



JOHANNES KEPLER
UNIVERSITÄT LINZ
Netzwerk für Forschung, Lehre und Praxis



Miniaturansichten und Zeitstempel unter Microsoft Windows

SCHRIFTLICHE AUSARBEITUNG

Praktikum aus Netzwerke und Sicherheit

LVA-Nummer: 353.000, SS 2010

Angefertigt am *Institut für Informationsverarbeitung und Mikroprozessortechnik (FIM)*

Betreuung:

Priv.-Doz. Mag. iur. Dipl.Ing. Dr. Michael Sonntag

Eingereicht von:

Thomas Schmittner, BSc (0556270)

Linz, Oktober 2010

Kurzfassung

Diese Arbeit behandelt zwei voneinander unabhängige Themen unter Microsoft Windows und geht dabei speziell auf *Microsoft Windows XP*, *MS Windows Vista*, *MS Windows 7*, *MS Windows Server 2003* und *MS Windows Server 2008* ein.

Im ersten Teil wird die automatische Erstellung von Miniaturansichten bei der Arbeit mit Bildern untersucht und dabei auf die Unterschiede zwischen den einzelnen Windows-Versionen genauer eingegangen. Es werden einige Tipps gegeben, wie man Festplattenplatz sparen oder Rückschlüsse auf gespeicherte Bilder ziehen kann, was vor allem im Bereich der Computerforensik Bedeutung hat.

Der zweite Teil widmet sich den unterschiedlichen Zeitstempeln von Windows mit einem NTFS-Dateisystem. Dabei wird beschrieben, bei welchen *Aktionen* (Erstellen, Öffnen, Kopieren, Umbenennen, Entpacken, . . .) welche *Zeitstempel* von Dateien geändert werden. Diese Auswertung wird mit einer in C# selbst implementierten Software durchgeführt.

Abstract

This paper deals with two completely unrelated topics under Microsoft Windows with a main focus on *Microsoft Windows XP*, *MS Windows Vista*, *MS Windows 7*, *MS Windows Server 2003* and *MS Windows Server 2008*.

The first part is about the automatic creation of thumbnails when working with images under Windows and the differences between various Windows versions. Furthermore it is shown how to restrict the size of these thumbnail caches or how to avoid creating them. This might save disk space and lets you cover your tracks, which is a very important fact in computer forensics.

The second part deals with different timestamps under Windows with an NTFS-filesystem and how they change when performing various actions (create, open, copy, rename, unzip, ...). Additionally it shows how a self-written program in C# helps collecting and analyzing this data.

Inhaltsverzeichnis

1	Einführung	1
2	Automatisch erstellte Miniaturansichten unter MS Windows	2
2.1	Microsoft Windows XP	2
2.1.1	Speichern von Miniaturbildern (Datei <i>thumbs.db</i>)	2
2.1.2	Löschen der Datei <i>thumbs.db</i>	3
2.1.3	Größe der Datei <i>thumbs.db</i> minimieren	4
2.1.4	Caching dauerhaft ausschalten	5
2.1.5	Zusammenfassung	5
2.2	Microsoft Windows Vista	5
2.2.1	Erstellung von Thumbnails	6
2.2.2	Format der Thumbcaches	7
2.2.2.1	Thumbcache_idx.db (IMMM)	7
2.2.2.2	Thumbcache_*.db	8
2.2.2.3	Thumbcache_sr.db	10
2.2.3	Suchen von Thumbnails im Cache	10
2.2.4	Löschen von Thumbcaches	11
2.2.5	Dauerhaftes Ausschalten des Thumbnail-Caching	12
2.3	Microsoft Windows 7	13
2.3.1	Minimale Unterschiede zu Windows Vista	14
3	Untersuchung von Zeitstempeln	15
3.1	Allgemeines	15
3.2	Entwicklung des Testprogramms	16
3.2.1	Packen und Entpacken einer Datei	17
3.2.2	Löschen und Wiederherstellen einer Datei (Papierkorb)	17
3.3	Auswertung	18
3.3.1	Microsoft Windows XP	20
3.3.2	Microsoft Windows Server 2003	20
3.3.3	Microsoft Windows Vista	23
3.3.4	Microsoft Windows Server 2008	24
3.3.5	Microsoft Windows 7	26
	Literaturverzeichnis	28

