorting when turning off all filters with a single mouse click, to avoid longer delays when suddenly all files are listed again recursively.

- A 3-state checkbox controls whether clicking/selecting a file or directory in the directory browser will navigate to the data associated with that object in Disk/Partition/Volume mode or to the object's defining data structure in the file system. Please note that a quick jump to the latter can also be achieved by clicking the FS offset cell of that object even if a click elsewhere navigates to the former. If the box is unchecked, no navigation in the lower half of the data window will take place at all, which could be beneficial if you are operating directly on a physically damaged disk, where accessing certain sectors or regions may cause hanging in the application or a crash in the operating system.

- **Directory browser settings** (in particular column width, filter settings and sort orders) can be optionally **stored in cases** and reactivated when loading cases (if stored by a compatible version).

- **Dynamic e-mail and timestamp columns** lets X-Ways Forensics decide whether to include the columns Sender and Recipient in the directory browser. They will be included if at least one extracted e-mail message is in the visible portion of the directory browser, otherwise not. Helpful because that leaves more room for other columns when the columns exclusively filled for extracted e-mail messages are not needed. The columns with alternative timestamps, including the content created column, can also be shown dynamically, i.e. only when items that have such timestamps in the volume snapshot are displayed in the visible portion of the directory browser.

- An option exists to **show** the **file type ranks** in the Type status column, which also causes sorting by that column to **sort** by those ranks. Ranks are defined in the File Type Categories.txt file.

- The threshold above which backdating is brought to your attention in the display of a timestamp column in the directory with the clock+arrow icon and a brief representation of the time discrepancy and which is used for the backdating filter can be set in the directory browser options. The detection of backdating activities in NTFS timestamps can be limited to instances where the subsecond part of the timestamp (the digits after the decimal point) is zeroed out, with the expectation that backdating was performed manually by some timestomping tool that did not bother to create randomized subsecond digits and rather set those parts of the

timestamps to zero. (Malware that tries to cover its tracks and backdates files automatically and algorithmically may be better than that.)

- **File counts** can optionally be displayed in the directory browser at the end of the names of directories and files with child objects. If fully checked, that will happen also in the directory tree in the Case Data window.

- By default, the Path column displays a **partial path** from the current exploration base when exploring recursively. That is the same path that you would get with the Recover/Copy command when reproducing a partial path only. Useful for example if you wish to share directory listings including subdirectories with someone (Export List command), distinguishing files in different subdirectories, without revealing the complete path of the files (e.g. on your own storage drive). If fully checked, a partial path starts with the directory name. If half checked, it starts with ...\ to point out the omission.

- A button with an arrow allows to right-align the path columns in case you are more interested in the end of the path and would like to keep the column width compact. The arrow points to where the paths will be aligned.

- **Special file icons** for pictures, archive files, documents, videos, text file and audio files are available, very useful when your main focus is on any such kind of files. Depending on whether the checkbox is fully checked or half checked, symbols like question marks, arrows, scissors, hammers, etc. that further reveal the status of the file get superimposed additionally or not. If not, that is easier on the eye. You can still tell the exact deletion status from the Description column, and the rough deletion/existence status is still obvious from the contrast of the icon.

- **Icons** that **show** the **categorization** of a file (as notable, irrelevant or unknown) can optionally be displayed right in the Name cell of that file.

- **Conditional cell** background **coloring** helps to draw your attention to items of interest without having to filter out all non-matching items. Matching items are found through a substring search in the cell contents of a selected column. Substring expressions may be up to 15 characters long. You may use an asterisk to match anything except blank cells. If a match is detected in a cell, either only the background of that particular cell can be colored (called "cell-targeted coloring") or the entire line. To color an entire column, regardless of the cell contents, activate cell-targeted coloring for that column and specify an empty condition string, i.e. no condition at all. If a cell matches multiple conditions, all of these conditions have an effect and contribute to the final color mix (the last one most), so that hopefully none of the targeted properties go unnoticed. For line-targeted coloring, only the first 255 characters in the respective cell are guaranteed to be searched.

  Conditions cannot be defined for search hit specific columns, but for event specific columns. That can prove useful when trying to identify patterns in events. For example, you could color all events of type "Program started" in red and log-in events in yellow and see more easily how far apart from each other they are. Conditional cell background coloring is case-specific if "Store directory browser settings in cases" is selected. The color settings are also stored in a

file named "Conditional Coloring.cfg", and they are stored in and loaded from .settings files along with other directory browser settings. Up to 255 conditions may be defined.

Columns

Various columns are available in the directory browser. They are all optional. They are displayed if they have a non-zero column width in pixels, or hidden if their width is zero. You can toggle column visibility purely with the mouse if you like, by clicking the column label in the dialog window.

It is possible to redefine the *order* of the columns in the directory browser. This will also change the order of the fields in the case report (i.e. in report tables), on print cover pages, in exported file listings, and the Export/Copy log. You can select a column for relocation by clicking its radio button. Then use the vertical scrollbar that appears at the top. You can reset the column order to the default one by *right*-clicking that scrollbar.

9.3 Notation/Output

There are notation/output settings for the main GUI (graphical user interface), accessible from the main menu, and separate settings specifically for output like the case report, the Export List command, and Recover/Copy. Among other settings:

Choose your preferred **date, time, and number** notation settings. This is important especially to be independent of the Windows regional settings of live system that you want to preview if you are using X-Ways Forensics on a computer that is not your own one. You may also choose to display years in dates with 2 digits only.

There is an option to output dates in the directory browser and in some other parts of the user interface in a nicer, longer and more locale-specific notation, which can include the weekday and the name of the month based in your language or in English. Also, that format is Unicode-capable, which allows for example for original Chinese notation of dates. Please see http://msdn.microsoft.com/en-us/library/dd317787%28v=vs.85%29.aspx for a complete explanation of what kind of notation is possible. Examples of how to represent the month (in English): MMMM = April, MMM = Apr, MM = 04, M = 4. Example of a complete format: d/MMM/yyyy (ddd) = 2/Apr/2014 (Wed).

There is an option to display timestamps with a precision of milli**seconds** or higher. You can enter the desired **number** of **decimals**. The effect depends on the available precision of the original timestamp format and on where the timestamps are stored. Timestamps in the volume snapshot are displayed with up to 4 decimals (fractions of seconds), where the 4th digit is rounded. Available higher precision is used for sorting purposes even if not displayed. All of this is useful for example for file systems like NTFS that provide for a very high precision in all or some timestamps. If you wish to see timestamps in the volume snapshot of an NTFS file system with more than 4 digits after the decimal point, you do not have to know low level file system data structures and point the Data Interpreter to the timestamp in the 0x10 attribute of a FILE record. You can simply *open* the file of interest or switch to File mode for the selected file and

refer to the Info Pane for the maximum of 7 such digits. This full precision is also available in the Info Pane for files in an evidence object that is a directory in an NTFS file system as well as for files opened directly with the File | Open command.

Optionally, the actually used **time zone conversion bias**, including daylight saving where appropriate, can be displayed right in the timestamp columns in the directory browser.

User can decide whether redundant timestamps from **0x30 attributes** in NTFS FILE records should be displayed/output or not. By default they are hidden, in order to not unnecessarily clutter up the screen, with the goal in mind to require the user's time and attention only for timestamps that actually contain additional information. However, if you feel you temporarily need to see all timestamps to double-check or if the recipients of exported lists that you are sharing wish to see all timestamps, you can selectively enable redundant timestamps for where they are needed. A middle setting allows to see redundant timestamps dimmed in a light gray color in the directory browser just like previously known from never updated last access timestamps. (The middle setting is not available in the Notation settings for case reports, the Export List command or the Recover/Copy command.) Filters and sorting work no matter whether redundant timestamps are currently visible or not. If a timestamp filter specifically targets a column with redundant timestamps, those timestamps will be displayed and highlighted even if they were otherwise invisible.

Offsets can be presented and prompted for in a decimal or **hexadecimal** notation.

Hexadecimal digits beyond 9 can be displayed in either **lower** or upper **case** characters (a-f, A-F).

File sizes can optionally **always** be **displa**yed **in bytes** instead of rounded. If the checkbox is half checked, that applies to items in volumes only, otherwise also items on physical, partitioned media.

File sizes can also be displayed in units of sectors. If not found on storage devices or images with sector-level access, but e.g. in evidence objects that are zip archives or directories, a standard sector size of 512 bytes is assumed. The display sector count is either rounded up (because a file occupying 1 full sector plus 2 bytes actually utilizes 2 sectors where files are stored as sector-aligned) or it is displayed with one decimal digit. The display style with one decimal digit can give you an idea how precisely or roughly carved files were sized because if a file size is an exact multiple of the sector size, it will be displayed with no decimal, whereas .0 indicates a few extra bytes that just do not amount to one tenth of a sector. This can also give you an idea which file types are naturally rounded in size, e.g. Windows registry hives and OLE compound files. On the other hand, if a JPEG or HEIC or any other usually unrounded file is shown with no decimal digit, that is a candidate for a file that was truncated, e.g. by carving or file system corruption. (Though if file sizes are equally distributed, one in 512 files would happen to be a multiple of the sector size naturally.)

Checksums are by convention usually displayed in big-endian byte order. That means for example that a 2-byte file with the byte values 0x33 0x00 has a 16-bit checksum that is shown as 0033. That is understandable because checksums are true integer number with a meaning (the

result of an addition of assumed integer numbers), and numbers are read and written by humans with the most significant digit first from left to right (even in Arabic and Hebrew). CRCs like CRC16, CRC32 and Adler32 are also often (but not always) byte-wise reversed for display purposes. However, this is already harder to justify because those hash values are more random in nature and do not have an actual meaning as numbers. Longer hashes are never byte-reversed. One notation setting allows you to see **checksums and CRCs in litte-endian / original byte order**, for compatibility with specific other tools if necessary or when comparing checksums and CRCs to how they are actually stored on disk / in files, e.g. in zip archives.

SHA-1 and **TTH192** hashes can optionally be displayed in **Base32** notation in the directory browser, as common in P2P programs.

Choose from an infinite number of different ways to **represent** the **existence** status of files and directories. Users can describe the "existent" or "non-existent" status in their own words. Particularly useful for example for the Export List command to match the expectations of a 3rd party and for the Recover/Copy command when grouping files by their existence/deletion status so that you get directories named accordingly.

Choose what kind of **labels** you would like to see in the Labels column at all, and whether the existence of labels should be hinted at in the Name column with smaller or larger triangles. Label names can optionally be truncated in the Labels column after *x* characters, so that more label names fit into the cells of the directory browser. Half-checked means that truncations are marked with an ellipsis.

Decide what kind of descriptions you would like to see in the **Description** column. Some "internal" flags can also be shown if you wish. Those flags identify the status of a file in volume snapshot refinement.

[Emb]: checked for embedded data to uncover
[Arc]: file archive checked for content
[Enc]: encryption test already performed
[Ext]: e-mail or e-mail archive checked for extractable content
[Met]: checked for internal metadata
[Xtn]: created by an X-Tension

In the **evidence object** column you can either get to see the title of the evidence object, which is user-definable (and only up to 79 characters long), or the complete internal path of the evidence object.

You can choose to not see filename extensions in the columns "**Name**" and "Parent name", which could be useful for users of X-Ways Investigator in particular who do not care what type the file is (or pretends to be).

You can tell X-Ways Forensics what you like to see in the **Int. Parent** column: The internal ID of the parent or its name or its description or a combination, where the filename can optionally be truncated before the extension. in the name can optionally be in the columns "Name" and "Parent name", which could be useful for users of X-Ways Investigator in particular.

The **1st sector** column can optionally show offsets instead of sector numbers if you need more precise locations. And it can show physical start sector numbers for files in partitions (counted from the start of the partitioned, physical storage device) instead of logical start sector numbers (starting with 0 for the first sector in the partition), if the partition was opened from within a partitioned, physical storage device (or image). The column label contains a P in a circle to remind you if this setting is active, avoid misinterpretation (P for physical). This works only for ordinary partitioning (MBR or GPT style), not Windows dynamic volumes or LVM2 volumes. Physical offsets are also possible.

9.4 Volume Snapshot Options

These options usually take effect when taking a new volume snapshot.

Left half of the dialog window

• Extended attributes in NTFS are optionally included in the volume snapshot as child objects of the directory or file to which they belong, with the name "$EA" and marked in the Attr. column with "($EA)". Either all such attributes (if the box is fully checked) or only non-resident ones (if half-checked, default). If none at all, the clusters that belong to non-resident extended attributes of existing objects will be covered by the virtual file "misc non-resident attributes" as before. Background information: Microsoft uses extended attributes on system binaries as part of the secure boot components. Attackers have been using large extended attributes to hide malware in some high profile cases. Large extended attributes are flagged automatically with labels.

• **Including** logged utility streams (**LUS**) in **NTFS** in newly taken volume **snapshots** is optional. Either *all* LUS can be included (if fully checked) or only non-$EFS LUS (if half checked) or no LUS at all. Useful for NTFS volumes written by Windows Vista, if you are not interested in $TXF_DATA LUS.

• Downloaded files in NTFS can be conveniently recognized if their alternative data stream "Zone.Identifier" is represented as a label instead of as a child object in the volume snapshot. That means you do not need to navigate to the child object to find out what the child object might be. "ZoneId=3" as the name of the label identifies files downloaded from the Internet.

• X-Ways Forensics has the ability to detect unusual or suspicious short filenames (SFNs, 8+3 character names) in NTFS. Such short filenames can optionally be output in the volume snapshot either as alternative names or as fully valid hardlinks themselves (i.e. like additional copies of the same files). They can also be labeled as "peculiar SFNs" to make you aware of them. Unexpected SFNs that don't seem to match their corresponding LFNs could be interesting if they reflect previous names of files that have been renamed, or because they may have been specially engineered to replace sensitive files with fixed names (such as DLLs or configuration files), while their LFNs are different and perfectly innocuous. If you find that too many normal files are flagged that way, you can try UNchecking the box for "more strict matching", so that some of the less severe discrepancies are ignored.

• You can choose which copy of a FAT12/FAT16/FAT32 file allocation table to work with. This can be either a user-designated copy or the one that is defined as active in the boot sector (in case of FAT32). If neither the user selects a copy nor the boot sector defines a single copy as active, the first copy will be used, labelled as "FAT 1". The copy that was selected at the time when the volume snapshot was taken will be used for the whole lifetime of that volume snapshot, even if the settings are changed. It is displayed in the Info Pane. The Technical Details Report informs which copy or copies are considered active in the file system.

• By default, allocated clusters in FAT12, FAT16, FAT32, and exFAT file systems are skipped when reading the data of deleted files. That means that data of deleted files is not necessarily assumed to be contiguous, but assumed to occupy as many free clusters from the start cluster number as are necessary to accommodate the known file size, while skipping clusters that are marked as in use by existing files. If the end of the volume is reached that way, the next free clusters are taken from the start of the volume, replicating the built-in logic of typical FAT32 file system drivers to rotate through the volume on the search for allocatable clusters. This option retroactively changes the assumption about the storage location of files that are already contained in the volume snapshot, thus changing this option will also cause hash values to change if they are re-computed.

• The extra effort that X-Ways Forensics makes to include deleted objects in FAT32 file systems correctly in the volume snapshot is available only with a specialist license or higher, and optional. If only half checked, the extra effort is made only for subdirectories, not files.

• A directory tree depth at which an error in the file system will be presumed and at which recursion will be aborted when taking a volume snapshot of FAT* or Ext* file systems can be defined and helps to avoid stack overflow errors, which would otherwise occur in some very rare cases. If this situation occurs, a message will be output: "Probably circular link detected. Recursion depth ...".

• If you get read errors on a CD/DVD (e.g. because of scratches on the surface) when the volume snapshot is taken, you know that not all sectors with the data structures of the file system are readable. **List**ing the **ISO9660** file system's directory tree on CDs *in addition* to a possibly also existing **Joliet** file system can be useful because that means a second chance to get all directories and files listed, if the corresponding data structures of the same directories are located in *readable* sectors in the ISO9660 area.

• In Ext file systems, more in-depth parsing of deleted directory entries during the initial creation of the volume snapshot is an option, covering even misaligned entries in relation to the existing directory entries. This might find additional previously existing files in Ext, at a likely manageable risk of finding some garbage entries as well.

• Complete output of EA (extended attributes) in HFS+/APFS is not checked by default. All extended attributes deemed relevant by X-Ways Forensics are still processed and output either in the Metadata column if they are textual in nature or as file contents of resident or compressed files or as links to related directories, or as child objects that are marked in the Attr. column with (EA). If half selected, "firstlink" attributes and "quarantine" attributes are output in the Metadata column additionally. If the new option is fully checked, even empty binary PLists and ordinary "Security" attributes are output as child objects.

• Output of simple extended attributes in Apple file systems as special lines in the Metadata column instead of child objects is optional. If included in the Metadata column, the Metadata field will also be shown in Details mode.

• For better results when matching hash values against special hash sets, only the invariable **header of loaded modules** can be listed in main memory analysis.

• Directory listings obtained from the operating system ("OS dir list"), which you get for example when adding a directory or a single file to a case as an evidence object, can be made to not show any timestamps from the file system or only the modification timestamp. That is useful if the timestamps of the files do not have the usual significance, e.g. if they reflect when you collected the files and not what timestamps they had originally at their original location.

• When taking a volume snapshot of directories (or entire drive letters without sector-level access), where it's not X-Ways Forensics itself that parses the file system, but Windows (internally referred to as file system "OS dir list" = operating system supported directory listing), alternate data streams can also be included. This can be turned off if you are not interested in ADS and/or wish to save time.

• There is an option for incremental snapshot completion when dealing with OS directory listings as evidence objects (when you add a directory to your case). If selected, the volume snapshot initially just contains the contents of the top-level directory, and it is further completed only on demand, step-by-step when you manually explore subdirectories. This is exactly how the Windows Explorer/File Explorer in Windows works, and useful when dealing with slow and huge network drives that would take a long time up front to scan completely. But it's very different from the usual approach in X-Ways Forensics, and will obviously prevent you from getting a complete listing of all files when exploring recursively, simply because there is no guarantee that all files have been included in the volume snapshot yet until you have explored all subdirectories. If at any time you decide that you wish to include the contents of a certain directory in the volume snapshot recursively, you can use the "Expand all" command in the context menu of the Case Data window (right-clicking that directory) or unselect the option to complete the volume snapshot on demand and then explore that directory. Please remember that the most convenient way to expand an entire subtree is by clicking its root and pressing the multiplication key on the numeric keypad (standard feature in Windows). If the incremental completion option is active, directories that have not been explored yet are marked with an asterisk (*) in the Attr. column.

• Computing the total amount of data in files found in OS directory listings is optional. Any discrepancy between the original amount of data and the new amount calculated when re-opening the evidence objects is brought to the user's attention and triggers an offer to take a new volume snapshot.

• In volume snapshots based on directory listings of the active operating system ("OS dir list"), write-locked files that are open in other processes and cannot be changed are optionally shown with an upper-case "L" in the Attr. column (for "locked"). Files that are merely kept option may be shown with a lower-case "o" if the box that represents this option is fully checked (for "open"). This could be useful when previewing or acquiring a live system, to find out which files

are/were open in running processes or background services, or which executable files appear(ed) to be running/loaded. Please note that checking this for many files will take a long time. It may be practical only for specific directories of interest. This option has no effect on mapped network drives. It is possible to use the Attr. filter to quickly target open or write-locked files, and these files are higher in the sort order for the Attr. column.