Abbildungsverzeichnis

2.1	Ansicht <i>Filmstreifen</i> oder <i>Miniaturansicht</i>	3
2.2	Magic Number <i>IMMM</i> von <i>Thumbcache_idx.db</i>	7
2.3	Magic Number <i>CMMM</i> von <i>Thumbcache_*.db</i>	8
2.4	Eindeutige ID eines Thumbnails in der Datei <i>thumbcache_idx.db</i>	11
2.5	Gleiche ID des Thumbnails in der Datei <i>thumbcache_32.db</i>	12
3.1	Vergleich der Ergebnisse bei allen getesteten Windows-Versionen	18
3.2	Ergebnis der Dateioperationen unter Windows XP	19
3.3	Ergebnis der Dateioperationen unter Windows XP - Manueller Test	19
3.4	Ergebnis der Dateioperationen unter Windows Server 2003	21
3.5	Ergebnis der Dateioperationen unter Windows Server 2003 - Manueller Test	22
3.6	Ergebnis der Dateioperationen unter Windows Vista	23
3.7	Ergebnis der Dateioperationen unter Windows Vista - Manueller Test	24
3.8	Ergebnis der Dateioperationen unter Windows Server 2008	25
3.9	Ergebnis der Dateioperationen unter Windows Server 2008 - Manueller Test	25
3.10	Ergebnis der Dateioperationen unter Windows 7	26
3.11	Ergebnis der Dateioperationen unter Windows 7 - Manueller Test	27

Kapitel 1

Einführung

Microsoft Windows ist für viele Benutzer ein täglich Brot. Dabei ist es laufend notwendig, sich an neue Versionen und Veränderungen der Systeme anzupassen. In dieser Arbeit werden zwei Themen behandelt, die in allen (aktuelleren) Versionen von MS Windows eine Rolle spielen.

Im ersten großen Teil der Arbeit geht es um Miniaturansichten, die meist ohne das Wissen des Durchschnittsbenutzers automatisch erstellt werden. Dabei wird untersucht, unter welchen Umständen diese Thumbnails erzeugt werden, wie sich die verschiedenen Windows-Versionen dabei unterscheiden und wie man die automatische Erstellung einschränken oder verhindern kann.

Der zweite Teil geht auf die unterschiedlichen Zeitstempel von Dateien in einem NTFS-Dateisystem ein. Dabei wird festgestellt, welche Aktionen welche Auswirkungen auf die Zeitstempel haben und ob sich dieses Verhalten bei allen Windows-Versionen beobachten lässt, oder ob es Unterschiede zwischen den einzelnen Versionen gibt. Außerdem wird hier eine für diesen Zweck in C# implementierte Software vorgestellt, die vollautomatisiert Informationen zu den verschiedenen Aktionen (Kopieren, Verschieben, Umbenennen, etc.) sammelt und so einen relativ schnellen und einfachen Überblick darüber gibt, wie sich die Windows-Versionen unterscheiden.

Beide Problemstellungen beschränken sich auf die gängigen Versionen von MS Windows, nämlich *Windows XP*, *Windows Vista* und *Windows 7*. Während im zweiten Teil zusätzlich auf *Windows Server 2003* und *Windows Server 2008* eingegangen wird, spielen diese Versionen bei den Miniaturansichten im ersten Teil wenig bis gar keine Rolle, da es sich um keine End-User-Versionen handelt, auf denen ein Benutzer mit Fotos und Bildern arbeiten kann.

Kapitel 2

Automatisch erstellte Miniaturansichten unter MS Windows

Bei der Betrachtung von Bildern unter MS Windows werden unter gewissen Umständen automatisch Miniaturansichten erstellt. In diesem Teil der Arbeit wird dieses Thema behandelt und versucht, festzustellen, bei welchen konkreten Operationen Miniaturbilder erzeugt werden, welche Unterschiede zwischen den Windows-Versionen *XP*, *Vista* und *7* zu finden sind und wie lange diese „Thumbnails“ im System gespeichert bleiben.

Vorweg ist festzuhalten, dass sich Windows XP von seinen Nachfolgern Windows Vista und Windows 7 beim Umgang mit „Thumbnails“ wesentlich unterscheidet. Windows XP speichert pro Verzeichnis eine Datei, welche die Miniaturbilder enthält, während bei Windows Vista und Windows 7 eine völlig andere Strategie verfolgt wird und alle Bilder zentral gespeichert und verwaltet werden. Mehr Informationen dazu geben Kapitel 2.1 auf Seite 2 (Windows XP) und die Kapitel 2.2 auf Seite 5 (Windows Vista) sowie 2.3 auf Seite 13 (Windows 7).

2.1 Microsoft Windows XP

2.1.1 Speichern von Miniaturbildern (Datei *thumbs.db*)

Windows XP speichert von Bildern in einem Verzeichnis automatisch Miniaturansichten, um eine schnellere Vorschau der Bilder im Explorer zu ermöglichen und um diese nicht immer neu erstellen zu müssen.

Dies geschieht in einer Datei mit dem Namen *thumbs.db*. Dieser Dateiname steht für „Thumbnails“, was soviel wie „Miniaturansichten“ bedeutet und hat die Dateierweiterung „.db“, die für „Database“ (Datenbank) steht. Dieser Dateiname lässt darauf schließen, dass es sich dabei um eine Art Datenbank handelt, die die Miniaturansichten der einzelnen Dateien abspeichert.

Diese Datei wird von Windows allerdings nicht automatisch in jedem Verzeichnis generiert, sondern nur dann, wenn man im Explorer im Menüpunkt Ansicht „Miniaturansicht“ oder „Filmstreifen“ wählt. (siehe Abbildung 2.1 auf Seite 3)

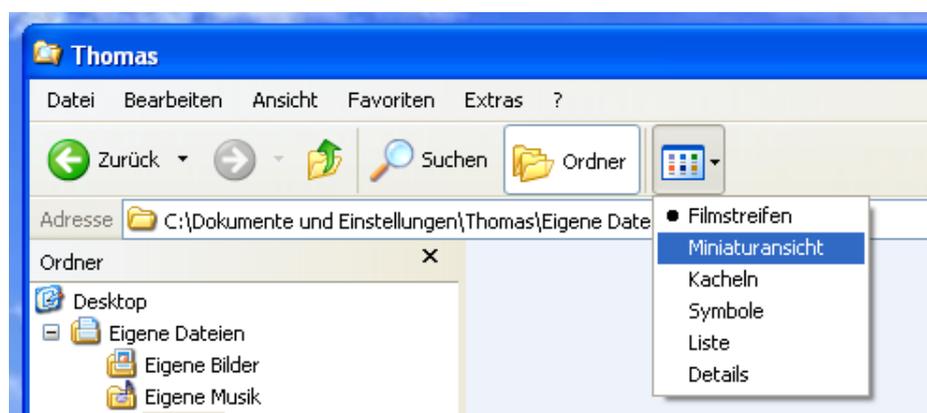


Abbildung 2.1: Ansicht *Filmstreifen* oder *Miniaturansicht*

Wichtig ist, dass es sich bei der Datei *thumbs.db* um eine versteckte Systemdatei handelt, die mit den Standardeinstellungen für den Benutzer nicht sichtbar ist. Um diese Datei anzuzeigen, muss im Explorer im Menüpunkt *Extras* > *Ordneroptionen* > *Ansicht* der Punkt „Geschützte Systemdateien ausblenden (empfohlen)“ deaktiviert werden.