• Evidence file containers of v18.8 and later specifically remember the volume snapshot refinement (RVS) status of the files that they contain, e.g. whether still images have been captured already from a video or whether embedded data already has been uncovered from a file. If you choose to accept and trust this status, these files will not be processed again if you decide to refine the volume snapshot of the container. You may occasionally not want to accept the RVS status of files in containers, to avoid missing something, if you suspect that the original examiner did not apply as thorough settings as you would or that they may have used an older, less capable version of X-Ways Forensics to process the files. Adopting the RVS status is also a must to get videos within a container represented in the gallery with rotating captured still images.

Right half of the dialog window

• Whether **previously existing files** should be included in volume snapshots at all is something that you can decide. If you are not interested in such files because you are not conducting any data recovery or thorough conventional computer forensics etc., you can save time by omitting such files.

• **Inherit deleted state**: Causes deleted partitions to pass on their deleted state to everything that they contain (files and directories), and deleted e-mail archives to pass on their deleted state to all the e-mails, directories and attachments that they contain. This may seem logical, but results in a loss of information, as depending on the reference *everything* may be listed as deleted, even files/e-mails that from the point of the file system/the e-mail archive still existed when the partition/file was deleted. By default, this option is not selected, so that X-Ways Forensics distinguishes between existing and deleted files and e-mails etc. even in deleted partitions/deleted e-mail archives, so that more information is retained.

• Additional hard links for the same file in NTFS can optionally be omitted already when taking a volume snapshot, which means they will not be included at all and not shown in the directory browser as additional files. That could be helpful for example when trying to make sense of storage space utilization, where counting the same files 10 or 100 times does not make sense. The "Link count" column still shows the true number of hard links (which, however, does not count pure 8.3 character filenames and which, by the way, may differ significantly from the not very well maintained hard-link count in the FILE record).

• Newly discovered names (e.g. e-mail subject lines of original .eml files or names of files in iPhone backups) can become the main filenames in a volume snapshot (and thus also potentially part of paths if they have child objects), so that the original names as per the file system become secondary names, or they can become the secondary names themselves, displayed in a lighter color after the main names in square brackets as additional information. The default setting is half selected, which means only for original .eml files newly identified names (i.e. the subject lines) become the main name in the Name column, and the potentially unhelpful generic filenames according to the file system are preserved as the secondary names.

• **Net free space computation**: Allows you to work with an adjusted virtual free space file that is net of clusters that were identified as belonging to previously existing files, to minimize the amount of space in file systems that is read *twice* for logical searches and indexing. After changing this option or after discovery of more previously existing files, the virtual free space file is updated when it is opened next time, for example selected in File mode or when it is that file's turn during a logical search. Relative offsets of search hits in this virtual file may become wrong when it changes (for example when some more clusters are allocated to more identified previously existing files, so that the net free space file becomes smaller), so they cannot be used to navigate to the search hits in File mode. Only physical offsets of search hits, usable in Partition/Volume mode, are guaranteed to remain valid. The virtual free space will be frozen and not change any more once it has been indexed, or once it gets child objects, i.e. usually files that have been carved within it manually in File mode, because those depend on unchanged relative offsets within the virtual free space file.

• Optionally, **files** on the logical drive letters A: through Z: can be **opened** from within the directory browser with the help of the **operating system** instead of with the built-in logic at the sector level. Please note that this is forensically sound only for write-protected media. On writeable media, Microsoft Windows may update (i.e. alter, falsify) the last access timestamp of files you open. The benefit, however, is that access to such files will be noticeably faster in many situations, especially on slow media such as CDs and DVDs, e.g. when you compute hashes or skin color percentages for files in a volume snapshot, because Microsoft Windows employs read-ahead mechanisms and entertains a file caching system. Another benefit is that files opened with the help of the operating system are editable in WinHex. Limitation: Files on multi-sessions CDs and DVDs cannot be read that way.

• Known **uninitialized** portions at the end of a file in certain file systems that remember such conditions (valid data length < logical file size) are presented by Windows towards ordinary applications that open files through the operating system (and don't read file contents directly from the sectors of a volume) as binary **zeroes**. That means the data that is actually stored in the allocated clusters and that is older than and usually unrelated to the respective file, is ignored. This behavior of Windows can be optionally reproduced. "Read uninitialized areas as zeroes" is a 3-state check box. If fully checked, it has an effect on all read operations except logical searches, indexing, and search hit context preview. If half checked, it has an effect on all read operations except those three and on how files contents are presented in File mode and Preview mode and in separate data windows. If checked (fully or half), that is a useful setting to achieve file hash compatibility with ordinary (user level) Windows applications. If not checked at all, that is the setting required for hash compatibility with ordinary forensic tools, and it causes all file-specific read operations to return the data that is stored in the allocated (but uninitialized) clusters from previous usage, for example also for the Recover/Copy command. Changing this setting has an immediate effect even on already opened files, for the next internal read operation. Volume shadow copy host files are treated as if their data was initialized/valid although the NTFS file system says otherwise, to avoid unnecessary complications.

• There is an option to reveal **fragmented files** and directories in newly taken volume snapshots. In evidence objects such items get a special label. When not working with a case such items are partially tagged. The identification can be useful for educational purposes (to find files for which

the file system needs to rememember non-contiguous cluster chains with special data structures, and to better understand using which logic free clusters are picked by file system drivers for allocation) or to draw some rough conclusions about volume usage. (Files are more likely fragmented if they were created later in the lifetime of the file system, at a time when many other files had already been deleted, but many others still existed, leaving allocation holes.)

• You can indicate whether you are interested in getting files included in the volume snapshot whose **clusters** (and therefore data) are totally **unknown**, with only metadata (e.g. just filename and path and/or timestamps), in Ext*, XFS, Reiser* and NTFS. If fully checked, all previously existing files of which metadata only is known will be included in a volume snapshot. If not checked at all, those files will be ignored. If half checked, only files for which more than just the name or timestamps are known will be included, but not directory entry remnants in Ext* or Reiser file systems.

• **Quick snapshots without cluster allocation** speeds up taking a volume snapshot (in particular for the file systems Ext2, Ext3 and ReiserFS, and in particular also when the volume snapshot files are created across a slow USB 1.1 interface or network), however, causes WinHex to lose its ability to tell each sector's and cluster's allocation (for which file it is used). You may use the command "Take New Volume Snapshot" of the Tools menu to update the view of a volume, e.g. after unchecking this option.

• With the option **Keep volume snapshots between sessions** enabled, all information on file systems in opened volumes collected by WinHex (Disk Tools menu and/or Specialist menu) remains in the folder for temporary files even when WinHex terminates. WinHex can then reuse the snapshots in later sessions. Volume snapshots of evidence objects in a case are *always* kept, regardless of this setting, in that evidence object's metadata subdirectory.

• **Keep more data** of the volume snapshot **in memory**, e.g. for much quicker sorting by timestamps.

• To accelerate various operations such as volume snapshot refinement, logical searches, and especially the optional dynamic context preview rendering around search hits in the search hit list, X-Ways Forensics can **keep** more decompressed contents of **file archives** in the volume snapshot **cache**. This generally accelerates opening files in archives again after the first time, especially nested archives. The volume snapshot cache could become very large that way. It can be discarded optionally whenever closing the data window if you like (useful if you are done dealing with that evidence object for the moment, or done with the entire case), and that is a case-specific setting in the case properties. Once discarded, files can get cached again afterwards at any time if/when they are opened again, if the option for that is active. If the box for caching is half checked, that means only nested archives are cached.

• Option to convert certain RTF-formatted e-mail bodies from Outlook e-mail archives to plain UTF-8 (when extracting e-mails) to be able to better view generated .eml files in external e-mail clients and to allow for the alternative .eml preview.

• The option for an alternative interpretation of extended timestamps has an effect when including the contents of file archives in the volume snapshot. Cf. chapter "Archive exploration".

• **UFDR** reports can be added as evidence objects just like normal Zip archives. The file report.xml in .ufdr archives is presented as a virtual file because it contains metadata for the examination and is not an original file. It can optionally be parsed to present all the other files in the archive with their original timestamps and in their original paths whenever possible. (The timestamps that the other files have according to the Zip archive records can optionally be discarded altogether if you find them too unreliable/misleading.) If report.xml interpretation is fully selected, X-Ways Forensics will also extract messages of certain types and present them as events. Messages of the following types are usually supported:
Instant Messages: Android CallLog database
Instant Messages: Android
Instant Messages: Phone
Chats: Kik Messenger
Chats: Native Messages
Chats: WhatsApp
Chats: Snapchat
Chats: Facebook Messenger
Chats: Instagram
Chats: TikTok
Chats: Telegram
Chats: Skype

9.5 Viewer Programs & Gallery Options

Here you may activate the separate viewer component and specify the path where it is located. By default and for reasons of simplicity the files of the viewer component should be extracted from the zip archive into the same directory where you run the main program (X-Ways Forensics or X-Ways Investigator). If you did that, you can simply enter a period as the path. The period is a placeholder for the directory in which the main program is executed. More complicated relative paths are also possible, for example "..\viewer" refers to a directory named "viewer" in the parent directory of the installation directory. Absolute paths are possible as well, of course, for example "X:\Viewer854".

The viewer component is a separate download because it is not as frequently updated as X-Ways Forensics and because the same version and copy of the viewer component can be shared by multiple installations/versions of X-Ways Forensics. It is definitely recommended to activate the viewer component on your own examination machine because its functionality is required in particular to view various file types and to decode text for searches and indexing. The viewer component stores its settings in files in the Windows profile directory of the current user. For that reason you may not want to activate the viewer component when running the main program on a live system that is not your own computer and that you wish to examine, to prevent avoidable changes to that system.

Keeping Track of Viewed Files: With a forensic license, the program can optionally keep track of which files were already viewed and flag them visually with a green background color around the tag. This is especially useful when reviewing hundreds or thousands of documents or pictures

over a longer period, to avoid accidentially viewing the same documents multiple times. A file can automatically be flagged as already viewed when viewing it in full window or Preview mode, when viewing pictures in the gallery, or when identifying a file as known good based on the hash database.
When identifying duplicate files based on hash values, and one of the files has been marked as already viewed, then the duplicates can optionally be marked as already viewed, too. Similarly (only if the corresponding checkbox is fully checked), if files have been marked already as having duplicates and their hash values are available, when they are viewed, known duplicates within any open volume will be marked as already viewed at the same time, but this is potentially slow when used in conjunction with the gallery. When viewing a file with further hard links (which are also duplicates), those will be automatically marked as already viewed as well, except in HFS+.
To manually mark files as already viewed, you can press Alt in combination with the cursor keys. Alt+Left removes the mark. You can also right-click the tag area of a file in the directory browser to mark it as already viewed or to remove that mark.
A directory is considered viewed if all the files and subdirectories that it contains are flagged as such.

If the internal graphics viewing library is used to view pictures, not the viewer component, then optionally the picture viewer window can be closed automatically when a new picture is viewed (if "View multiple pictures simultaneously" is not selected). In that case an auto update option is available that allows to automatically load the next picture into the single picture viewer window as soon as a new picture is selected, one way or the other, for example with a single mouse click or when creating a label for the preview picture or when pressing one of the arrow keys. This should be useful mainly when working with multiple monitors, where the picture viewer window remains on the 2nd monitor. If pictures are viewed with the internal graphics viewing library, this will automatically adjust the orientation of JPEG photos based on their Exif data if necessary.

An alternative e-mail representation is available in Preview mode (also in the case report). Attachments are not linked directly from this kind of e-mail representation yet in Preview mode. The e-mail headers can optionally be excluded (not Raw mode). Useful with the standard e-mail representation if you would like to see more of the body of the e-mail without scrolling. You can see subject, sender, recipient and dates already in the directory browser, and attachments are listed when exploring the parent .eml file.

"Clean up after GDI font object leaks" mainly has the function to allow for mass operations with the viewer component that potentially permanently consume GDI handles. To avoid a crash for example when generating thumbnails for thousands of PDF files for the case report, this option should be active. By default the check box is half checked. Fully checked means that the necessary checks for handle leakage are performed more often.

Applying Exif orientation metadata in Preview mode, for the View command, in the gallery, for OCR and for Excire is optional and controlled by a 3-state checkbox. If fully checked, the Exif orientation is strictly applied. If half checked (default), it is not applied if X-Ways Forensics thinks it is most likely correct to *not* (further) rotate or flip the picture. Thumbnails and low-resolution alternatives embedded in JPEG files inherit the Exif orientation from their parent files.

When viewing or previewing pictures with the internal graphics display library, low resolution

pictures are automatically magnified to some extent. This depends on which factor you feel comfortable with at most to avoid pixelation, and the maximum can be set in the Options | File Viewing dialog window. By default, only natural magnification factors are used (100%, 200%, 300%, ...) to avoid the need for interpolation, but there is a checkbox to change that. The difference can be seen best with a small picture and a high maximum magnification when you resize the preview area. Under the constraints of the user-editable maximum magnification and the potential restriction to simple pixel multiplication (no interpolation), pictures are magnified in Preview mode and in view windows of the internal graphics display library to the maximum extent possible given the size of the preview mode area and the size of the screen workspace, respectively. The magnification applied to pictures in Preview mode is displayed in the lower left corner of the preview area in percent. The magnification applied in a view window of the internal graphics display library is displayed in the window caption after the filename.

Gallery Options

Gallery screen space is utilized very efficiently because thumbnails are not forced to be squares. You can specify your preferred thumbnail width and height separately, in pixels. The specified dimensions will be dynamically adjusted (increased) to best fill the available screen space without partial thumbnails being visible. Since most photos and practically all videos are shot in landscape format, you may want to select width and height accordingly (width larger than height) when viewing pictures. Document thumbnails can often be freely adjusted to any rectangle shape, for example those representing word processing documents or spreadsheets, but not presentations. For most documents other than presentations, portrait format feels like a more natural way of representation. The aspect ratio of the width and height that you specify is displayed in the options dialog to quickly give you a rough idea how compatible the measures will be with ordinary photos, videos or documents.

- If the creation of thumbnails for **pictures within** large (e.g. solid RAR) **archives** for **gallery** view is too slow, you may want to disable it. This will also disable search hit context preview for search hits in files in archives.

- If large JPEGs already contain embedded thumbnails and those have been included already in the volume snapshot or if internal thumbnails have been computed for large pictures, then they can be optionally used as **auxiliary thumbnails** in the **gallery** to represent the main picture. The benefit is that they are of course *much* quicker to load than the main large picture. Also video stills exported from videos can be used as auxiliary thumbnails to represent the video, even all of them dynamically rotating if fully checked.

- The gallery has its own "Dbl-click=View instead of Explore" 3-state option, analogously to the directory browser. By default, double-clicking means View in the gallery.

- There is an option to view files with a single click in the gallery instead of with a double click. Useful for example if you wish to view certain pictures on a separate monitor, where you do not have to close the view window to see the gallery again, when not viewing all pictures one after the other (for which the Page Up or Dn key is more efficient).

- Another option allows to tag a file by clicking anywhere in the thumbnail, not just in the tag

square. That makes it more convenient to tag a large number of files, and is more comfortable that selecting multiple files while holding the Ctrl key.

- The gallery can optionally show thumbnails for any file type supported by the viewer component, including Office documents, PDF, HTML, e-mails, and pictures that the internal graphics viewing library cannot display (e.g. .emf, .wmf, jp2, ...). You can choose between normal and slightly shrunk and strongly shrunk thumbnails of documents. Shrunk thumbnails show much more detail from an original document and the original layout, but at the cost of readability. Larger fonts (in particular captions) in an original document, if not shrunk, are typically readable in the thumbnail and can already give you an idea what kind of document it is even if don't view it, so you can more quickly find the documents that you are looking for. Plus, you will be able to see which files can be nicely viewed with the viewer component at all. It is strongly recommended run X-Ways Forensics with Aero enabled in Windows when using the gallery with the non-picture option.

  If this box is only half checked, non-picture files will be shown as thumbnails only if their type status is confirmed or newly identified or mismatch detected. That means files with unintelligible data will more likely ignored, and supported document types with proper formatting are more likely shown. Files that are larger than 16 MB are not represented with a thumbnail, for performance reasons. X-Ways Forensics tries to abort the generation of a thumbnail if it takes longer than a few seconds. If the generation of a true thumbnail is unsuccessful, you may see a viewer component error message like "Operation cancelled" in tiny red letters in the thumbnail instead. If thumbnail generation is not even attempted by X-Ways Forensics, you will just see the filename and an icon.

- Thumbnails of true-color pictures can optionally be color-adjusted in the gallery. This option is meant for law enforcement users whose job is to review child pornography photos, to reduce the mental impact and stress level. If the checkbox for this option is fully checked, the thumbnails will appear in grayscale. If half checked, color swapping will take place in such a way that human skin will appear very unnatural. Thumbnails can also be intentionally blurred, slightly or more noticeably, for the same purpose. All of these effects can be limited to just pictures that were identified as notable already.

- The timeout in milliseconds when loading pictures with the internal graphics display library for the gallery is aborted (e.g. corrupt or unsupported or extremely large picture files), is user-definable. Timeouts for loading pictures for picture analysis and processing and for the XWF_GetRasterImage() API function and for the report are twice as long as defined for the gallery.

Text decoding for logical searches, indexing, and text submode of Preview mode

Crash-safe text decoding: If enabled, text extraction from certain file types for logical searches and indexing will be done by the viewer component in a separate process, such that if the viewer component crashes or becomes unstable, it does not render the main process (X-Ways Forensics) unstable or cause it to crash.

There is an option to filter out spaces around common Chinese characters in decoded text. Such

spaces could otherwise appear unexpectedly for example when processing certain PDF documents and can thwart keyword searches in Chinese.

Buffer decoded text for context preview: If enabled, the result of the text extraction from certain file types for logical searches and indexing will be stored by X-Ways Forensics in the volume snapshot for reuse when searching/indexing again, to save time.

External programs, custom viewer programs

You may select your preferred text editor and HTML viewing program. The HTML viewer program can be e.g. MS Word or NVU, i.e. a program that can be used to further edit the HTML case reports the X-Ways Forensics can create automatically. For merely viewing and printing we recommend Internet Explorer.

You can also specify the path of the .exe file of MPlayer, a program that allows X-Ways Forensics to extract pictures from videos. If mplayer.exe is found in a subdirectory \MPlayer of the installation directory of X-Ways Forensics, it will defined as the video extraction program and as an external viewer program automatically. Relative paths started with .\ or ..\ are supported, where . stands for the directory from which X-Ways Forensics is executed and .. its parent directory. Please note that we cannot provide support for external programs.

You may also specify up to 32 custom viewer programs that can be conveniently invoked from inside X-Ways Forensics via the directory browser context menu. Also, you may specify which file types you prefer to view in the program that is associated with their extension in your system, typically file types that the separate viewer component does not support. There is a checkbox labelled "Append type as extension if newly identified" checkbox. Allows to more easily get Windows to run the right program for misnamed files, files without extension etc. The paths of these external viewer programs are defined in a separate file, named Programs.txt, so that it is easy to share a collection of external programs separately, or keep them when taking over all other program settings from someone else. In that text file you may also change absolute paths to relative paths (with . and ..), for programs that are as portable as X-Ways Forensics itself and that you wish take along on a USB stick for analyses of live systems. If you need to call external programs from within X-Ways Forensics with certain parameters in addition to the name of the file that they should open, you can specify those parameters in the same line of Programs.txt, delimited from the path of the executable file with a tab. The name of the file to open will automatically be appended at the end, after your own parameters, unless you include the placeholder %1 anywhere in your list of parameters. That placeholder will be replaced with the filename.