2.1.2 Löschen der Datei *thumbs.db*

Diese Datei kann jederzeit vom Benutzer gelöscht werden, wird jedoch vom System unter gewissen Umständen automatisch wiedererzeugt.

Hat man die Datei per Hand gelöscht, wird diese - solange der Ordner geöffnet ist (egal ob minimiert oder maximiert) - nicht neu erstellt. Trotzdem erfolgt die Anzeige der Miniaturansichten auch bei einer Aktualisierung (F5) nach wie vor sehr rasch. Dies funktioniert vermutlich deswegen, weil sich die Bilder noch immer im Hauptspeicher befinden.

Erst beim Schließen des Fensters, dem Öffnen des gleichen Verzeichnisses in einem

anderen Explorer-Fenster oder dem Zurückwechseln zum Verzeichnis von einem anderen Pfad, wird die Datei *thumbs.db* sofort neu generiert.

Obwohl die Datei normalerweise unsichtbar ist, wird sie beim Verschieben oder Kopieren des Ordners mitkopiert oder mitverschoben. Markiert man allerdings im Verzeichnis direkt (*Strg + A* oder *Bearbeiten > Alles markieren*) alle Bilder, so bleibt die Datei unberührt im Verzeichnis liegen, selbst wenn alle Bilder gelöscht werden. Wird allerdings das Verzeichnis selbst gelöscht, verschwindet die Datei natürlich mit diesem.

Werden in einem Verzeichnis die eigentlichen Bilder gelöscht oder verschoben, die Datei *thumbs.db* aber nicht, so kann mit einer entsprechenden Software (z.B. *Windows File Analyzer*) sehr einfach nachvollzogen werden, welche Bilder hier ursprünglich gespeichert waren, denn die Datei lässt sich ohne Probleme auslesen.

Was hier unbedingt noch erwähnt werden muss ist, dass die Datei, sofern sie nicht per Hand gelöscht und vom System neu erstellt wurde, alle jemals in diesem Ordner in Miniaturansicht angezeigten Bilder archiviert. Die Datei wird zwar jedes Mal verändert, wenn Bilder angezeigt werden, aber nicht überschrieben. Alte Thumbnails bleiben in der Datei gespeichert, bis diese vom Benutzer explizit gelöscht wird. Diese Tatsache ist vor allem im Bereich der Forensik interessant und oft sehr nützlich.

2.1.3 Größe der Datei *thumbs.db* minimieren

Da diese Datei bei einer Menge von Bildern in einem Verzeichnis unter Umständen sehr groß werden kann, gibt es unter Windows XP zwei Möglichkeiten, die Dateigröße zu minimieren. Beide erfordern eine Änderung des Registry-Schlüssels `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer`. Unter dem Menüpunkt *Bearbeiten* können die beiden Werte *ThumbnailSize* und *ThumbnailQuality* angelegt werden.

- **ThumbnailSize**
Über das Menü *Bearbeiten > Neu > DWORD-Wert* wird ein neues *DWORD* mit dem Namen *ThumbnailSize* erzeugt, dessen Wert zwischen 32 und 256 Pixel pro Miniaturbild liegt. Der Standardwert liegt bei 96 Pixel.
- **ThumbnailQuality**
Hier ist ein neues *DWORD* mit dem Namen *ThumbnailQuality* zu erstellen, dessen Wert zwischen 50 und 100 die Qualität der Thumbnails angibt.

2.1.4 Caching dauerhaft ausschalten

Um zu verhindern, dass Windows die Miniaturansichten in einer Datei abspeichert, kann man diese Funktionalität deaktivieren. Dadurch verhindert man allerdings auch den bequemen und schnellen Zugriff auf die Miniaturbilder in den zuvor genannten Explorer-Ansichten *Filmstreifen* und *Miniaturansicht*. Im Menü *Extras > Ordneroptionen > Ansicht* wird dazu der Punkt „Miniaturansichten nicht zwischenspeichern“ aktiviert. Diese Einstellung hat zur Folge, dass von diesem Zeitpunkt an keine Datei *thumbs.db* mehr vom System generiert wird und keine dauerhafte Speicherung der Thumbnails erfolgt.