Tesseract: OCR

OCR-derived text is ignored if it does not contain at least x contiguous useful characters. Such OCR results will not be stored/output/copied/indexed/searched. This is beneficial if you apply OCR to unknown/random/ordinary pictures (i.e. not known textual data), to reduce the number of files that later will (misleadingly) respond to the Description filter for files with OCR-derived text or for which child objects are (unnecessarily) created by the "Copy: Extracted Text" function etc. A "useful" character is defined here as a character with an ASCII/Unicode value of 0x30 or higher. That means whitespaces <=0x20 are not counted, and neither are the printable characters

!=#$%&'()*+,-.& (0x21-0x2F range) because some of them are occasionally misdetected in random pixels. All real letters in any language count, and so do numbers ("0" through "9").

9.6 Undo Options

The availability of the “Undo” command depends on the following options:

- Specify how many sequential actions are to be reversed by the Undo command. This option does not affect the number of reversible keyboard inputs, which is only limited by the available RAM.

- In order to save time and space on your hard disk, you can specify a file size limit. If a file is larger than this limit, backups will not be created and the Undo command is not available except for keyboard input.

- Automatically created backups for the internal use with the Undo command are deleted by WinHex when closing the file, if the corresponding option is fully selected. If it is partially selected, they are deleted when WinHex terminates.

- For all kinds of editing operations you choose whether they should be reversible or not. If so, an internal backup is created before the operation takes place.

9.7 Security Options

- Before modifications to an existing file are saved (i. e. before the **file** is **updated**), you are by default **prompted** for confirmation, but this behavior can be changed.

- If any of the operations Refine Volume Snapshot, Logical Search, or Indexing crashes when processing a file, X-Ways Forensics when started next time will tell, which file was likely responsible for the crash, if you had it **collect information for** a **crash report**. If fully checked, should volume snapshot refinement crash the program, restarting the program will also point out which suboperation exactly was applied to the problematic file(s) when the program crashed. It has not been tested whether this enhanced granularity of logging might cause any noticeable slowdown. There may be multiple candidates for the problematic file that triggered the instability if multiple worker threads were active at the time of a crash.

- X-Ways Forensics has the ability to automatically resume certain operations after a crash (an involuntary program termination), without any user intervention. The currently supported operations are the stages "file header signature search" and "processing of individual files" of volume snapshot refinement when invoked from the main menu or the command line or by adding evidence objects to a case. Following a crash, these operations will be resumed at a point that depends on when the volume snapshot was last saved. (That in turn depends on the auto-save interval in the case properties because whenever the case is saved, the volume snapshots of all open evidence objects are saved as well. You can also save the case manually

while volume snapshots are being refined.) If it is not clear which particular file has triggered a crash because you were running the operation with additional threads, then the operation will be resumed first with no additional threads. With some luck, that will not trigger the crash again. If it does, the operation is resumed once more. Once the exact file is identified, it will be skipped automatically. In case of a crash during the file header signature search, the sector that triggered the creation of a problematic file will be skipped.

- Only in Preview and Beta releases, you can simulate crashes if you wish to observe, test, or demonstrate this automatic work-around, for example because you wish to benefit from it when running X-Ways Forensics more or less automatically with command line parameters, and need to react to the situation where one instance of X-Ways Forensics disappears and is immediately replaced by another instance that you didn't start yourself. For the simulation, you provide the name of a file that you want to trigger a crash in the supported operations. The filename should be rather unique and target ideally just one file that you know is in the initial volume snapshot or that you expect to be added to the refined volume snapshot. It's case-sensitive. Note that if you have X-Ways Forensics assign names based on incrementing numbers to carved files, and you make it simulate a crash with a carved file whose name is expected to be 012345.jpg, then even if X-Ways Forensics successfully learns to avoid the sector where that file is found in the file header signature search, the next carved file after that might be named 012345.jpg as well (depending on the file type), triggering yet another crash. Unique names of carved files are those derived from the intelligent naming option (like "Canon DIGITAL IXUS 950 IS 2007-07-01 12:01:46.jpg" or from the option to name files based on start sectors. To simulate a random, non-repeatable crash, you can simply terminate X-Ways Forensics with the Windows Task Manager.

- X-Ways Forensics can monitor the additional threads employed for volume snapshot refinement and attempts to terminate and resume hanging threads if found to be unresponsive for e.g. 15 minutes. This assumes that the main thread/user interface is not frozen. Even if a particular file takes longer to process (e.g. large Outlook PST e-mail archive with many e-mails and attachments), the corresponding thread makes it known that it is still alive, so that alone will not trigger any recovery measures. If a **thread** is **recover**ed, the Messages window will inform you with the note "A thread involved in this operation appears to be unresponsive" (depends on the language of your Windows installation) and identify the processed file and sub-operation, followed by the word "Continue...". The file will also receive the label "Omitted".

- **Output messages about exception**s: Determines the verbosity of the program in case of exception errors. If totally unchecked, only exception errors with a potentially serious impact (like considerably incomplete analysis results) will be brought to your attention in the Messages window. If fully checked, all of them will be output, even those that occur typically with corrupt files only and have no negative impact on other analysis results. The middle state is a reasonable compromise. Regardless of this option, exception errors will be noted in the error.log file.

- All notices and warnings output to the **Messages** window can optionally be automatically saved in a text file "**msglog.txt**" in the installation directory. If at that time a case is active, the notice/warning will be written to the msglog.txt file in the log subdirectory of that case

instead. The default behavior is that the box is half checked. Fully checked means that even messages in the Progress indicator window (descriptions of operations as well as names of processed files) are output.

- Use the option **Check for virtual memory changes** to make sure the memory editor inspects the structure of virtual memory every time before *reading* from or *writing* to it. If the structure has changed, a possible read error is prevented. Especially under Windows NT the checking may result in a loss of speed. When editing the “entire memory” of a process, WinHex generally *never* checks for alterations before reading, even if this option is enabled.

- **Strict drive letter protection**: Only available with a forensic license.Active by default in X-Ways Forensics. Ensures that saving and editing files is only possible on certain drive letters, namely those that X-Ways Forensics even when examining a live system can assume are located on the examiner's own media. They are: 1) the drive letter that hosts the active case if one is active, 2) the drive letter with the directory for temporary files, 3) the drive letter from which X-Ways Forensics was run and 4) the drive letter that contains the directory for image files.

- Decide whether or not WinHex shall generally **prompt before executing a script or an X-Tension**, or only before executing the same through command line parameters.

- The **key** that is required for encryption and decryption can be entered in a normal edit box. Optionally, you **enter** it **blindly** (asterisks are displayed instead of the actual characters). In this case you have to confirm the key in a second edit box to detect typos.

- By default, the **key** is **kept in** main **memory** (in an encrypted state) as long as WinHex is running, so that you do not have to type it again and again if you use it several times. Possibly you prefer WinHex to erase the key after use.

- Internal passwords/encryption keys, such as for encrypted .e01 evidence files and cases, optionally support Unicode, depending on the state of the check box "Encode internal passwords as UTF-8". Make sure the box is *un*checked if you are using or have previously used passwords that contain non-ASCII characters to preserve password/encryption compatibility with v21.5 and earlier.

- Optionally, checksums with multi-byte accumulators (16-bit, 32-bit, and 64-bit checksums) are computed byte-wise instead of adding units that are equivalent in size to the accumulator itself, e.g. 4 bytes for 32-bit checksums. Both variants exist in real life applications.

- The CRCs in .e01 chunks can be automatically checked on the fly when chunks are read, and any discrepancies will be reported in the Messages window. This costs a little computing power.

- Whether a password verification hash for .e01 evidence files created with 256-bit AES encryption is included in the .e01 evidence file or not is up to you to decide. The hash allows X-Ways Forensics to check whether the password that you enter when opening such an image is correct.

- If an .e01 evidence file that found to have very inefficient layouts (less than 32 chunks per table section or compressed chunks with a compression ration of less than 0.1%), that is brought to the users attention so that they can avoid whatever software or hardware created that image.

- If certain metadata about large .e01 evidence files is kept in separate files (with the .xmet extension), that allows X-Ways Forensics to re-open the images much faster next time. This can make a big difference if an image is stored on media with slow access, in particular remote network drives. If this box is fully checked, the separate file is stored in the same directory as the image itself, so that even other cases / other users that open the same copy of the same image benefit from the increased performance if the separate file has been created once before. If half checked, the separate file is stored in the evidence object's internal metadata directory of the current case. Paranoid users who do not only write block suspect storage devices, but also their own storage devices with images, are well advised to half-check this box for obvious reasons if they wish to benefit from the increased speed.

A general password collection can be maintained by clicking a button in the security options dialog window. It is stored in the file “Password.txt”. The password collection of a newly created case is initialized with that general password collection. The password collection of a case is used with encrypted archives as well as encrypted documents whenever the case is loaded.

One of the buttons in this dialog box allows to exhaust system memory, for example in order to get comparable results with performance tests that could get distorted if for example Windows still has parts of an image file in its file buffer. Under Windows 10 or newer it requires this setting in Windows: Local security policy | Local Policies | User Rights Assignments | Lock pages in memory. Add user to this policy. Restart to make it take effect.

Another button allows to schedule a shutdown or hibernation of the machine after a certain number of minutes. Is guaranteed to work only if nothing keeps the machine from powering down, e.g. other application programs with unsaved work etc. If you half-check to proceed "brutally", that should power down the machine even if an application is hung. If fully checked, that will not even wait for other applications that prompt the user what to do with any unsaved work longer than a few seconds. If you exit the instance of WinHex/X-Ways Forensics in which you have scheduled the shutdown, the shutdown won't happen. It is possible to cancel a previously scheduled shutdown without restarting the program.

9.8 Search Options

Match case: If a search is case-sensitive, that means that upper and lower case characters are distinguished and e. g. “Option” with a capital “O” is not found in the word “optionally”. By unchecking the checkbox, you search for all upper-case/lower-case variants of the search terms. Searches are fully case insensitive only with the Simultaneous Search, with the Find Text command only for letters from the Latin/English alphabet and German umlauts. In the Simultaneous Search you may use case-sensitive and non-case-sensitive search terms at the same time if the “Match case” option is half selected. In that case you may prepend search terms with

“case:” to mark them as case-sensitive.

Unicode: The specified text is searched in UTF-16 Little Endian. The simultaneous search allows to search for the same text at the same time in Unicode and in other code pages.

You may specify a **wildcard** (one character or a two-digit hex value), which represents one byte. For example this option can be used to find “Speck” as well as “Spock” when searching for “Sp?ck” with the question mark as the wildcard.

Only whole words: The search term is found only if it occurs as a whole word, i.e. if delimited from other words by any character other than a...z, A..Z and German and French letters (e. g. by punctuation marks, blanks, binary control codes, digits). If this option is enabled, for example "tomato" is not found in "automaton". In a Simultaneous Search either all search terms are searched as whole words or only those that are indented (prepended with a tab character) or none, depending on the state of the corresponding check box. If you wish to combine the indention for a search as a whole word with the "case:" prefix for case sensitivity, enter the "case:" prefix first and then insert the tab character for the indention.

For the Simultaneous Search function "whole words" is a 3-state option. The middle state allows to match starts of words only (require a word boundary at the beginning of the search hit). That means e.g. with "box" you can find "boxes" at the same time (but not "checkbox") and with "tend" you can find "tends" and "tended" at the same time (but not "attended" or "extended"). This is otherwise possible with regular expressions only, and if you wish to search some search terms as whole words and others as starts of words at the same time you still need to use regular expressions, please.

You may customize the word boundary detection for languages that utilize the Latin 1 code page, i.e. make it more strict (for less search hits) or more relaxed (for more search hits), by defining the alphabet of characters that are considered letters (i.e. characters belonging to words) as opposed to non-word characters. A word character followed by a non-word character or the other way around is considered a word boundary. There are three easy-to-use pre-defined settings. The setting for the most thorough search results is the default. Users that are overwhelmed by garbage hits for short keywords in non-text data such as Base64 or binary garbage may want to try the other two options. These other two options could lead to valid search hits being missed in some constellations (depends on the file format), but can still be justifiable as a great time saver for searches in text documents, e.g. rather in electronic discovery, rather not in computer forensics.

For more explanation and an example of how the whole words option works, please read on: A word boundary is a boundary between two consecutive characters of which one character is a word character and the other character is not a word character. If two consecutive characters are both word characters (e.g. "ns"), then obviously the "s" does not start a new whole word, and the "n" cannot be the end of a whole word. It can be somewhere in the middle of a whole word (e.g. "mansion"), but in between these two characters "ns" there is definitely no word boundary. If both characters are non-word characters (e.g. "! ", exclamation mark followed by a space), then obviously the position between the two is not a word boundary either. The exclamation mark cannot be the end of a word (cannot occur anywhere within a word), and the space cannot be the start of a word (cannot occur anywhere within a word either, excluding compound words). If you are searching for "man" as a whole word within "our mansion", then XWF will

provisionally/internally find "man", and then first check whether the character before the "m" is a word character. That character is a space. A space character is not a word character. Then it also checks whether "m" is a word character according to the alphabet. It is. That means there is a word boundary before the "m". Next XWF needs to check whether "n" and "s" are word characters. Both are. That means that after the "n" there is no word boundary. Hence the three letters "man" within "mansion" are not considered a whole word occurrence of "man".

The whole words only restriction of the Simultaneous Search is not applied to search hits that are not words according to the user's selected alphabet definition (checking only the first and the last character in the search hit). For example if you are searching for "LOL!!", then this cannot possibly be a whole word because the exclamation mark is not a letter and thus not contained in the defined alphabet (well, unless you have added the exclamation mark to it manually). However, the RegEx word boundary indicator \b is still applied in such a case, for example to be able to search for certain data in between words, data that is not considered a word itself.

In addition to the alphabet of characters for the Latin 1 code page (for all Western European languages), an optional additional alphabet can be defined for letters of another language. If activated, it is used for searches in UTF-16, UTF-8 and regional ANSI/OEM/IBM/ISO/Mac code pages with only 1 byte character such as for Cyrillic, Greek, Turkish, Arabic, Hebrew, Vietnamese, and various Central/Eastern/South Eastern European languages. The Cyrillic alphabet is predefined.

Search direction: Decide whether WinHex shall search from the beginning to the end, or downwards or upwards from the current position.

Condition: Offset modulo x = y: The search algorithm accepts search string occurrences only at offsets that meet the given requirements. E.g. if you search for data that typically occurs at the 10^{th} byte of a hard disk sector, you may specify x=512, y=10. If you are looking for DWORD-aligned data, you may use x=4, y=0 to narrow down the number of hits.

Search in block only: The search operation is limited to the current block.

Search in all open windows: The search operation is applied to all open edit windows. Press F4 to continue the search in the next window. If “Search in block only” is enabled at the same time, the search operation is limited to the current block in each window.

Count occurrences/Save occurrence positions: Forces WinHex not to show each single occurrence, but to count them. If this option is fully enabled, WinHex will enter all occurrences into the Position Manager.

Search for “non-matches”: In “Find Hex Values” you may specify a single hex value with an exclamation mark as a prefix (e.g. !00) to make WinHex stop when it encounters the first byte value that *differs*.

Regular expressions: Search option available with the Simultaneous Search only. Regular expressions are a powerful search tool. A single regular expression may match many different words. Either all search terms are considered regular expressions or only those prepended with

"grep:" or none, depending on the state of the corresponding checkbox. You may prepend a search term with both "case:" (see above) and "grep:" in that order. The following characters have a special meaning in regular expressions, as explained below: ( ) [ ] { } | \ . # + ?. Where these special characters are to be taken literally, you need to prefix them with a backslash character (\).

The | operator is used to denote alternative matches. You can use the regular expression *car (wheel|tire)* to search for the words "car wheel" and "car tire". Any match must equal the parts before, after, or between any | operators present. The effect of | is only limited by parentheses.

. and # are wildcards: . matches any character, # matches any numeric character. You can define sets of characters with the help of square brackets: [xyz] will match any of the characters x, y, z. [^xyz] will match any character except x, y, or z. You can define ranges of characters using a dash: [a-z] matches any lower-case letter. [^a-z] matches all characters except lower-case letters. The listing may comprise individually listed characters and ranges at the same time: [aceg-loq] matches a, c, e, g, h, i, j, k, l, o, and q. Within [] brackets, the characters .*+?{}()| are not treated as special characters, but literally.

\b stands for the start or end of a word, i.e. the boundary between a word character and a non-word character. Which characters/letters are considered word characters by the Simultaneous Search is user-defined. The start and end of a file also count as word boundaries. \b is only supported at the start and/or at the end of the search term, and not in conjunction with |. \b, ^, and $ anchors only work only when searching in evidence objects of a case, and not for index searches.

Byte values that correspond to ASCII characters that cannot be easily produced with a keyboard can be specified in decimal or hexadecimal notation: For example, \032 and \x20 are both equivalent to the space character in the ASCII character set. This kind of notation is supported even in between square brackets. E.g. [\000-\x1f] matches non-printable ASCII characters.

Multiplier characters (*, +, and ?) indicate that the preceding character(s) may or must occur more than once (see below). Complex example: a(b|cd|e[f-h]i)*j matches aj, abj, acdj, aefij, aegij, aehij, abcdj, and abefij.

Brief overview of supported syntax features (everything else is interpreted literally)

. A period matches any single character.
A pound sign matches any numeric character [0-9].
\nnn A byte value specified with three decimal digits (0...255)
\xnn A byte value specified with two hexadecimal digits (0…FF).
For example, \x0D\x0A is the Windows line break.
\unnnn A Unicode value specified with four hexadecimal digits.
Depending on the selected code page(s), corresponds to different byte values.
? Matches one or zero occurrences of the preceding character or set.
* Matches any number of occurrences of the preceding character, including zero time.
+ A plus sign after a character matches any number of occurrences of it except zero.
[XYZ] Characters in brackets match any one character that appears in the brackets.
[^XYZ] A circumflex at the start of the string in brackets means NOT.
[A-Z] A dash within the brackets signifies a range of characters.

\ Indicates that the following special character is to be treated literally.
{X,Y} Repeats the preceding character or group of characters X-Y times.
(ab) Functions like a parenthesis in a mathematical expression.
Groups ab together for +, ?, *, | and {}.
a | b The pipe acts as a logical OR. So it would read "a or b".
\b Matches a word boundary.
^ Matches the start of a file.
$ Matches the logical or physical end of a file, depending on the search options.

Regular Expression Examples

E-mail addresses
[a-zA-Z0-9_\-\+\.]{1,20}@[a-zA-Z0-9\-\.]{2,20}\.[a-zA-Z]{2,7}
(the + before the @ is supported in Gmail addresses)

Internet addresses starting with http://, https://, ftp://
[a-zA-Z]+://[a-zA-Z0-9/_\?$&=\-\.]+

Visa and Mastercard credit card numbers
[^#a-z][45]###############[^#a-z]
[45]###-####-####-####
[45]### #### #### ####
(ideally check results via an X-Tension with the Luhn algorithm to reduce the number of false hits and search without [^#a-z])

Allow overlapping hits: If you use regular syntax to search for search hits of variable length, multiple valid hits at the same location may be the result. If you search for example for e-mail addresses, and the search algorithm is fed with the character sequence "mail@x-ways.com", then it will determine that the characters from the "m" in "mail" match the regular expression and it will record a hit. After that, it proceeds with the "a" in "mail" and realizes, that ail@x-ways.com fits the bill as well, and so do il@x-ways.com and l@x-ways.com. All of these might be valid e-mail addresses. So the search algorithm is entirely right, but typically users do not wish to see those additional hits. So if you do not allow for overlapping hits, new hits are recorded only after the "m" in ".com". Not allowing overlapping hits means to exclusively assign the characters covered by a hit to that hit and not to potential other hits any more.

Search window, proximity searches

The search window width is 128 bytes by default. That means it is not guaranteed that with a variable-length regular expression (i.e. using {}*+ syntax) you can find data that is longer than 128 bytes. You may increase the search window width if you need to cover more than that.

This is needed for example for proximity searches. If you require that a document contains two search terms at the same time, and that the search terms should occur close to one another, you could search for these search terms with two regular expressions and specify the maximum distance allowed between them as the second parameter in the braces:

keyword1.{0,*maxdistance*}*keyword2*
keyword2.{0,*maxdistance*}*keyword1*

The search window width in bytes required when searching with an 8-bit character set is the sum of *maxdistance*, length(*keyword1*) and length(*keyword2*).

Please note that the preferred method to find two search terms near to each other is the NEAR combination in the search term list, when two search terms are already combined with a logical AND, after they have been searched for separately.

9.9 Replace Options

Prompt when found: WinHex awaits your decision when an occurrence has been found. You may either replace it, continue or abort the search.

Replace all occurrences: All occurrences are replaced automatically.

Case sensitive: The characters that are to be replaced are searched using this option (cf. Search Options).

Unicode character set: The specified characters are searched and replaced in Unicode format (cf. Search Options).

You may specify one character or a two-digit hex value as a **wildcard**. This is usually done in the search string. If the *substitute* contains a wildcard, the character at the corresponding position in an occurrence will not be changed. Thus, “black” and “block” can be replaced simultaneously with “crack” and “crock” (enter “bl?ck” and “cr?ck”).

Only whole words: The searched string is recognized only if it is separated from other words e.g. by punctuation marks or blanks. If this option is enabled, “tomato” is not replaced in “automaton”.

Search direction: Decide whether WinHex shall replace from the beginning to the end, or downwards or upwards from the current position.

Replace in block only: The replace operation is limited to the current block.

Replace in all opened files: The replace operation is applied to all files not opened in view mode. If “Replace in block only” is enabled at the same time, the replace operation is limited to the current block of each file.

Hint:

WinHex is able to replace one string or hex value sequence with another one that has a different length. You will be prompted, which of the following methods shall be applied:

1st method: The data behind the occurrence is moved due to length difference. So the file size is changed. This method must not be applied to certain file types, such as executable files. It is even possible to specify nothing as the substitute, which means all occurrences will be removed from

the file!

2nd method: The substitute is written into the file at the position of the occurrence. If the substitute is shorter than the searched character sequence, the exceeding characters will remain in the file. Otherwise even the bytes behind the occurrence will be overwritten (as far as the end of the file is not reached). The file size is not affected.

10 Miscellaneous

10.1 Block

You can mark a range of bytes or sectors of an open file or disk as a "block". This part can be manipulated by several functions in the edit menu just as selections in other Windows programs. If no block is defined, these functions usually are applied to the whole file or disk.

The current position and size of the block are displayed in the status bar. Double-clicking the right mouse button or pressing the **ESC** key clears the block.

10.2 Modify Data

Use this command to modify the data within the block or within the whole file, in case no block is defined. In this version of WinHex, four types of data modifications are available. Either a fixed integer number is added to each element of the data, the bits are inverted, a constant is XORed with the data (a simple kind of encryption), ORed, or ANDed, bits rotated left in a circular pattern (first byte rotated by 1 bit, second byte by 2 bits, and so on), bits are shifted logically, or bytes are swapped. The XOR function supports 8, 16 and 32 bit constants. The desired granularity of the mathematical operation will be deducted from the number of hex digits that you enter (2, 4, or 8), so you can use leading zeroes to force interpretation as a 16-bit or 32-bit integer constant even if the actual value is as small as an 8-bit integer. By shifting bits, you can simulate inserting or removing single bits at the beginning of the block. You may also shift entire *bytes* (currently to the left only, by entering a negative number of bytes). This is useful if you wish to cut bytes from a very huge file in in-place mode, which would otherwise require the creation of a huge temporary file.

Addition

Specify a positive or negative, decimal or hexadecimal number, which is to be added to each element of the current block. An integer format defines size (1, 2 or 4 bytes) and type (signed or unsigned) of an element.

There are two ways how to proceed if the result of the addition is out of the range of the selected integer format. Either the range limit is assumed to be the new value (I) or the carry is ignored

(II).

Example: unsigned 8-bit format
I. FF + 1 → FF (255 + 1 → 255)
II. FF + 1 → 00 (255 + 1 → 0)

Example: signed 8-bit format
I. 80 - 1 → 80 (-128 - 1 → -128)
II. 80 - 1 → 7F (-128 - 1 → +127)

- If you decide to use the first method, WinHex will tell you, how often the range limit has been exceeded.

- The second method makes sure the operation is reversible. Simply add -x instead of x based on the same integer format to recreate the original data.

- When using the second method it does not make a difference whether you choose a signed or an unsigned format.

Reverse byte order

This command assumes all data to consist of 16-bit elements (32-bit elements resp.) and swaps high-order and low-order bytes (and high-order and low-order words resp.). Use it in order to convert big-endian into little-endian data and vice versa.

10.3 Conversions

WinHex and X-Ways Forensics offer the "Convert" command of the Edit menu for easy conversions of different data formats and for encryption and decryption as well as compression and decompression (depending on license type). The conversion can optionally be applied to all opened files instead of only the in the active data window. The formats marked with an asterisk (*) can also be converted block-wise, i.e. they do not have to be applied to the whole data in the data window. The following formats/conversion methods are supported:

• ANSI ASCII, IBM ASCII (two different ASCII character sets)*
• EBCDIC (an IBM mainframe character set)*
• Lower-case/upper-case characters (ANSI ASCII)*
• Binary (raw data)
• Hex ASCII (hexadecimal representation of raw data as ASCII text)
• Intel Hex (=Extended Intellec; hex ASCII data in a special format, incl. checksums etc.)
• Motorola S (=Extended Exorcisor; ditto)
• Base64
• UUCode
• Percentage URL Encode
• Quoted Printable
• Stretch packed 7-bit ASCII to readable 8-bit ASCII, useful e.g. for SMS from conventional

mobile phones.
• Conversion of so-called Nandroid backup files of the NAND flash memory of Android devices to regular raw images.
• ZLIB compression/decompression
• LZFSE compression/decompression
• ZSTD compression/decompression
• LZVN decompression

Please note:
• When converting Intel Hex or Motorola S data, the internal checksums of these formats are not checked.
• Depending on the file size, the smallest possible output subformat is chosen automatically. Intel Hex: 20-bit or 32-bit. Motorola S: S1, S2, or S3.
• When converting from binary to Intel Hex or Motorola S, only memory regions not filled with hexadecimal FFs are translated, to keep the resulting file compact.

The Convert command can also decompress any number of complete 16-cluster compression units compressed by the NTFS file system and (with a forensic license) entire hiberfil.sys files of certain Windows versions that were copied off an image as well as individual xpress chunks from such files.

Encryption/Decryption

Specify a string consisting of 1-16 characters as the encryption/decryption key. The key is case-sensitive. The more characters you enter, the safer is the encryption. The key itself is not used for encryption and decryption, instead it is digested to the actual key. The key is not saved on your hard disk. If the corresponding security option is enabled, the key is stored in an encrypted state in the RAM as long as WinHex is running.

It is recommended to specify a combination of at least 8 characters as the encryption key. Do not use words of any language, it is better to choose a random combination of letters, punctuation marks, and digits. Note that encryption keys are case sensitive. Remember that you will be unable to retrieve the encrypted data without the appropriate key. The decryption key you enter is not verified before decrypting.

Encryption algorithm: 256-bit AES/Rijndael, in counter (CTR) mode. This encryption algorithm uses a 256-bit key that is digested with SHA-256 from the 512-bit concatenation of the SHA-256 of the key you specify and 256 bits of cryptographically sound random input (“salt”). The file is expanded by 48 bytes to accommodate the 256 bits of salt, and a randomized 128-bit initial counter.
WinHex allows you to encrypt not only an entire file, but also a block of data only. In that case you are warned, however, that no salt is used and no random initial counter is used, so you must not reuse your key to encrypt other data with the same encryption method. The size of the block is left unchanged.

10.4 Sector Superimposition

With this feature you can superimpose other data on top of disks or interpreted images that are opened as read-only. Useful when you need to make minor temporary virtual adjustments to data in sectors within the scope of the program to get it interpreted correctly internally, but do not wish to or are not allowed to alter the sectors on the disk or in the image itself (or cannot because it is not a raw image, but an .e01 evidence file) and also do not want to make another complete working copy of an image that is e.g. 2 TB in size if just 1 byte needs to be changed. Such adjustments can be necessary for example in cases of partitioning or file system metadata corruption, where just a missing magic number keeps WinHex from detecting the file system or just one flipped bit keeps WinHex from finding $MFT in NTFS or just one wrong nibble in the partition table keeps WinHex from recognizing a partition as an LVM2 container partition etc. etc. In these situations you can manually provide and superimpose the corrected data and then hopefully work with the disk or image with no further problems, getting all partitions and files listed immediately as if nothing was wrong. This functionality is intended for advanced users that do not give up easily when at first they see "nothing" and have some understanding of low level data structures and know how to fix them.

You can enable and disable superimposition for the disk or partition in the active data window using the Edit | Superimpose Sectors menu command. This command allows you to select any file with the raw contents of disk sectors. For example, you can create such a file by selecting one or more sectors as a block, copying the block into a new file, making the necessary adjustments (possible even in X-Ways Forensics because ordinary files unlike disks or interpreted images can be edited) and saving that file. When applied, the contents of this file are superimposed to the sectors starting with the sector in which the cursor is located, or if the file is named "*n*.sector", where *n* is a number, it will be applied to the sectors starting with sector *n*, and all other files in the same directory matching the same mask will also be applied to sector numbers as indicated within the filename. You will immediately see the superimposed data when navigating to the affected sectors, and can continue making adjustments to the imposed raw data file if you keep it open in a separate window. As soon as you have saved changes in that window, they will take effect in the data window that represents the disk or partition whose data you are trying to fix when you refresh the view, take a new volume snapshot, define the start of a partition, try again to open a file with a corrupt FILE record etc. etc.

Please note that only complete sectors, not partial sectors, can be superimposed. Superimposition can be active only for one open disk or disk partition or image at a time. If active for a physical partitioned disk or image of a physical partitioned disk, a partition opened from within the physical disk will also show the superimposed data. If desired, you can make a copy (image or cloned disk) of the virtually repaired disk or image with the usual commands while the superimposition is in effect, so that the copy will have the superimposed sectors directly embedded. Active sector superimposition is remembered in an evidence object and automatically re-activated when the evidence object is opened next time, and you will be reminded of that.

10.5 Wiping and Initializing

To securely erase/shred/wipe/redact data in disk sectors, unused disk areas (Disk Tools menu), areas defined on storage devices, partitions or in files as a block, or entire files selected with the Wipe Securely command, and also simply to fill files with certain byte values, WinHex offers the following options:

With a "watermark": You can enter some human-readable characters/words in any language, like "REDACTED" or "wiped", which will be written repeatedly, automatically interspersed with line breaks and null terminators (to avoid complications if large amounts of such data are loaded later by file viewers). Those characters/words can be used to convey a message and give 3rd parties a hint that it is intended that they can't see the original data. If you use any character that is not included in the Latin 1 code page, the entire text will be encoded in UTF-16.

With constant byte values specified in hexadecimal notation: Specify up to 16 two-digit hex values, which will be written repeatedly. Very fast.

With simple pseudo-random byte values: Specify a decimal interval (0 to 255 at max.) for random numbers, which will be written repeatedly. The random bytes are Laplace-distributed. Fast.

With pseudo-random data that simulates encryption: Random data that is supposed to be indistinguishable from encrypted data. Quite fast.

With cryptographically sound pseudo-random data: Cryptographically secure pseudo-random number generator (CSPRNG) named ISAAC, *very* slow.

In case in all open files *either a block or no block is defined*, this command can optionally be applied to all these files at the same time.

To maximize security, if you wish to totally wipe (sanitize) slack space, free space, unused NTFS records, or an entire media, you may want to apply more than one pass for overwriting disk space (up to three).

According to the Clearing and Sanitization Matrix, the standard outlined in the U.S. Department of Defense (DoD) 5220.22-M operating manual, method "c", a hard disk or floppy disk can be cleared by overwriting (once) all addressable locations with a single character. This is usually the hexadecimal value 0x00, but can be any other value. To sanitize hard disks according to method "d", overwrite all addressable locations with a character, its complement, then a random character, and verify. (This method is not approved by the DoD for sanitizing media that contain top secret information.)

The "DoD" button configures WinHex for sanitization, such that it will first overwrite with 0x55 (binary 01010101), then with its complement (0xAA = 10101010), and finally with random byte values.

The "0x00" button configures WinHex for simple initialization, wiping once with zero bytes.

10.6 Disk Cloning

Tools | Disk Tools | Clone Disk. This function copies a defined number of sectors from a source to a destination. Both the source and the destination can be either a *disk* (click the button with the disk icon) or a *file* (click the button with the file icon).

In case both the source and the destination are disks, both disks must have the same sector size. In order to effectively *duplicate* a medium (i.e. copy all sectors), simply copy *all* sectors. Select the appropriate option, so the correct number of sectors is entered automatically. The destination disk must not be smaller than the source disk. As a *disk* you can also select an interpreted image or a partition opened from within a physical disk in the background. As a target you cannot select an interpreted .e01 evidence file as such images cannot be rewritten, only raw images. As a *file* you can only specify unsegmented raw images, e.g. .dd, .001, .img etc., no other image types such as .e01, .vhd., .vmdk etc.

Disk cloning offers options that control the behavior when bad sectors are encountered on the source disk:

- By default, you are notified of the error and prompted for either continuing or aborting the operation. “Log procedure silently” creates a complete log file of the entire operation in the folder for temporary files (filename “Cloning Log.txt”), including a report on unreadable sectors (which cannot be copied), and prevents WinHex from reporting each unreadable sector separately.
- WinHex can either leave a destination sector that corresponds to a damaged source sector unchanged or fill it with an ASCII pattern you specify (e.g. your initials, or something like “BAD ”). Leave the pattern edit box blank to fill such sectors with *zero* bytes. BTW, the chosen pattern is also used to display a bad sector's contents in the disk editor.
- Bad sectors often occur in contiguous groups, and each attempt to read a bad sector usually takes a long time. You may have WinHex avoid such damaged disk areas. When a bad sector is encountered, WinHex can skip a number of subsequent sectors you specify. This is useful if you wish to accelerate the cloning process and if you do not care about some actually readable sectors not making it to the clone.

Regular disk cloning is not an option if you want to duplicate a disk in a removable drive (e.g. a floppy disk) with only one removable drive present. The correct concept for this application is *disk imaging*, where the data is first stored in an image *file*. The image can then be copied to a different disk. The result is the same as disk cloning.

When you specify a file named “dev-null” as the destination, the data will only be read and not copied anywhere (and you will be warned of this). This is useful if you are interested in the report about bad sectors, but do not wish to actually clone or image a disk.

You may try “simultaneous I/O” if the destination is not the same physical medium as the source. Offers a chance to accelerate the cloning process by up to 30%.

Specialist license or higher: In conjunction with simultaneous I/O you may also have WinHex copy the sectors of a disk in *reverse* direction, *backwards* from the end of the source disk. Useful if the source disk has severe physical defects that for example cause a disk imaging program or

your entire computer to freeze or crash when reaching a certain sector. In such a case you can additionally create an image in reverse order, by reading sectors from the disk backwards one by one, or better, you can even automatically *complete* an existing incomplete unsegmented conventional ("forward") raw image from the rear end to get an image that is as complete as possible, filled from both ends, with ideally only a small zeroed gap in the middle that represents the unreadable damaged spot on the source hard disk. For that you simply select an incomplete raw image file that you already have as a destination file, and you will be asked whether you wish to complete it instead of overwrite. WinHex will do the rest, e.g. allocate the missing sectors in the image file (zeroed out) so that it has the complete size of the source disk and then fill the file backwards as much as possible. Be sure to create reverse images on NTFS volumes, not FAT32. The source start sector to specify for reverse imaging is the same as for conventional forward images, i.e. usually 0 when imaging a complete hard disk.

For disk imaging in general it is recommended to use the File | Create Disk Image functionality instead, for various reasons (with a forensic license: support for .e01 evidence files, compression, splitting, hashing, encryption, metadata, technical details report, more convenient). Only in specific cases, for example when dealing with several physical disk defects or when the goal is to copy only certain ranges of sectors, advanced users can use Tools | Disk Tools | Clone Disk to have more detailed control over which sectors are copied from where to where in which order.

10.7 Images and Backups

This command “Create Disk Image”/“Make Backup Copy” in the File menu allows to create a backup or image of the currently open logical drive, physical disk, or individual file. There are three possible output file formats, each with unique advantages.

File format:	WinHex Backup	**Evidence File**	**Raw Image**
Filename extension:	.whx	.e01	e.g. .dd
Interpretable as disk:	no	yes	yes
Splittable:	yes	yes	yes
Compressible:	yes	yes	no
Encryptable:	no	yes	no
Optional hash:	integrated	integrated	separate
Optional description:	integrated	integrated	separate
Range of sectors only:	yes	(yes)	(yes)
Applicable to files:	yes	no	no
Automated maintenance:	Backup Manager	no	no
Compatibility:	no	(yes)	yes
Required license:	none	forensic	personal

The major advantage of evidence files and raw images is that they can be interpreted by WinHex like the original disks (with the command in the Specialist menu). This also makes them suitable for usage as evidence objects in your cases. This holds true for evidence files in particular because they can store an optional description and an integrated hash for later automated verification. Raw images have the benefit that they can be easily exchanged between even more forensic tools. All output file formats support splitting into segments of a user-defined size. A segment size of 650 or 700 MB e.g. is suitable for archiving on CD-R. Evidence files must be split at 2047 MB at most to make them compatible with X-Ways Forensics versions before v14.9

and EnCase versions before v6 and certain other tools. With a forensic license, raw image files and evidence files can automatically be verified immediately after creation, by recomputing the hash value that was originally computed from the medium, with the image instead.

Evidence file and WinHex backup compression is based on the “Deflate” compression algorithm that is part of the popular general-purpose library *zlib*. This algorithm consists of LZ77 compression and Huffman coding. With the “normal” compression level you can reach a compression ratio of 40-50% on average data. However, this comes at the cost of a considerably reduced imaging speed. “Fast/adaptive” compression is a *very good* and *intelligent* compromise between speed and good compression, not like the ordinary fast compression option in other programs. With “high” compression you gain only a few percentage points more compression, but at disproportional high cost. For WinHex backups, “adaptive” is the same as “normal”.

Raw image files can be compressed at the NTFS file system level, if they are created on NTFS volumes. Either normal NTFS compression is used, or the image file can be made “sparse”, such that large amounts of zero-value bytes won’t need drive space.

Cleansed images: With a forensic license, there is an acquisition option for those users who need to or want to exclude certain files from forensic images, called "Omit excluded files". The data stored in clusters that are associated with files that you exclude before starting the imaging process will automatically be zeroed out in the image. That won't have any effect on files whose contents are not stored in their own clusters. Before you start the imaging process for a partitioned disk, open the partitions in which the files are located that you would like to exclude. Wait till the volume snapshot has been taken if it was not taken before. Then exclude the files. You do not need to open and take volume snapshots of partitions whose data you would like to include completely. All other data is copied to the image normally. There is an option to "watermark" wiped sectors in the image with an ASCII or Unicode text string, so that when working with the image you are reminded of the omission when you look at the affected areas. Cleansed images are useful for anyone who needs to redact certain files in the file system, but otherwise wants to create an ordinary forensically sound sector-wise image, compatible with other tools. A must in countries whose legislation specially protects the most private personal data of individuals and certain data acquired from custodians of professional secrets (e.g. lawyers and physicians, whose profession swears them to secrecy/confidentiality). Limitation: Not available for disks partitioned as Windows dynamic disks or with Linux LVM*. Only files in supported file systems can be omitted. Note that you can also retroactively cleanse (redact) already created conventional raw images, in WinHex, by securely wiping files selected files via the directory browser context menu. The granularity of this operation is not limited to entire clusters that way. For example, that means it can also wipe files in NTFS file systems with so-called resident/inline storage and it does not erase file slack along. For a comparison of evidence file containers, skeleton images and cleansed images please see our web site. All of those are images that only transport a subset of the original data.