2.1.5 Zusammenfassung

- Datei *thumbs.db* befindet sich in jedem Ordner, in dem Bilder als Miniaturansicht angezeigt wurden
- Datei ist eine versteckte und geschützte Systemdatei und somit mit den Standardeinstellungen für den Benutzer unsichtbar
- Datei bleibt erhalten, außer sie wird explizit oder mit dem ganzen Verzeichnis gelöscht
- Datei beinhaltet alle jemals angezeigten Bilder dieses Verzeichnisses, sofern sie nicht zwischendurch per Hand gelöscht wurde
- Datei wird mit dem Verzeichnis mitkopiert, nicht allerdings wenn die Bilder einzeln kopiert oder verschoben werden
- Größe und Qualität der Thumbnails kann in der Registry eingestellt werden
- Erstellung kann über Menü verhindert werden

2.2 Microsoft Windows Vista

Im Gegensatz zu Windows XP verwendet Windows Vista nicht für jedes Verzeichnis eine eigene Datenbank mit dem Namen *thumbs.db*, sondern verwaltet einen zentralen Thumbs-Cache. Das hat den Vorteil, dass nicht in jedem Ordner eine Datenbank-Datei mit Miniaturbildern angelegt wird. Diese Tatsache mag einem Benutzer durchaus etwas

lästig erscheinen, da sich immer eine unsichtbare Datei im Ordner befindet. Dadurch stimmt weder die Anzahl der Dateien im Verzeichnis, noch die Gesamtgröße dieser. Der Pfad zum zentralen Cache unter Windows Vista ist `%homedrive%\Users\%username%\AppData\Local\Microsoft\Windows\Explorer` und das Verzeichnis beinhaltet mehrere verschiedene Dateien:

- Thumbcache_32.db
- Thumbcache_96.db
- Thumbcache_256.db
- Thumbcache_1024.db
- Thumbcache_idx.db
- Thumbcache_sr.db

In diesen Dateien sind die Miniaturbilder in unterschiedlichen Größen abgespeichert. Diese gehen von 32 Pixel bis 1024 Pixel und sind je nach Größe im jeweiligen Thumbcache abgelegt. Die Datei *thumbcache_idx.db* ist eine Art zentraler Index und erleichtert den Zugriff auf die Bilder in den verschiedenen Caches.

Bilder mit einer Auflösung von 1024 Pixel werden beispielsweise erstellt, wenn das „Vorschaufenster“ im Windows-Explorer aktiviert ist. (*Organisieren > Layout > Vorschaufenster*) Diese haben bereits eine durchaus akzeptable Qualität, die zum Beispiel auch in der Computerforensik eine große Bedeutung haben kann, da ein Bild mit solcher großer Auflösung einen guten Ersatz für ein gelöscht Originalbild darstellt.

2.2.1 Erstellung von Thumbnails

Windows Vista erstellt im Gegensatz zu Windows XP auch für Bilder auf Wechseldatenträgern, Netzwerklaufrwerken und verschlüsselten Datenträgern Miniaturbilder im lokalen zentralen Cache. Je nach gewählter Ansicht und damit verbundener Auflösung der Kleinbilder werden die Thumbnails im entsprechenden Cache abgespeichert.

2.2.2 Format der Thumbcaches

In diesem Kapitel wird beschrieben, wie die in Kapitel 2.2 auf Seite 5 beschriebenen Dateien genau aufgebaut sind.

2.2.2.1 Thumbcache.idx.db (IMMM)

Diese Datei ist eine Art Index aller Miniaturbilder in den verschiedenen *Thumbcache_XX-Dateien*. Sie besteht aus einem Header und beliebig vielen Einträgen für die Einzelbilder. Der Aufbau des Headers ist folgender:

```
typedef struct {
    CHAR magic [4];
    DWORD unk1;
    DWORD unk2;
    DWORD headerSize;
    DWORD entryCount;
    DWORD unk4;
} IMMMH;
```

Die ersten vier Zeichen der Datei sind eine „Magic Number“ und lauten „IMMM“, was in Abbildung 2.2 zu sehen ist.