Another kind of cleansed image is an image in which all the clusters marked by the file system as free are zeroed out (specialist or forensic license only). That is very useful if you create the image for backup purposes and not for forensic purposes, or if for forensic purposes you do not require data in free space or are not supposed to acquire it (to only examine existing files). In conjunction with compression, this option has the potential to save a lot of drive space, depending on how much free space there is, and imaging speed can be greatly accelerated if there are large

contiguous free drive space areas in volumes/partitions. Note that in case of file system inconsistencies clusters could be erroneously regarded as free. If you wish to omit *both* certain (excluded) files *and* free clusters, also exclude the virtual file "Free space" and turn off "net free space computation" in the volume snapshot options.

You have to specifically confirm the creation of cleansed images as in the traditional sense they are not forensically sound (though in a more modern sense of the word they can be, depending on the jurisdiction that you work in in countries with stricter personal privacy rights and depending on the overall situation).

X-Ways Forensics checks for and warns of overlapping partitions when creating a cleansed image of a partitioned physical disk. Clusters in affected disk areas are not omitted. In such a situation, it is recommended to image the relevant partitions separately.

Forensic license: When creating an image, the technical details report is created and written to a text file that accompanies the image file. For an .e01 evidence file it is also incorporated directly into the .e01 file as a description. The SMART information is queried and written to the text file again upon completion of the image, so that you can see whether the status of a hard disk in bad shape has further deteriorated during imaging. Secondly, you can see how the “power on time” has changed, which is useful to deduce its unit of measurement (usually hours, but can be different on certain hard disk models). The text file also indicates the amount of time spent creating the image, the compression ratio achieved, the result of an immediate verification of the image based on the hash value (if selected), and any sector read errors.

Forensic license: Ability to create a second copy of an image immediately when imaging a disk, which is much quicker than copying the image file later and makes sense if the 2nd copy is created on a different drive. File spanning (i.e. when to start another image file segment) is kept in sync between both copies even when running out of space on one of the two target drives only.

Forensic license: You may specify an overflow location in advance where further image file segments will be stored should space on the primary output drive be exhausted. If you leave that field blank or if even the overflow location has no more space left, you will be prompted for a new path as before when needed. If an overflow location is specified in advance and at the same time you chose to create two copies of the image, then please note that the overflow location is used only for the first image copy that runs out of space, if any. For the other image copy you would be prompted if space is scarce.

Forensic license: Ability to compute two hash values simultaneously. If you make use of this option, then both hash values will be stored in the descriptive text file. The first hash value is the one that can be automatically verified when imaging completes. You could intentionally choose the faster algorithm for that as the main purpose at that point is to detect I/O errors and file errors. The second hash value is imported into the evidence object properties when adding the image to a case.

A special option allows to exhaust system memory prior to the hash verification to invalidate and thwart any file buffers employed by Windows so that the data of the image is read directly from the disk for the verification and not taken from the memory buffer. This option exists for small images and for somewhat paranoid or uber-diligent users. It is not required for images that are

much larger than the physical amount of RAM that is installed in your machine because by the time when the final parts of the image have been written, the initial parts are no longer in the buffer, and once the final parts are about to be verified they are no longer in the buffer because at that time the initial parts are in the buffer as they have been verified just before. Your system may behave a little bit sluggish for a while when using this option, and verification may be slightly slower than normally. This function is also available at any time via Options | Security. Under Windows 10 or newer it requires this setting in Windows: Local security policy | Local Policies | User Rights Assignments | Lock pages in memory. Add user to this policy. Restart to make it take effect.

Forensic license: Ability to schedule in advance subsequent disk imaging operations in additional instances that will wait until already ongoing imaging operations in previous instances have completed, to avoid inefficient simultaneous creation of multiple images on the same output disk (which is unnecessarily slow and produces highly fragmented image files). Additional instances only wait for previous instances in which the checkbox for waiting was checked as well, but not for others.

Forensic license: If you cancel disk imaging in the middle of the process, X-Ways Forensics quickly finalizes the .e01 evidence file format (more precisely, the current segment) to guarantee a consistent image even though it is not a complete image. Useful for example in an emergency situation when imaging media on site, because a incomplete image that can be used without errors is better than an unusable corrupt image. If hashing was enabled, incomplete .e01 images even have a hash value that can later be verified later.

Forensic license: The descriptive text file that is generated for images points out the exact sizes in bytes of all segments of raw images files and the exact chunk counts in all segments of .e01 evidence files. If for whatever reason one or more segments get lost or corrupted, this allows to create artificial placeholder segments of the right capacity to fill in any gaps (with the File | New command), such that all the data in subsequent segments will have the correct logical distance from the data in preceding segments, to preserve validity of pointers within the data (partition start sectors in the partition table, cluster numbers in file system data structures) as long as the original image file segments that contain source and destination are available. That the username of the logged-in user who creates an image is included in the descriptive text file is optional. Fully unchecked even the examiner name known in X-Ways Forensics is not included.

Forensic license: You can choose between the legacy compression algorithm for .e01 evidence files that is compatible with other tools and a much more modern compression algorithm, which offers a much better trade-off between compression ratio and compression speed plus decompression speed. Roughly speaking, depending on the data, with an almost as strong compression ratio as the "normal" setting of the compatible algorithm (a few % points less), the modern "normal" setting might require only 1/4 of the time for compression and 1/3 of the time for decompression. (We are referring to the mere computational work with a single thread here, excluding time needed for I/O.) When set to "stronger+", the modern algorithm might achieve a comparable compression ratio as the former "normal" (or slightly better), but require only 1/2 the time for compression and 40% of the time for decompression (or less). "Stronger++" takes noticeably more time and is usually not recommendable because the extra compression that it can achieve is usually limited, but it may still be faster than the old compression algorithm, especially for decompression (which NB typically occurs more than once, e.g. for immediate image

verification after creation, for image verification at a later date, a file header signature search, one or more more keyword searches, analysis and copying of files, etc.). Generally we recommend the modern "normal" setting. Please note that the modern compression style will render an image suitable for use in X-Ways Forensics and X-Ways Investigator 20.9 and later only. The "sparse" setting of the modern compression style, however, which is already extremely efficient when acquiring storage devices that have been minimally used, in fact 11 times (!) more space-saving for zeroed data than the sparse setting of the compatible compression style, is understood by v18.9 and later already.

Forensic license: You may adjust the compression level (not the selected algorithm) while .e01 evidence files are being created. Useful if your priorities (higher compression rate or higher speed) change, for example when you see that the remaining drive space suddenly seems scarce or you have to finish the process sooner than previously thought. Also useful to experiment, when not sure which compression option might be best for a particular system configuration (e.g. when imaging a live system on site and having to write the image to an external hard disk via USB, where I/O could be slow and the overall process may be faster with extremely successful compression than without).

Forensic license: For the .e01 evidence file format, you may choose the internal chunk size. Might be regarded as useful by some to achieve a marginally better compression ratio for ordinary data, at the expense of more time needed when creating the image and when later randomly accessing data in the image, but improves compression noticeably for extremely compressible data (e.g. a wiped or unused areas of a hard disk). A 512 KB chunk size reduces the image size with ideal data (e.g. only 0x00 bytes) ceteris paribus by an additional 40% compared to a 32 KB chunk size. Special optimizations are applied internally for chunk sizes of 32, 128, 512 and 1024 KB.

Forensic license: When imaging with active compression in .e01 format, X-Ways Forensics provides immediate visual feedback about the data density detected on the storage device. That is possible because disk areas that were never written as well as disk areas that were wiped achieve extremely high compression ratios, and data that is already compressed or encrypted will achieve zero additional compression, and "normal" data is somewhere in between. Taller red bars represent higher data density = more storage space requirement for the image = less compression achieved = more data to analyze = (if the bars reach the ceiling) potentially encryption. With a single mouse click anyhwere in the window you can toggle between data density and the reverse, compression ratio. Taller blue bars indicate higher compression = lower data density = no encryption = less storage space requirement for the image = less data to analyze = less work. Please note that the lengths of the bars may vary depending on the selected compression method/strength. The statistics are stored in the .e01 evidence file, so that the same chart is also available at any later time from the evidence object properties dialog when you click the button that is labeled "Data Density" or "Compression", if the evidence object is open.

Forensic license: Ability to specify how many extra threads to use for compression when creating .e01 evidence files. By default X-Ways Forensics will use no more than 4 or 8, and it depends on how many processor cores your system has, but you could try to increase the number on very powerful systems with even more cores usually without problems, for a chance to further increase the speed, or you can reduce it you run into stability problems.

Forensic license: You have the option to change the nature of an image (disk or volume) and its sector size when creating the image. This is possible not only for .e01 evidence files, where both is explicitly defined in the internal metadata (compatible with other tools), but also for raw images (via external metadata, compatible only with X-Ways Forensics/Image v18.4 and later, lost if the image leaves the realm of NTFS file systems). Useful whenever the source of the data is not an ideal interpretation. For example if a reconstructed RAID actually represents a volume, not a physical disk, then you can already adjust the nature of the image accordingly when you create it. Or if the sector size of the reconstructed RAID or a disk in an enclosure does not match the sector size of the file system in a partition, you can adjust the sector size of the image accordingly. All of this will allow for smoother and more successful usage of the image later, in particular by users who do not pay much attention to details such as image type and sector size. With the additional metadata present for a raw image, X-Ways Forensics does not need to prompt users for the nature of the image and its sector size even if under normal circumstances it would (for example because the image does not start with an easily identifiable partitioning method or volume boot sector).

Technically minded users may want to set the desired attributes of newly created image files, such as "read-only" or "encrypted", as well as buffering flags for performance tweaking in unusual environments such as "write through". Attributes are defined most thoroughly at https://docs.microsoft.com/en-us/windows/win32/fileio/file-attribute-constants, flags at https://docs.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-createfilea. The flag for "no buffering" should not be used. Attributes and flags are combined by oring (or adding) them and have to be specified in hexadecimal notation.

At the end of the imaging process, the computer can be optionally either shut down or (if supported by your system) hibernated, to save power. If you select hibernation and Windows signals that hibernation fails, X-Ways Forensics will instead try to shut down the system.

There is an option to add newly created images to the case and start refining their volume snapshot(s) automatically without further user interaction if the source disk had not been added to the case yet and if a case is open at that time when you start imaging.

Using this command is the recommended way to create a disk image. In order to image an arbitrary range of sectors, you could select a sector range as a block and copy it to a file via Edit | Copy Block | Into New File, or use Tools | Disk Tools | Clone Disk. The latter is particularly useful to partially image hard disks with severe physical defects (not just ordinary bad sectors) and can even copy sectors in reverse order.

For imaging automation please see the chapter about command line parameters.

The encryption algorithm optionally used in .e01 evidence files is either 128-bit or 256-bit AES/Rijndael, in counter (CTR) mode. This allows for random read access within evidence files. The 128-bit implementation is newer and faster and supported only by X-Ways Forensics v16.4 and later. Encryption will render an .e01 evidence file incompatible with other tools. The encryption algorithm uses a 256-bit key that is digested with SHA-256 from the 512-bit concatenation of the SHA-256 of the password you specify and 256 bits of cryptographically sound random input (“salt”), which is stored in the header of the evidence file. For 128-bit AES the 256-bit key is reduced to 128 bit by xor-ing the first and second half. The 128-bit counter is

randomized and incremented per encryption block, as a little-endian integer in 256-bit AES, as a big-endian integer in 128-bit AES. The encryption block size of AES is 128 bits. An additional SHA-256 is stored in the header as well (optionally for 256-bit AES, see Security Options) and used later to determine whether a password, specified by the user for decryption, is correct or not. The SHA-256 algorithm is applied to a concatenation of the salt, hash *x*, and hash *y* to compute this password verification hash, where hash *x* is the SHA-256 of the user-supplied password and hash *y* is the SHA-256 of the concatenation of the user-supplied password and hash *x*. For 128-bit AES, *y* becomes *x* and is concatenated and hashed over and over again, 100,000 times, to practically render rainbow table attack computationally infeasible. Please note that when you use compression and encryption at the same time, each chunk in an .e01 evidence file is first compressed, then encrypted. So an educated guess about the *nature* of the data in a given chunk might be possible, merely judging from the compressed size of the chunk (i.e. its compression ratio), even if the compressed data is encrypted.

If you have WinHex assign a filename for a WinHex backup automatically, the file will be created in the folder for backups (cf. General Options), named with the next free "slot" according to the Backup Manager's naming conventions ("*xxx*.whx"), and will be available in the Backup Manager. If you explicitly specify a path and a filename, you can restore the backup or image later using the Restore Backup command, and in case of split backups WinHex will automatically append the segment number to the filenames.

10.8 Dummy Image Segments

With the File | New command, in X-Ways Forensics you have the option to conveniently create dummy/makeshift segments for .e01 evidence files that can substitute missing/lost/corrupt original segments. The user has to specify the required chunk size and the number of chunks as well as a filename for the desired segment (must be with the correct extension, identifying the segment number, not number 1). The data written into the chunks can be recurring textual pattern/watermark ("MISSING IMAGE FILE SEGMENT!" when running X-Ways Forensics in English), so that you know that you are looking at a gap in between available data when browsing the interpreted combined image later. However, that is optional, for performance reasons. Zeroed out chunks are faster to generate.

The idea of such an artificial dummy segment is that if correctly created it can serve as a placeholder that ensures that data in subsequent segments has the correct logical distance from the data in preceding segments. Of course, the hash of the entire image cannot be successfully verified any more if the original data is not present, and of course, this functionality should be used only as a last resort if there is no backup of the missing segment file and if data recovery fails etc., and creation and usage of such a dummy image file segment should be properly documented.

When interpreting an .e01 evidence file that contains dummy segments, you will be notified, and the total number of placeholder chunks are noted in the evidence object properties when the image is added to the case.

If you require a placeholder for a single missing segment of which you don't know the chunk size

and chunk count because the image was created without a descriptive text file that contains this information, we can think of at least two ways how to find out:

1) Change the filename extension of the penultimate segment to that of the missing segment so that there is no gap. Then rename the last segment to the now missing penultimate segment. (If the missing segment actually is the penultimate one, the last step is sufficient; if the missing one is the last, no renaming is required at all.) Then add the image (first segment) to a case in X-Ways Forensics as usually. X-Ways Forensics will bring the misnamed segment to your attention in the Messages window, which can be ignored. Check the evidence object properties for the chunk size as well as the expected chunk count and the actually referenced chunk count. Subtract the actually referenced chunk count from the expected chunk count. Now you know how many chunks are missing. Change the filename extensions back to what they were before, and then create the missing dummy segment with the correct chunk size, correct chunk count, and correct extension.

With a variation, this approach also works if multiple consecutive segments are missing, just you rename more available segments to fill the gap in the first step, and you create as many dummy segments as necessary to fill the gap. Which dummy segment exactly contains how many surrogate chunks is not important as long as the total number of surrogate chunks must account exactly for the total number of missing chunks.

or (less complicated)

2) You make a note of the chunk size already when adding the image with the missing segment to the case. Then you quickly create a very small tentative dummy segment based on that chunk size with an arbitrary small number of chunks, e.g. 1000. Then you add the image to the case again. You will be informed by the application of how many chunks are missing (let this be *x*), and then you create the final dummy segment with 1000+*x* chunks.

If multiple discontiguous segments are missing, neither of these methods work; suitable dummy segments can only be created with the details from the descriptive text file as generated by X-Ways Forensics and X-Ways Imager.

10.9 Hints on Disk Cloning, Imaging, Image Restoration

Cloning or imaging with WinHex/X-Ways Forensics makes exact sector-wise, forensically sound copies, including all unused space and slack space. An image is usually preferable to a clone, as all data (and metadata such as timestamps) in an image file is protected from the operating system.

If you clone/image a disk for backup purposes, try to avoid that the disk is being written to by the operating system or other programs during the process, e.g. by unmounting partitions that are mounted as drive letters before starting. Such write operations are unavoidable, of course, if you clone/image the disk that contains the active Windows installation from where you execute WinHex/X-Ways Forensics. If the source disk is being written to during the process, the clone/image may have an inconsistent state from the point of view of the operating system (e.g. it

may not be able to boot a Windows installation any more). From a forensic standpoint, however, when cloning/imaging a live system, although it is highly desirable that no writing occurs any more, that should not be a major problem, as you still get an accurate snapshot of each and every sector.

If the destination of cloning or image restoration is a partition that is mounted as a drive letter, WinHex will try to clear all of Windows' internal buffers of that destination partition. If nonetheless you don't see the new contents in Windows Explorer on the destination after the operation has complete, you may simply need to reboot your system.

Note that WinHex does not dynamically change partition sizes and adapt partitions to destination disks larger or smaller than the source.

10.10 Skeleton Images

Forensic license only. A typical X-Ways feature that cements X-Ways Forensics' position as the tool that gives its users the greatest amount of control when selecting/targeting/filtering data at any conceivable level: The ability to create *forensic physical skeleton disk images* that contain only those sectors that are needed for certain purposes, while maintaining compatibility with other tools. These can be sectors with partition tables, file system data structures, their neighboring sectors as well as sectors with file contents or any sectors in unpartitioned no man's land. A skeleton image is typically sparsely populated with data, with vast areas in between remaining undefined, so that it makes sense to utilize NTFS sparse file technology for it. Unwritten areas in a sparse skeleton image will act as if zeroed out when read later. If not sparse, these areas are actually zeroed out.

You start skeleton imaging by invoking the File | Create Skeleton Image menu command. Which sectors from then now will be copied into the image is defined indirectly, by making X-Ways Forensics *read* those sectors from the source disk that are needed for a certain purpose. When the target image is open in the background, next you typically open the disk or partition or open and interpret the image that you wish to acquire partially. That way it will be automatically defined as the source, and that way even read operations during the important opening or interpretation step are triggered already, when partition tables and boot sectors have to be parsed, so that these essential data structures that define partitions and identify file systems are included in the skeleton image.

So after opening a partitioned physical disk, you have a "basic skeleton" in your target image: Partition tables pointing to partition boot sectors or nested partition tables, whose function is to support all the other data in between (file system data and user data). If you also wish to ensure that from the skeleton image it is possible to take a volume snapshot of a certain partition, i.e. get a listing of all files and directories referenced by the file system in that partition, then you open that partition from the source hard disk so that a volume snapshot is actually taken. Again, all the sectors read from the source hard disk in the process are simultaneously copied to the image, and that is the file system data structures, e.g. $MFT in NTFS, all directory clusters in FAT, and the catalog file in HFS+. That adds considerably more administrative data and also metadata to your skeleton image, but still no or almost no user contents. Unrelated sectors that are not used by the

file system are not read and therefore not copied. That also means that the ability to find previously existing files in the skeleton image will be limited. Note that you do not absolutely have to take a volume snapshot while filling the skeleton image and thus to transfer all essential file system data structures into the skeleton image. That could easily include a hundred thousand names of files and directories names, which may or may not be necessary or appropriate for your purpose. If you just need the contents and some metadata of certain files in an NTFS file system for example, you can specifically include the FILE records and contents of those files, without the entire $MFT, and thanks to the inclusion of sector 0 (the boot sector) X-Ways Forensics will know what the file system and the cluster size were, and can find the FILE records with a particular thorough file system data structure search in the skeleton image (quickly, thanks to the sparse nature of the image) and will therefore know the storage locations and names and timestamps etc. of those files in the volume.

File contents with so-called resident storage in $MFT in NTFS can optionally be excluded from the acquisition (wiped when the $MFT file is transferred) if you take a volume snapshot while filling a skeleton image. This may seem like a natural choice since ordinary (larger) files are by default not included in the target image either unless you specifically include them. However, this involves redacting data within certain sectors and as such alters the hash value of the affected sector range in the target image compared to the source volume. As a compromise, if hashing is active, a second hash value for the redacted data is included in the .log file, and that second hash value is the one that is re-computed when you have X-Ways Forensics verify the integrity of a skeleton image created with this option. Resident main file contents and resident alternative data streams that share the same FILE record as storage space are excluded or included together.

If you wish to include an arbitrary range of sectors in the image, you only need to find a way to make X-Ways Forensics read those sectors. For example, to include sectors from number 1,000,000 to 1,000,999, define those 1,000 sectors as a block and hash that block (in Disk mode) using the Tools | Compute Hash command, or run a physical search in that block only. Or, to acquire an unusually large partition gap between partition 1 and 2, you could hash the virtual file representing that gap. You can also manually navigate to any single sector of interest that you want to be included (e.g. Navigation | Go To Sector) or use any of the file system navigation menu commands. All of that works because reading sectors triggers their acquisition.

However, if you wish to specifically acquire selected *files*, that is easier, and it might be a good idea to turn off the indirect acquisition of any sectors that are read for whatever purpose along the way, so that for a example file that you preview and that turns out to be irrelevant is not acquired by the preview action already. For that, you can change the state of the skeleton image that is open in the background to "idle", using the State command in the File menu. In "idle" mode, only the "Add to [name of the skeleton image]" command in the directory browser context menu allows to acquire selected files (by temporarily activating the image and triggering read operations).

If you wish to include some operating system files, for example, such as Windows registry hives, explore the partition recursively from the root directory, filter for those files and invoke the "Add to" command in the directory browser context menu. (Only available if no evidence file container is open in the background for filling at that time.) The examiner who only has the resulting skeleton image will consequently be able to view the hives and create a registry report about them, assuming you had already copied the file system data structures which are required to find

out *which* sectors contain the data of the file.

The dialog window to change the state of the target image also allows you to close it, i.e. stop the acquisition for the moment or finalize the image. The same skeleton image can be further completed at any later time by selecting it again with the "Create Skeleton Image" command, but then you choose to not overwrite, but to update it.

As you see, you have full control over what data will make it into the image. The methology just assumes that you have some understanding of what data you want/need and, should that data not be stored in ordinary easy-to-select files, where to find it/how to get it physically. The sectors can be targeted in any order. Multiple reads of the same sectors don't change anything in the skeleton image and have no negative effect, except they may cause unnecessary duplicate lines in the optional log file that X-Ways Forensics can produce. Such a log file is created in the same directory as the skeleton image and will list all sector ranges that were copied, optionally along with the hash value of each sector range, which allows to manually verify the data in certain areas should there ever be doubt about it. If you use the "Add to" command to copy files to a skeleton image, the name of each such file will also be output in the log, followed by the sector ranges that correspond to to it (more than one if the file is fragmented or if X-Ways Forensics simply chooses to copy sectors in multiple chunks).

You may want to convert the resulting raw skeleton image into a compressed and/or encrypted .e01 evidence file and hash it or compress it with WinRAR or 7-Zip etc. before passing it on to other users. The compression rate will be unusually high if the skeleton image is only sparsely populated, and the speed of reading extremely high because undefined/unallocated areas do not have to be read from the disk. For your own use, you can just keep it as is since it does not use as much drive space as the nominal file size suggests thanks to NTFS sparse storage. If you wish to copy the raw skeleton image, be sure to copy it as a sparse file (can be done in X-Ways Forensics using the Tools | File Tools | Copy Sparse command) so that the copy will also be a sparse file and only takes as much drive space as the original file. A conventional copy command would copy even the vast unused and unallocated areas within the sparse file as binary zeroes.

To verify that the data transferred to a skeleton image has not changed, such an image can be hashed entirely, just like an ordinary image. Alternatively, and much quicker, you can use the command "Verify Skeleton Image" to hash only those sector ranges again that were actually transferred, according to the .log file (reading from the skeleton image), and compare the hash values to those in the .log file. Then, to verify that the .log file has not changed, it will be hashed itself, and the resulting highly valuable all encompassing master hash value is compared to the hash value stored in the optional .log.log file, if that file was created. It might be desirable to additionally verify that all unused areas in a skeleton image are still unallocated or at least filled with binary zeroes. This is not done by this function.

Options:

- A skeleton image should be created as an NTFS sparse file unless you intend to copy more than half of the sectors perhaps (just a very rough rule of thumb).
- If you don't have X-Ways Forensics set the nominal (logical) image file size to the full size of the source disk, then when interpreting the skeleton image and reading from it, a smaller "capacity" will be reported and you may get sector read errors. Still worth

thinking about it for example if you wish to capture merely the first 1 MB of a 1 TB hard disk. Saves a lot of time if you wish to convert the skeleton image to an .e01 evidence file or want to hash it in its entirety.

- Skipping already zeroed out source sectors (sectors of the source disk that only contain binary zeroes) will treat such sectors exactly like sectors that were not acquired. This makes the resulting skeleton image smaller ("more sparse"), but it prevent you from showing with just the skeleton image that these sectors only contained zeroes on the source disk. They are indistinguishable from sectors that were not acquired.
- "Include directory data structures of the file system" has an effect when you apply the "Add to" command of the directory browser context menu to selected directories. If this option is selected, you will also copy the data structures of the file system for these directories, if there are any, e.g. INDX buffers in NTFS, subdirectory clusters in FAT, etc. (nothing in HFS+), otherwise only the contents of the files *in* these directories.
- "Labels" will create a label for every file that you specifically add to the skeleton image in the source volume snapshot, so that it is easy to see which files were copied already in case of any doubt.
- If "Create log file" is at least half checked, a .log file will be created that references all copied sector ranges. X-Ways Forensics makes an effort to prevent acquiring duplicate sectors, e.g. when copying the exact same sector range a second time or when copying overlapping sector ranges, so that can explain why you may not get more lines in the .log file when copying the same sectors again. If the checkbox is fully checked, a .log.log file about the .log file will be created with a hash of the .log file.
- All copied sector ranges can be optionally hashed, and the hash values can be written to the .log file and can be verified after closing the skeleton image.

Benefits of skeleton images:

- Partial image, saves drive space.
- Quick to create, especially when acquiring remote hard disks through a slow network connection using F-Response.
- Transports/reveals only specifically targeted data, excludes unrelated data, as may be required by law, common sense, time pressure or the customer.
- Ideally suitable for technical data structures (partition tables, file systems) and files in a file system as well.
- Ability to acquire all essential file system data without knowing anything about the file system and in which sectors its data structures are stored.
- Result works exactly like a conventional raw image of the disk for all the intended purposes if adequately prepared, with original offsets and relative distances between data structures preserved (unlike in an evidence file container).
- The file format is universal, and all forensic tools that support raw images have a chance to understand the data, unless they need more data than was included or already don't understand the partitioning method or file system etc. of the original complete disk/image.

Caveats:

- Note that a search hit list on the screen with context previews around the search hits for example will cause a lot of read activity, so you may want to change the state of the skeleton image to idle mode when it is open in the background in certain situations.
- To avoid that the start sectors of files or directories that you merely click in the directory

browser in Partition/Volume mode are copied to the skeleton image (because such a click automatically jumps to the respective 1st sector), you can navigate the directory browser in Legend mode instead, or have to change the status of the image to "idle".

- Reading data from most *extracted* files such as e-mail messages, attachments, video stills, pictures embedded in MS Excel spreadsheets etc. do not trigger corresponding read operations at the disk level, so they cannot be copied. Skeleton images are suitable only for files at the file system level, not at any other level seen in volume snapshots. Use evidence file containers instead for such purposes.
- Note that to an unsuspecting examiner a skeleton image may look very much like an ordinary complete image. Such an examiner must be made aware of the incomplete, sparsely populated nature of the image. Unlike in a logical evidence file container, files whose contents are not contained in the image are not specially marked as such in a volume snapshot taken of an incomplete physical image. X-Ways Forensics v17.1 and later informs the examiner of the nature of an image when it's added to a case, if it detects a skeleton image.

A comparison of evidence file containers and skeleton images can be found on the web site.

Snippet imaging

A variant of skeleton imaging is called "snippet imaging". Click the button labelled "Snippet imaging" in the file selection dialog of the File | Create Skeleton Image menu command to start snippet imaging. Any sectors that are being read by X-Ways Forensics from any disk or image while snippet imaging is active are written into separate files named after the sector number, with a .sector extension, in a subdirectory of the default directory for images named after the disk or volume. Contiguous sector reads are copied to a single file.

Snippet imaging mode can be deactivated by invoking the File | Snippet Imaging menu command. Snippet imaging is helpful in specific situations only, for example for debugging purposes, when in need for very specific sectors only that are best located by the software automatically (e.g. data structures needed when opening a particular file). Compared to skeleton imaging, snippet imaging can be beneficial because no image file of the same size as the source disk is created. (Even if it's a nominal size only and the image is sparse, sparse does not help if the file needs to be sent via Internet or copied to a file system that does not preserve the sparse nature of the file.)

Because of their compatible names, snippet image files can be directly used for sector superimposition. They can also conveniently and because of their typically small size very, very quickly be restored to a other disks, all such files in the same directory at the same time, of course taking the sector numbers in the filenames into account, by clicking the button "Snippet imaging" in the File | Restore Image dialog window.

10.11 Backup Manager

Displays a list of previously created WinHex backups. The items can be listed in a chronological

or alphabetical order. Choose the backup you would like to restore. When that function completes, the original file or sector contents is shown.

You can restore the backup
- into a temporary file first such that you will still need to save it,
- directly and immediately to the disk, or
- to a new file.

In the case of disk sectors you may also wish to specify a different destination disk or a different destination sector number. It is also possible to only extract a subset of the sectors from the backup. (However, sectors at the beginning of a *compressed* backup cannot be left out during restoration.) If the backup was saved with a checksum and/or a digest, data authenticity is verified before the sectors will be directly written to the disk.

The backup manager also allows to delete backups which you do not need any longer. Backups that were created for internal use by the Undo command can be deleted by WinHex automatically (cf. Undo Options).

Backup files that are maintained by the backup manager are located in the folder specified in the General Options dialog. Their filenames are "*xxx*.whx" where *xxx* is a unique three-digit identification number. This number is displayed in the last column of the backup manager list.

10.12 Recover/Copy Command

Allows to copy the selected files from their current location to a location available for a standard Windows file dialog, e.g. out of an interpreted image file or from a local disk. This can be applied to both existing and deleted files and directories. Illegal filename characters are filtered out.

If necessary, you can manually enter the output path by clicking the "..." button in the same line where the path is displayed. Useful if you wish to specify a network location that Windows does not list by default in the dialog window for the path selection. If you enter a non-existing output path, you will be notified and may proceed anyway, in which case that path will be created automatically if possible. The unlabeled check box next to the "..." button can be used to indicate that you would like to get a Windows Explorer window opened for the output path once copying has completed to check out the result.

Numerous extra features are available in X-Ways Forensics:

- The complete original path can optionally be recreated in the output directory, or optionally (if half checked) only a partial path. The evidence object name becomes part of the recreated path, too, if you either copy from within the case root or if you do not have X-Ways Forensics default to the evidence object folder as the output directory (see case properties). A partial path is the path starting from the currently explored directory, or when copying from the recursively explored case root window only the evidence object name, not the path within the evidence object.
- Overlong paths are supported for output path + optional original path + original filename, of

more than 259 characters, up to 511 in WinHex Lab Edition and 1023 characters in X-Ways Forensics. You can still limit paths to the ordinary length of 259 characters or less if you would not be able to access (e.g. view, copy or delete) such files otherwise (because ordinary tools like the Windows Explorer in Windows 7 do not allow that). If the output path of a selected file exceeds the limit, the name of the is shortened until it fits. If shortening the name does not help to stay under the specified path length limit, the file is not copied, but labeled, so that you can conveniently select all the omitted files later and copy them separately without original path if you like.

- It is possible to create a 2nd copy of all selected files in a separate directory. Useful if you need to provide two parties with copies of relevant files and wish to save time. The logging option is for the 1st copy only, though.
- Output files can be named optionally after any other column in the directory browser, such as unique ID, hash value, ID, comment, offset in the file system etc. etc. Such metadata information can also be prepended or appended to the name, which for example could be useful in conjunction with alternative name, existence status, labels, timestamps, author, sender, description, attributes, analysis result, hash set, ... If the cell text consists of multiple lines (e.g. comments or metadata column), only the first line is used. Backslashes in the path columns are automatically replaced with underscores. That allows to name a file after its complete original path.
- Files that could not be copied (e.g. if path too long) get a label.
- The presumed correct file type of newly identified files, if different from the extension in the original filename or if the filename does not have any extension, can optionally be appended to the output filename. This option also has an effect when copying files to view them with the associated program.
- Unless you choose to overwrite or skip files with identical names that exist in the output directory, duplicate filenames will be changed to unique filenames by inserting incrementing numbers before the extension. So if you copy all files to the same directory, even those from different evidence object, all output filenames will be unique (and the copylog file allows you to later find out which file was originally named how and originated from where and which metadata it had).
- If "Apply original timestamps to copies" is half checked, the file system level creation, modification and last access timestamps, if available, are re-applied to the recovered/copied files, plus the internal content creation timestamp if available may substitute for a missing file system level creation timestamp. If the box is fully checked, that means X-Ways Forensics will make extra efforts to set creation, modification and last access to some original timestamps to avoid that any of these three standard timestamps will reflect the time when the Recover/Copy command was used. For example extracted e-mails or attachments or files in archives or carved files may not have all or any timestamps. X-Ways Forensics may resort to record change timestamps, alternative creation timestamps, content creation timestamps, and modification timestamps as substitutes for creation, modification as well as last access. If you check an extra box, the output files may even inherit creation timestamps of parent files and directories. If half checked, only timestamps of parent files are inherited (think of e-mails that contain e-mail attachments or pictures that contain thumbnails). If fully checked, timestamps can also be inherited from parent directories (or grandparent directories or great-grandparent directories etc.). An extreme example is a carved files with no timestamps at all. Its parent directories are virtual directories and have no original timestamps either. Hence the creation timestamp of the root directory will be adopted, if available (not in FAT file systems). A

parent directory creation timestamp could be regarded as a lower chronological limit for the unknown creation timestamp of a file. A parent file creation timestamp could be regarded as an upper limit for the unknown creation timestamp of a file if the parent is a file archive or an e-mail message. If the file is a thumbnail embedded in a JPEG file, the creation timestamp of the parent should be exactly right for the child object.

- When working with an active case and if special logging for this command is enabled, the copy/recovery process is documented in the file “copylog.html” or “copylog.txt”. All available metadata and the output filename (optionally including target path) can be recorded. The file can be created either in the _log subdirectory of the case or in the Recover/Copy target folder. Cf. also Case Properties.
- Slack space can optionally be included in the output, either as part of the file or separately, or *solely* slack can be copied.
- You can choose whether to also copy child objects of selected files or not.
- You can also choose whether to copy files that are filtered out.
- If you have X-Ways Forensics recreate the original path for copied files, the hierarchical location of files that are child objects of other files must be reflected appropriately, too. And that must happen with the help of a directory, because ordinary file systems do not support the concept that a file can contain further files, as is normal with volume snapshots in X-Ways Forensics. However, there would be a name conflict if an artificial directory was created with the same name as the parent file, as that parent file might be selected for copying as well, and would of course be created in the same directory as the aforementioned artificial directory that is needed to reflect the path of the child object. Hence the artificial directory must be named slightly differently. It can be truncated after a user-defined number of characters, and this is useful in particular for e-mail messages that are named after the subject line and of course can contain attachments as child objects, to avoid overlong paths. Also either a single suffix character of your choice is appended (and by default that is a special Unicode character that is invisible in complete Unicode fonts, such that the directory seems to have exactly the same name as the corresponding parent file), or otherwise some descriptive words like " child objects" are appended to the name (but that unfortunately increases the total path length, which all too often exceeds common limits). If the edit box for the suffix character seems to be blank, that is most likely because the aforementioned invisible Unicode character is in there. It has a width of 0. To replace it with any other character, remove the invisible character first, by clicking in the edit box and hitting the backspace key on your keyboard.
- Files can be grouped/classified in separate output directories based on up to two selected directory browser column, e.g. existence status (in order to easily distinguish between originally existing and deleted files), description, evidence object, file type, file type description, file type category, sender, owner, hash set, hash category, labels, search terms. It is also possible to limit the grouping directory name to a certain number of characters. That could be very useful for example in order to group files by year (the first four characters in creation or modification timestamps, given suitable notation settings) or to simply to split up a huge number of output files into roughly equally large subdirectories (with the first one or two characters of the hash value, for 16 or 256 such subdirectories), based on the law of large numbers, or simply to reduce the risk of overlong paths.
- Files of certain supported types can be converted to PDF format, to share with computer users who otherwise would not have suitable applications to view the files. You can define the file types that do not need to be converted, e.g. those that can easily be displayed by a web

browser or with Windows tools. If no conversion is possible, the original file is copied unconverted. There is also an pption to turn all selected files into a single PDF document. This includes even file types that would usually not be converted to PDF individually.

- • Pure text can be extracted from files of certain supported types and output as plain text files. That is the same representation that you get when switching from ordinary Preview mode to text Preview mode, and the same text that a logical search would see of a file when you have X-Ways Forensics "decode" it. Files that are not suitable for text extraction (e.g. picture files) or from which no text can be extracted for whatever other reason are copied normally (with their original contents) if the corresponding checkbox is only half checked, or are omitted if fully checked (which means the output is 100% pure text).
- If both an attachment and the corresponding e-mail message (its parent) are selected for copying and not excluded by filters, the attachment can optionally be embedded in the resulting output .eml file as Base64 code instead of copied separately. That facilitates viewing the complete e-mail including attachments. To view .eml files you can use Outlook Express, Windows Mail, Windows Live Mail or Thunderbird (all free of charge). If certain attachments cannot be embedded, you will be informed via the Messages window, and in such a case they will be copied separately, as if the embedding option was not selected.
- NTFS alternative data streams (ADS) can optionally be output as ADS. By default, they are recreated as ordinary files, to make them more easily accessible.
- X-Ways Forensics can try to encode zeroed out areas in a file as sparse when writing the data. This will have an effect only if the zeroed areas are somewhat aligned and sufficiently large, and of course only when writing to an NTFS or ReFS volume, not FAT. Works no matter whether the source file is defined as sparse or not. This option will reduce the data transfer rate and is only recommendable if you know that the data that you are copying is probably suitable.
- You may use the alternative names of files, if available, for the output. The alternative name, if one exists, can be seen in the directory browser in square brackets. For example, when parsing iPhone backups, X-Ways Forensics automatically changes artificial generic filenames back to what they were originally. Or, when parsing $I files from the Windows recycle bin, the corresponding $R files are given their original names. If for some reason you prefer the untranslated filenames when copying such files off the image to your own hard disk, for example because you wish to process these files with some external tool that expects the artificial filenames, then you can use this option.

When using the Recover/Copy command in search hit lists, directories that contain hits are recreated in the output folder as files, as the user likely wishes to retain the original data that contain the actual search hit. Child objects are never copied along with their parent objects from within a search hit list.

10.13 Duplicate File Detection

If you wish to review duplicate files only once and file system level metadata like timestamps and deletion status are of secondary importance, then you can use the command "Find duplicates in list" in the directory browser context menu to identify duplicate files. All the currently listed files (listed, not selected!) are checked. They are sorted to find identical properties. If so desired, the duplicates can be automatically excluded in the volume snapshot. Only one file in each group

of identical files will not be excluded. Each group of identical files can optionally get assigned a unique label name, which makes it easy to use a filter to see all the members of a given group, even if they are contained in different evidence objects.

When in doubt which duplicate to exclude, this function chooses to keep existing (not deleted) files, and among deleted files rather discards carved files and keeps files found via file system data structures. And when in doubt, it prefers to keep the copy of a file whose owner is known. Optional special rules: Identical e-mail messages with different attachments (child objects) will be marked as duplicates, but not excluded. Identical attachments (child objects) will be marked as duplicates, but they will be excluded only indirectly if they are part of identical e-mail messages and those are excluded, too. This facilitates the examination and also avoids a situation where the parent (e-mail message) of one e-mail+attachment family and the child object (attachment) of another family is excluded.

If later you find relevant files for which there were duplicates and you are interested in the duplicates, too (wish to see their their filenames, paths, or timestamps etc.), you could for example create a hash set of that files to conveniently and automatically identify all the duplicates, by matching the hash values of all files against that particular hash set and using the hash set filter, or you could use the Hash column filter directly.

Pairs of duplicates in the same volume snapshot can be optionally linked as so-called related items, so that it's easy to navigate from one such file to at least one duplicate. However, that does not work across evidence object boundaries. Marking the files as duplicates in the Description column is optional.

The most common and reliable criterion to identify files as duplicate is a regular hash value. However, computing hash values in large data sets costs a lot of time, so any reasonable deduplication option that does not require hash values is hopefully appreciated by some users. Alternative criteria are available. You could compare files simply based on identical names. This is a case-insensitive comparison and of course should be used only if you know what you are doing, as it does not compare the file contents at all. Could be useful for example if you wish to get rid of multiple copies of the same files found in backups if you do not need to keep different versions of these files. If prior to the comparison for example you sort by last modification date in descending order, this will ensure that the newest version of the file will be kept and all older versions will be excluded. Files with identical names are not marked as duplicates in the Attr. column. The number of characters to compare in filenames is user-definable.

Another useful criterion is the modification timestamp. The timestamps are compared as strings, and the more characters are checked, the more precise you require timestamps to be identical. With your notation settings in mind, you could choose to compare only the date, or date + times with precision of minutes, or seconds, milliseconds or anything in between, by defining the number of characters. An immediate example of how truncated timestamps look like based on the number of characters that you want to compare and based on the current notation settings is displayed instantly. Note that both the number of characters to compare limit the precision (intentionally or inadvertantly) as well as the number of decimals allowed by your notation settings. Limited precision may be desirable for example to recognize files as identical even if the modification times of file copies in NTFS and FAT differ by 1 second because of FAT timestamp rounding. The notation settings can be accessed right from within the deduplication options

dialog.

You can also add one or two additional criteria for the identification of duplicates: The full-precision modification timestamp and size, useful and quite reliable for example in conjunction with the filename. Another option for the main criterion is the structure type, which actually identifies groups of similar or related files. Combined with modification time and size this also is relatively reliable for the identification of duplicates, but only available for certain file types.

If you have access to PhotoDNA in X-Ways Forensics, you may also identify and exclude duplicate pictures using PhotoDNA. All duplicates can be marked as "duplicates found" in the Description column, and all except one will be excluded. When in doubt, deleted files or pictures with a poor resolution will be excluded and existing files and pictures with a higher resolution will be kept. Please note that the hash value comparison is a potentially time-consuming operation if many pictures are listed in the directory browser, much more so than for conventional hash values. However, you can abort the comparison at any time. This operation requires that PhotoDNA hash values have been computed beforehand, using Specialist | Refine Volume Snapshot | Picture processing | Compute PhotoDNA hash values. It is useful for example for law enforcement agencies that wish create PhotoDNA hash sets of unique pictures only and for that purpose maintain a lawful collection of incriminating pictures without duplicates. The strictness of the picture comparison is the same as set in the Specialist | Refine Volume Snapshot | Picture processing dialog window for matching against the PhotoDNA hash database.

10.14 Surrogate Patterns

If the program has trouble reading data for Disk/Partition/Volume or File/Preview mode or for searches, hashing, imaging, etc. etc., the question is data it should present to the requester. For read errors at different levels it uses different surrogate/substitute strings (preset texts), many of which by the way are language dependent. These strings are repeatedly copied into the read buffer until it is full, forming a recurring pattern that is easy to spot visually if shown on the screen and that should hopefuly easily catch the user's attention and make him or her immediately aware of the problem.

1) "UNABLE TO READ FILE" for example means that at least certain portions/segments/extents of a file cannot read because the file system does not define where to find them or because it does but that definition is invalid or because it does but X-Ways Forensics does not understand it.

Example: The file system defines that a file consists of 6 clusters starting at cluster 1000 in the volume and 4 clusters starting at cluster 55,555 in the volume.

One possible reason for "UNABLE TO READ FILE" in this example would be that the volume consists of 40,000 clusters only. The first 6 clusters of the file can be read, but the last 4 clusters of the file cannot be read, simply because there is no cluster 55,555 that could be read. If this concerns an existing file, it is some kind of file system corruption or volume inconsistency. Could happen if something went wrong when a volume was shrunk, or if it's a spanned volume covering multiple disks of which only the first segment is available treated as if it was the entire volume. Another possible reason for "UNABLE TO READ FILE" would be that X-Ways

Forensics was able to reconstruct a previously existing file partially only. The size may be known from $LogFile or a volume shadow copy, and the first few clusters of the file may be known from the source, but the whereabouts of the remaining clusters may be unknown. Another possible reason for "UNABLE TO READ FILE" if it's a compressed file in a file archive would be that the file archive is corrupt so that the contained compressed file cannot be read completely any more

If it's a file system problem, then you can find more more precisely what is going on by looking at the file system data structures that define the volume. Users can usually easily locate them in 2 seconds via a right click on the file, Navigation | Seek [name of the data structure].

2) "BADEVIDENCEFILE!" refers to a problem in an image in .e01 evidence file format. A possible reason to see that pattern would be that the requested sector is contained in the 2nd half of a compressed chunk (also called block) in which a few bits flipped so that only roughly the first half could be successfully decompressed.

3) "UNREADABLESECTOR" is a pattern that is defined in Options | General, which is always used instead of the original data stored in disk sectors if these sectors cannot be read, for all purposes (display on the screen, imaging, cloning, hashing, searching, ...). If you are going to hash disks with bad sectors and want to compare/reproduce the results with other tools, then you can specify the same pattern as used by the other tool here. Just note that such hash values are difficult to reproduce because bad sectors could multiply in the course of several attempts. If when trying to read bad sectors you prefer to get zero-value bytes delivered back, totally remove the pattern (ensure that the edit box is completely blank). If you keep the pattern, it will make it much easier to tell which sectors could be read and which sectors could not be, on the original hard disk directly, and that is also the case when you look at the same sectors in an image of that hard disk, provided that the pattern was active at the moment when the image was created with X-Ways Forensics. A bad sector on a hard disk is for example one whose internal CRC does not match the payload data in that sector any more.

4) Other surrogate patterns are "MISSING IMAGE FILE SEGMENT!", "PAST END OF IMG", and "UNREADABLE PAGE", all of which should be basically self-explanatory. ("Page" refers to a memory page.)

10.15 Reconstructing RAID Systems

WinHex and X-Ways Forensics can internally destripe RAID level 0, 5, 5EE and 6 systems as well as JBOD consisting of up to 16 components. The components may be physical hard disks or images of physical disks for hardware RAIDs, or partitions for Linux software RAIDs. Components that are available as images need to be opened and interpreted before you use this function. Components that are partitions need to be opened first before the RAID reconstruction can take place.

You need to select the components in the correct order. WinHex lets you specify the strip size in sectors (often 128 or at least a power of 2 like 32, 64, 256) and different RAID header sizes per component (often simply 0). The strip size multiplied by the number of RAID component disks

gives the so-called stripe size, i.e. a whole row.

The header is a reserved area at the start of a component disk that some RAID controllers set aside for their private data and thus must be excluded from the reconstruction. If there are a few reserved sectors at the end of a component disk, prior to reconstruction you could specify the number of actually used sectors plus header size for each component via Tools | Disk Tools | Set Disk Parameters as the "Sector count", or alternatively you can specify a non-zero footer size in sectors on components of a RAID that you reconstruct, to exclude sectors at the end. This could be useful in particular for JBODs if the interspersed unused space disturbs the consistency of the resulting reconstructed data.

You can usually tell that either the component order, the strip size, the stripe pattern, or the RAID header size was selected incorrectly when no partitions are detected or partitions with unknown file systems or with file systems that cannot be interpreted properly.

When you add a reconstructed RAID system to a case (and optionally partitions opened from such a RAID system), the selected RAID configuration parameters are saved with the evidence object, which allows to access the RAID system instantly in later sessions (in X-Ways Forensics only).

In RAID level 5 and 6, data is not only striped across all component disks in a rotating pattern, but also interspersed with parity blocks for redundancy. RAID level 5 and 6 are implemented in different ways by different RAID controller manufacturers in that they employ different stripe/parity patterns. The supported patterns are the following:

```
Level 5: Backward Parity aka Left Asynchronous (Adaptec)
Component 1:  1  3  P
Component 2:  2  P  5
Component 3:  P  4  6

Level 5: Backward Dynamic Parity aka Left Synchronous (AMI and
Linux standard)
Component 1:  1  5  9  P
Component 2:  2  6  P  10
Component 3:  3  P  7  11
Component 4:  P  4  8  12

Level 5: Backward Delayed Parity (HP/Compaq)
Component 1:  1  3  5  7   9   11  13  15
Component 2:  2  4  6  8   P   P   P   P
Component 3:  P  P  P  P   10  12  14  16

Level 5: Forward Parity (aka Right Asynchronous)
Component 1:  P  3  5
Component 2:  1  P  6
Component 3:  2  4  P

Level 5: Forward Dynamic Parity (aka Right Synchronous)
Component 1:  P  6  8  10
```

```
Component 2:  1  P  9  11
Component 3:  2  4  P  12
Component 4:  3  5  7  P

Level 5: Forward Delayed Parity
Level 5: Forward Dynamic Delayed Parity (CRU/Dataport)

Level 5EE: Backward Parity (Adaptec)
Component 1:  1  3  S  P
Component 2:  2  S  P  7
Component 3:  S  P  5  8
Component 4:  P  4  6  S  (S = spare)

Level 5EE: Forward Parity
Component 1:  1  P  S  7
Component 2:  2  3  P  S
Component 3:  S  4  5  P
Component 4:  P  S  6  8

Level 6: Backward Parity (Adaptec/JetStor)
Component 1:  1  3  P  Q
Component 2:  2  P  Q  7
Component 3:  P  Q  5  8
Component 4:  Q  4  6  P

Level 6: Backward Dynamic Parity
Component 1:  1  4  P  Q
Component 2:  2  P  Q  7
Component 3:  P  Q  5  8
Component 4:  Q  3  6  P

Level 6: Forward Delayed Parity
Level 6: Forward Parity
```

The parity start component can be defined differently if necessary, for many RAID variants. To stick with the select standard pattern, leave that value at 0. In order to define a non-standard parity start component, specify the number of the component where the parity is located first (1-based).

The delay with that the parity moves on HP/Compaq controllers is most often 4 or 16, but freely configurable.

If one of the RAID component disks is not available, you can reconstruct a RAID 5 system nonetheless because one component is redundant. Simply select a dummy substitute (one of the *other, available* components of the same RAID system) as the *missing* component and declare that component “missing”! RAID 5EE and RAID 6 can also be internally reconstructed if one (and only one) component is missing.

Support for software RAIDs

Linux MD RAID container partitions are automatically recognized as such. They are represented as two distinct items: A static header area that contains metadata about the RAID in general and the following component in particular, usually at relative offset 4096, and an explorable partition that serves as the RAID component. In case of RAID level 1 that explorable partition contains a fully self-contained volume whose file system can be parsed normally (without any reconstruction effort) if supported. In case of other RAID levels, the reconstruction can be accomplished with the Specialist | Reconstruct RAID command, and some hints on the correct reconstruction parameters are shown as comments attached to the header area item. Note that you need to open all the relevant partitions first so that they are offered for selection as the components of the RAID. The result of the reconstruction will be a single volume, which is represented as encompassed in a virtual physical disk. The RAID components have to remain in the case as evidence objects for internal reasons, to allow to re-open the reconstructed RAID with a single mouse-click later.

Windows storage pool container partitions are also automatically recognized as such, and it is possible to properly open partitions whose sectors size is a multiple of the sector size of the underlying physical disk. This is important for example for Windows storage space partitions in Windows storage space pool disks. These partitions and disks have a simulated sector size of 4 KB even if they reside on physical disks with a sector size of 512 bytes. The search for lost partitions can find NTFS storage space partitions within storage space container partitions despite sector size discrepancies, which is a useful work-around for simple single-disk storage spaces.

Windows drive letters are accepted as components to internally reconstruct RAIDs. That doesn't make much sense, but allows you to reinterpret a drive letter as a physical storage device in X-Ways Forensics if necessary, by selecting it as the sole component of a JBOD. This could be useful if for some reason you need to apply menu commands to it that only make sense to apply to physical storage devices and are only available for physical storage devices, such as Scan For Lost Partitions. For example a RAID that is reconstructed/mounted outside of X-Ways Forensics may somehow present itself as a drive letter (although it does not have a volume boot sector / file system starting at sector 0 and thus cannot be put to any good use in Windows itself).

10.16 NSRL RDSv3 Format

NSRL RDSv3 files cannot be imported directly into an X-Ways Forensics hash database. First, a universal hash set text file must be generated from the NSRL RDSv3 database, and then the generated text file imported.

Either download the current full NSRL RDSv3 (SQLite) database that you need, or else update your local full copy from a downloaded delta, following the instructions on the NSRL web site.

To generate a hash set suitable for import by X-Ways Forensics:

Install sqlite3 if not already installed, and then from the command line:

```
cd "NSRL DB folder"
```

replacing "NSRL DB folder" with the location that contains your current copy of the NSRL RDSv3 database. Then type:

```
sqlite3 RDS_2022.01.3_curated.db
.mode tabs
.header on
.once md5.txt
select md5 from metadata;
.once sha1.txt
select sha1 as 'sha-1' from metadata;
.q
```

The date and version stamp "2022.01.3" above may be different for your copy of the database. Amend as necessary.

The above commands will generate a file of MD5 hashes into the file md5.txt in the same folder as the NSRL RDSv3 database, and a file of SHA-1 hashes into the file sha1.txt. You can omit the two lines containing md5 or sha1 in the instructions above if you don't need a hash set for that hash type.

You can now import the generated text files into your X-Ways Forensics hash library following the import procedure described in the chapter about the hash database. There will be duplicate hash values, and the import will discard them; this is normal.

Appendix A: Template Definition

1 Header

The header of a template definition has the following format:

```
template "title"
[description "description"]
[applies_to (file/disk/RAM)]
[fixed_start offset]
[sector-aligned]
[requires offset "hex values"]
[big-endian]
[hexadecimal/octal]
[read-only]
[multiple [fixed overall size]]
// Put any general comments to the template here.
begin
      variable declarations
end
```

Tags in brackets are optional. The order of the tags is irrelevant. Expressions must only be enclosed in inverted commas if they contain space characters. Comments may appear anywhere in a template definition. Characters following a double slash are ignored by the parser.

The keyword `applies_to` must be followed by one and only one of the words `file`, `disk`, or `RAM`. WinHex issues a warning if you are going to use a template on data from a different source.

While by default templates start interpreting the data at the current cursor position when applied, an optional `fixed_start` statement ensures interpretation always starts at the specified absolute offset within the file or disk.

If the template applies to a disk, the keyword `sector-aligned` ensures the template interpretation starts at the beginning of the current sector, regardless of the exact cursor position.

Similar to the `applies_to` statement, the `requires` statement enables WinHex to prevent an erroneous application of a template definition to data that does not match. Specify an offset and a hex-value chain of an arbitrary length that identifies the data for which the template definition was intended. For example, a valid master boot record can be recognized by the hex values 55 AA at offset 0x1FE, an executable file by the hex values 4D 5A (“MZ”) at offset 0x0. There may be multiple `requires` statements in a template definition header, which are all considered.

The keyword `big-endian` causes all multi-byte integer and boolean variables in the template definition to be read and written in big-endian order (high-order byte first).

The keyword `hexadecimal` causes all integer variables in the template definition to be

displayed in hexadecimal notation.

The keyword `read-only` ensures that the template can only be used to examine, but not to manipulate data structures. The edit controls within the template will be grayed out.

If the keyword `multiple` is specified in the header, WinHex allows browsing to neighboring data records while displaying the template. This requires that WinHex has knowledge of the record's size. That size may optionally be specified as a parameter to the `multiple` statement. The keyword "multiple" may alternatively be used in the body. In the body it must be accompanied by a parameter that provides the amount of data covered by the template in bytes, and may be a formula and (unlike in the header) may use constants or other variables. This allows to navigate to adjacent records to the left and to the right. If that amount of data is not explicitly defined in the body, i.e. the `multiple` keyword is used only in the header without parameter, then the amount is deducted indirectly based on the current read position at the end of template interpretation, and may be considered variable in length if variables in the template have a variable size, in which case no navigation to the left (to preceding data records) is possible.

2 Body: Variable Declarations

The body of a template definition mainly consists of variable declarations, similar to those in programming languages. A declaration has the basic form

```
type "title"
```

where `type` can be one of the following:

- `int8, uint8 = byte, int16, uint16, int24, uint24, int32, uint32, uint48, int64,`
- `uint_flex,`
- `binary,`
- `float = single, real, double, longdouble = extended,`
- `char, char16, string, string16,`
- `zstring, zstring16,`
- `boole8 = boolean, boole16, boole32,`
- `hex,`
- `DOSDateTime, FileTime, OLEDateTime, SQLDateTime, UNIXDateTime = time_t, JavaDateTime,`
- `GUID`

`title` must only be enclosed in inverted commas if it contains space characters. `title` must not consist only of digits. WinHex does not distinguish between upper and lower case characters in titles. 41 characters are used to identify a variable at most.

`type` can be preceded by at most one member of each of the following modifier groups:

```
big-endian      little-endian
hexadecimal     decimal         octal
```

```
read-only       read-write      hidden
local
```

These modifiers only affect the immediately following variable. They are redundant if they appear in the header already. "hidden" identifies variables whose values you wish to set during parsing and may need for subsequent calculations, but do not want to show to the user. Also useful for constants that you define and use in calculations that the user should not be distracted with. "local" translates timestamps except DOSDateTime from UTC to the timezone specified in the General Options.

The number at the end of a type name denotes the size of each variable (strings: of each character) in bits. With `char16` and `string16`, WinHex supports Unicode characters and strings. However, Unicode characters other than the first 256 ANSI-equivalent characters are not supported. The maximum string size that can be edited using a template is 8192 bytes.

The types `string`, `string16`, and `hex` require an additional parameter that specifies the number of elements. This parameter may be a constant or a previously declared variable. If it is a constant, it may be specified in hexadecimal format, which is recognized if the number is preceded by `0x`.

You may declare arrays of variables by placing the array size in square brackets next to the type or the title. Specify "unlimited" as the array size to make the template stop only when the end of file is encountered. The following two lines declare a dynamically sized ASCII string, whose length depends on the preceding variable:

```
uint8      "len"
char[len]  "A string"
```

The same could be achieved by the following two declarations:

```
byte       "len"
string len "A string"
```

The character "~" can be used as placeholder for later replacement with the actual array element number (see below). This does not apply to arrays of `char` variables, since they are automatically translated into a string.

Numerical parameters of `string`, `string16`, and hex variables as well as array size expressions may be specified in mathematical notation. They will be processed by the integrated formula parser. Such expressions need to be enclosed in brackets. They must not contain space characters. They may make use of previously declared integer variables whose names do not contain space characters either. Supported operations are addition (+), subtraction (-), multiplication (*), integer division (/), modular division (%), bitwise AND (&), bitwise OR (|), and bitwise XOR (^). Valid mathematical expressions are for example (5*2+1) or (len1/(len2+4)). The result is always an integer and must be a positive number.

`zstring` and `zstring16` are null-terminated strings whose size is determined dynamically at run-time.

Constants

You can also define integer constants in templates, with a name, an integer type and a value of your choice, and use those constants in calculations. If the name contains a space, it needs to be enclosed in quotation marks. The value can be a formula itself and depend on other constants or variables that are already defined at the time when the constant is defined. Constants will be listed in the template window along with variables. Example:

const int16 100 "My constant"
const int16 ("My constant"*10) "My other constant"
goto ("My other constant"*5)

This will change the current position of template interpretation to offset 5000.

There are internal predefined constants named Bytes_per_sector and Bytes_per_cluster that you can use in formulas, in templates that are applied to storage devices or interpreted images or partitions/volumes. Another predefined constant is Bytes_per_record. It depends on the record size defined in View | Record Presentation, if active. If that display option is not active, on partitions/volumes with an Ext2/Ext3/Ext4, XFS or UFS file system it is the inode size. In NTFS file systems it is the FILE record size. Another predefined constant is Base_offset. That's the offset in the active data window that the template was applied to by the user, which may change when navigating to adjacent records.

3 Body: Advanced Commands

When enclosed in braces, several variable declarations comprise a block that can be used repeatedly as a whole. Note, however, that blocks must not be *nested* in the current implementation. The character ~ can be used in a variable's name as a placeholder for later replacement with the actual repetition count. The optional `numbering` statement defines where to begin counting (0 by default).

```
numbering  1
{
byte       "len"
string len "String No. ~"
}[10]
```

In this example the actual variable names in the template will be "String No. 1", "String No. 2", ..., "String No. 10". Instead of a constant number of repetitions (10 in this example), you may also specify "unlimited". In that case WinHex will repeat the block until the end of file is encountered. "ExitLoop" can be used to break out of a loop at any time. "Exit" terminates execution of the template completely.

"IfEqual" is useful for the comparison of two expressions. Operands can be either both numerical values, be it constant values in decimal notation, integer variables or a formulas, or byte sequences given as text or hex values which are compared byte by byte. ASCII string expressions

must be enclosed in quotation marks, hex sequences must be preceded by a "0x" identifier. Formulas need to be enclosed in brackets.

```
{
byte       Value
IfEqual    Value 1
     ExitLoop
EndIf
} [10]
```

An "IfEqual" command block is terminated with an "EndIf" statement. If the compared expressions are equal, template interpretation continues after "IfEqual". Optionally, "IfEqual" can be followed by an "Else" statement. The template processor branches into the "Else" block if the expressions are not equal. "IfEqual" commands must not be nested. "IfGreater" is similar to "IfEqual". The condition is true if the first expression is greater than the second. Strings and hex values are compared lexicographically.

In order to facilitate reading and navigating the template, you may define groups of variables that are separated by empty space in the dialog box:

```
section    "...Section Title..."
...
endsection
```

The `section`, `endsection`, and `numbering` statements do not advance the current position in the data to be interpreted.

There are two commands that do not declare variables either, but are explicitly used to change the current position. This can be done to skip irrelevant data (forward movement) or to be able access certain variables more than once as different types (backward movement). Use the `move n` statement to skip `n` bytes from the current position, where `n` may be negative. `goto n` navigates to the specified absolute position from the beginning of the template interpretation (must be positive). `gotoex n` jumps to the specified absolute position based on the start of the data window (e.g. file or disk).

The following example demonstrates how to access a variable both as a 32-bit integer and as a four-part chain of hex values:

```
int32      "Disk serial number (decimal)"
move -4
hex 4      "Disk serial number (hex)"
```

4 Body: Flexible Integer Variables

A special variable type supported by templates is `uint_flex`. This type allows to compose an unsigned integer value from various individual bits within a 32-bit (4-byte) range in an arbitrary order and is even more flexible than a so-called bit field in the C programming language.

uint_flex requires an additional parameter string in inverted commas that specifies exactly which bits are used in which order, separated by commas. The bit listed first becomes the most significant bit (high value bit) in the resulting integer, and it is not interpreted as a + or - indicator. The bit listed last becomes the least significant bit in the resulting integer.

The bits are counted starting with 0. Bit 0 is the bit that is the least significant bit of the 1st byte. Bit 31 is the most significant bit of the fourth byte. Thus, the definition is based on little-endian philosophy.

For example,
`uint_flex "15,14,13,12,11,10,9,8,7,6,5,4,3,2,1,0" "Standard 16-bit integer"`
is exactly the same as `uint16`, the common unsigned 16-bit integer variable.

`uint_flex "31,30,29,28,27,26,25,24,23,22,21,20,19,18,17,16,15,14,13,12, 11,10,9,8,7,6,5,4,3,2,1,0" "Standard 32-bit integer"`
is exactly the same as `uint32`, the common unsigned 32-bit integer variable.

The benefit of `uint_flex`, though, is that the number, the position, and the usage order of all bits can be chosen arbitrarily. For example,
`uint_flex "7,15,23,31" "An unusual 4-bit integer"`
composes a 4-bit integer out of the respective most significant bits of each of the four bytes involved. If these four bytes happen to be
F0 A0 0F 0A =
11110000 **1**0100000 **0**0001111 **0**0001010,
bit 7 is **1**, bit 15 is **1**, bit 23 is **0**, and bit 31 is **0**.
So the resulting uint_flex is **11**0**0** = **1***8 + **1***4 + **0***2 + **0***1 = 12.

Appendix B: Script Commands

Script commands are case-*in*sensitive. Comments may occur anywhere in a script file and must be preceded by two slashes. Parameters may be 255 characters long at most. Where in doubt because hex values, text strings (or even integer numbers) are accepted as parameters, you may use quotation marks to enforce the interpretation of a parameter as text. Quotation marks are *required* if a text string or variable name contains one or more space characters, so that all characters in between are recognized as constituting *one* parameter. If the text within quotation marks is the name of a defined variable, the variable will be used as the parameter.

Wherever numerical parameters are expected (integer numbers), the integrated formula parser allows you to use mathematical expressions. Such expressions need to be enclosed in brackets. They must not contain space characters. They may make use of variables that can be interpreted as integer numbers. Supported operations are addition (+), subtraction (-), multiplication (*), integer division (/), modular division (%), bitwise AND (&), bitwise OR (|), and bitwise XOR (^). Valid mathematical expressions are for example (5*2+1), (MyVar1/(MyVar2+4)), or (-MyVar).

The following is a description of currently supported script commands, including example parameters.

Create "D:\My File.txt" 1000
Creates the specified file with an initial file size of 1000 bytes. If the file already exists, it is overwritten.

Open "D:\My File.txt"
Open "D:*.txt"
Opens the specified file(s). Specify "?" as the parameter to let the user select the file to open.

Open C:
Open D:
Opens the specified logical drive. Specify ":?" as the parameter to let the user select a logical drive or physical disk to open.

Open 80h
Open 81h
Open 9Eh
Opens the specified physical media. Floppy disk numbering starts with 00h, fixed and removable drive numbering with 80h, optical media numbering with 9Eh.

Optionally, you may pass a second parameter with the Open command that defines the edit mode in which to open the file or media ("in-place" or "read-only").

CreateBackup
Creates a WHX backup of the active file in its current state.

CreateBackupEx 0 100000 650 true "F:\My backup.whx"
Creates a WHX backup of the active disk, from sector 0 through sector 1,000,000. The backup file will be split automatically at a size of 650 MB. Compression is enabled (“true”). The output file is specified as the last parameter.
If the backup file should not be split, specify 0 as the third parameter. To disable compression, specify “false”. To have the Backup Manager automatically assign a filename and place the file in the folder for backup files, specify "" as the last parameter.

Goto 0x128
Goto MyVariable
Moves the current cursor position to the hexadecimal offset 0x128. Alternatively, an existing variable (up to 8 bytes large) can be interpreted as a numeric value, too.

Move -100
Moves the current cursor position 100 bytes back (decimal).

Write "Test"
Write 0x0D0A
Write MyVariable

Writes the four ASCII characters "Test" or the two hexadecimal values "0D0A" at the current position (in overwrite mode). Can also write the contents of a variable specified as the parameter. Moves the current position forward by the number of bytes written. When the end of the file is reached, to accomplish that, a null byte is appended. Useful so that further Write commands don't overwrite the last byte written by the previous Write command.

Write2
Identical to Write, but does not append a null byte if the end of the file has been reached. So it is not safe to assume that Write2 always moves the current position forward by the number of bytes written.

Insert "Test"
Functions just as the "Write" command, but in *insert* mode. Must only be used with *files*.

Read MyVariable 10
Reads the 10 bytes from the current position into a variable named "MyVariable". If this variable does not yet exist, it will be created. Up to 48 different variables allowed. Another way to create a variable is the Assign command.

ReadLn MyVariable
Reads from the current position into a variable named "MyVariable" until the next line break is encountered. If the variable already exists, its size will be adjusted accordingly.

Close
Closes the active window without saving.

CloseAll
Closes all windows without saving.

Save
Saves changes to the file or disk in the active window.

SaveAs "C:\New Name.txt"
Saves the file in the active window under the specified path. Specify "?" as the parameter to let the user select the destination.

SaveAll
Saves changes in all windows.

Terminate
Aborts script execution.

Exit
Terminates script execution and ends WinHex.

ExitIfNoFilesOpen
Aborts script execution if no files are already opened in WinHex.

Block 100 200
Block "My Variable 1" "My Variable 2"
Defines the block in the active window to run from offset 100 to offset 200 (decimal). Alternatively, existing variables (each up to 8 bytes large) can be interpreted as numeric values.

Block1 0x100
Defines the block beginning to be at the hexadecimal offset 0x100. A variable is allowed as the parameter as well.

Block2 0x200
Defines the block end to be at the hexadecimal offset 0x200. A variable is allowed as the parameter as well.

Copy
Copies the currently defined block into the clipboard. If no block is defined, it works as known from the Copy command in the Edit menu.

Cut
Cuts the currently defined block from the file and puts it into the clipboard.

Remove
Removes the currently defined block from the file.

CopyIntoNewFile "D:\New File.dat"
CopyIntoNewFile "D:\File +MyVariable+.dat"
Copies the currently defined block into the specified new file, without using the clipboard. If no block is defined, it works as known from the Copy command in the Edit menu. Can copy disk sectors as well as files. Allows an unlimited number of "+" concatenations in the parameter. A variable name will be interpreted as an integer if not be larger than 2^24 (~16 Mio.). Useful for loops and file recovery.

Paste
Pastes the current clipboard contents at the current position in a file, without changing the current position.

WriteClipboard
Writes the current clipboard contents at the current position in a file or within disk sectors, without changing the current position, by overwriting the data at the current position.

Convert *Param1 Param2*
Converts the data in the active file from one format into another one. Valid parameters are ANSI, IBM, Binary, HexASCII, IntelHex, MotorolaS, Base64, UUCode, LowerCase, UpperCase, and hiberfil,, in combinations as known from the Convert menu command.

AESEncrypt "My Password"
Encrypts the active file or disk, or selected block thereof, with the specified key (up to 32

characters long) with AES.

AESDecrypt "My Password"
Decrypts the active file or disk.

Find "John" [*MatchCase MatchWord Down Up BlockOnly SaveAllPos Unicode Wildcards*]
Find 0x1234 [*Down Up BlockOnly SaveAllPos Wildcards*]
Searches in the active window for the name John or the hexadecimal values 0x1234, respectively, and stops at the first occurrence. Other parameters are optional. By default, WinHex searches the entire file/disk. The optional parameters work as known from usual WinHex search options.

ReplaceAll "Jon" "Don" [*MatchCase MatchWord Down Up BlockOnly Unicode Wildcards*]
ReplaceAll 0x0A 0x0D0A [*Down Up BlockOnly Wildcards*]
Replaces all occurrences of either a string or hexadecimal values in the active file with something else. Can only be applied to a disk if in in-place mode.

IfFound
A boolean value that depends on whether or not the last Find or ReplaceAll command was successful. Place commands that shall be executed if something was found after the IfFound command.

IfEqual MyVariable "Hello World"
IfEqual 0x12345678 MyVariable
IfEqual MyVariable 1000
IfEqual MyVariable MyOtherVariable
IfEqual MyVariable (10*MyOtherVariable)
Compares either two numerical integer values (each of them being a constant value, an integer variable or a mathematical expression) or two variables, ASCII strings, or hexadecimal values at the binary level. Comparing two objects at the binary with a different length always returns False as the result. If equal, the following commands will be executed. If conditions must not be nested.

IfGreater MyVariable "Hello World"
IfGreater 0x12345678 MyVariable
IfGreater MyVariable 1000
IfGreater MyVariable MyOtherVariable
IfGreater MyVariable (10*MyOtherVariable)
Accepts the same parameters as IfEqual. If the first one is greater than the second one, the following commands will be executed. If conditions must not be nested.

Else
May occur after IfFound or IfEqual. Place commands that shall be executed if nothing was found or if the compared objects are not equal after the Else command.

EndIf
Ends conditional command execution (after IfFound, IfEqual, IfGreater).

{...

ExitLoop
...}
Exits a loop. A loop is defined by braces. Closing braces may be followed by an integer number in square brackets, which determines the number of loops to execute. This is may also be a variable or the keyword "unlimited" (so the loop can only be terminated with an ExitLoop command). Loops must not be nested.

Example of a loop:
{ Write "Loop" }[10] will write the word "Loop" ten times.

Label ContinueHere
Creates a label named "ContinueHere"

JumpTo ContinueHere
Continues script execution with the command following that label.

NextObj
Switches cyclically to the next open window and makes it the "active" window. E.g. if 3 windows are open, and window #3 is active, NextObj will make #1 the active window.

ForAllObjDo
The following block of script commands (until **EndDo** occurs) will be applied to all open files and disks.

CopyFile C:\A.dat D:\B.dat
Copies the contents of C:\A.dat into the file D:\B.dat.

MoveFile C:\A.dat D:\B.dat
Moves the file C:\A.dat to D:\B.dat.

DeleteFile C:\A.dat
Surprisingly, deletes C:\A.dat.

InitFreeSpace
InitSlackSpace
Clears free space or slack on the current logical drive, respectively, using the currently set initialization settings. InitSlackSpace switches the drive temporarily to in-place mode, thus saving all pending changes.

InitMFTRecords
Clears unused MFT FILE records on the current logical drive if it is formatted with NTFS, using the currently set initialization settings. Simply does nothing on other file systems. The changes are written immediately to the disk.

Assign MyVariable 12345
Assign MyVariable 0x0D0A
Assign MyVariable "I like WinHex"

Assign MyVariable MyOtherVariable
Stores the specified integer number, binary data, ASCII text, or other variable's contents in a variable named "MyVariable". If this variable does not yet exist, it will be created. Other ways to create variables: e.g. Read, GetUserInput, InttoStr. Up to 48 different variables allowed to exist simultaneously.

Release MyVariable
Specifically disposes an existing variable. Mandatory to invoke only when more than 48 variables with different names are to be used during the execution of a script, so that earlier variables that are not needed any more can be destroyed.

SetVarSize MyVariable 1
SetVarSize MyVariable 4
Explicitly sets the allocated memory size of a variable at a given time, in bytes. This can be useful e.g. for variables that hold integer values and that are the result of a calculation, if this value is to be written to a binary file with a fixed-length structure. Without SetVarSize, no assumption must be made about the size of the variable. For instance, the number 300 could be stored in any number of bytes larger than 1. If the new size set by SetVarSize is smaller than the old size, the allocated memory is truncated. If the new size is larger, the allocated memory is expanded. At any rate, the value of the persisting bytes is retained.

GetUserInput MyVariable "Please enter your name:"
Stores the ASCII text or binary data (0x...) specified by the user at script execution time (128 bytes at max.) in a variable named "MyVariable". The user is prompted by the message you provide as the second parameter. If the variable does not yet exist, it will be created. Other ways to create variables: Assign, Read.

GetUserInputI MyIntegerVariable "Please enter your age:"
Works like GetUserInput, but accepts and stores only integer numbers.

Inc MyVariable
Interprets the variable as an integer (if not larger than 8 bytes) and increments it by one. Useful for loops.

Dec MyVariable
Interprets the variable as an integer (if not larger than 8 bytes) and decrements it by one.

IntToStr MyStr MyInt
IntToStr MyStr 12345
Stores the decimal ASCII text representation of the integer number specified as the second parameter in a variable specified as the first parameter.

StrToInt MyInt MyStr
Stores the binary representation of the integer number specified as a decimal ASCII string in the second parameter in a variable specified as the first parameter.

StrCat MyString MyString2

StrCat MyString ".txt"
Appends one string to another. The second parameter may be a variable or a constant string. The first parameter must be a variable. The result will be saved in the variable specified by the first parameter and must not be longer than 255 characters.

GetClusterAlloc MyStr
May be applied to a logical volume. Retrieves a textual description of the current position's allocation, e.g. which file is stored in the current cluster, and saves that description in the specified variable.

GetClusterAllocEx IntVar
May be applied to a logical volume. Retrieves an integer value that indicated whether the cluster at the current position is allocated (1) or not (0), and saves that description in the specified variable.

GetClusterSize IntVar
May be applied to a logical volume. Retrieves the cluster size and saves that value in the specified integer variable.

InterpretImageAsDisk
Treats a raw image or evidence file like the original physical disk or partition. Requires a specialist or forensic license.

CalcHash HashType MyVariable
CalcHashEx HashType MyVariable
Calculates a hash as known from the command in the Tools menu and stores it in the specified variable (which will be created if it does not yet exist). The HashType parameter must be one of the following: CS8, CS16, CS32, CS64, CRC16, CRC32, MD5, SHA-1, SHA-256, PSCHF. CalcHashEx in addition displays the hash in a dialog window.

MessageBox "Caution"
Displays a message box with the text "Caution" and offers the user an OK and a Cancel button. Pressing the Cancel button will abort script execution.

ExecuteScript "ScriptName"
Executes another script from within a running script, at the current execution point, e.g. depending on a conditional statement. Calls to other scripts may be nested. When the called script is finished, execution of the original script will be resumed with the next command. This feature can help you structure your scripts more clearly.

Turbo On
Turbo Off
In turbo mode, most screen elements are not updated during script execution and you are not able to abort (e.g. by pressing Esc) or pause. This may accelerate script execution if a lot of simple commands such as Move and NextObj are executed in a loop.

Debug

All the following commands must be confirmed individually by the user.

UseLogFile
Error messages are written into the log file "Scripting.log" in the folder for temporary files. These messages are not shown in a message box that requires user interaction. Useful especially when running scripts on unattended remote computers.

CurrentPos
GetSize
unlimited
are keywords that act as a placeholders and may be used where numeric parameters are required. On script execution, CurrentPos stands for the current offset in the active file or disk window and GetSize for its size in bytes. unlimited actually stands for the number 2,147,483,647.