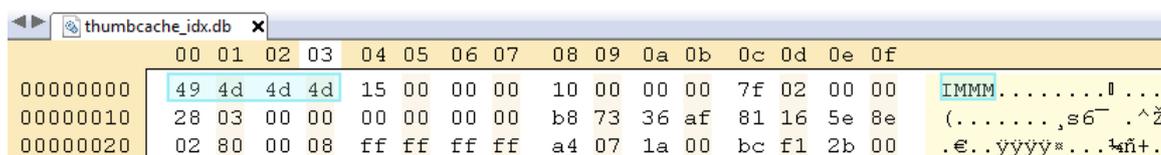


Abbildung 2.2: Magic Number *IMMM* von *Thumbcache.idx.db*

Eine weitere wichtige Größe ist *entryCount* im Header, sie legt die Anzahl der folgenden Einträge fest. Jeder Eintrag ist wie folgt aufgebaut:

```
typedef struct {
    UQUAD secret <format=hex>;
    FILETIME lastModified;
    UINT unk2;
    UINT offset32 <format=hex>;
    UINT offset96 <format=hex>;
    UINT offset256 <format=hex>;
    UINT offset1024 <format=hex>;
    UINT offsetsr <format=hex>;
```

```
} CMMM;
```

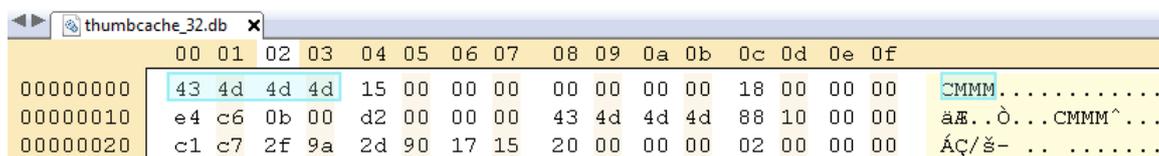
secret Der Eintrag *secret* ist 64 Bit lang und steht für den eindeutigen Bezeichner eines Bildes. Dieser basiert auf dem Dateinamen der Originaldatei, den Daten und unter Umständen dem Bearbeitungsdatum.

unk2 Der Wert *unk2* gibt an, ob der Eintrag in einem der Thumbcaches vorhanden ist (0) oder nicht (1).

offset* Die diversen Offsets geben an, an welcher Stelle im jeweiligen Cache das Miniaturbild zu finden ist. Ist ein Wert auf -1 (0xFFFFFFFF) gesetzt, bedeutet das, dass es keinen Eintrag in der gewünschten Größe gibt.

2.2.2.2 Thumbcache_*.db

Die eigentlichen Thumbcaches haben ebenfalls einen Header und danach Einträge für die einzelnen Thumbnails. Der Header beginnt wieder mit einer *Magic Number*, in diesem Fall mit „CMMM“, was in Abbildung 2.3 zu sehen ist.



	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
00000000	43	4d	4d	4d	15	00	00	00	00	00	00	00	18	00	00	00	CMMM.....
00000010	e4	c6	0b	00	d2	00	00	00	43	4d	4d	4d	88	10	00	00	äÆ..ò...CMMM^...
00000020	c1	c7	2f	9a	2d	90	17	15	20	00	00	00	02	00	00	00	ÁÇ/š- ..

Abbildung 2.3: Magic Number *CMMM* von Thumbcache_*.db

Der Header ist folgendermaßen aufgebaut:

```
typedef struct {
    CHAR magic[4];
    DWORD unk1;
    DWORD unk2;
    DWORD headerSize;
    DWORD offsetLastEntry;
    DWORD entryCount;
} CMMH;
```

unk1 Das DWORD *unk1* gibt die Größe des Headers in Bytes nach der 4 Byte langen Magic Number an.

unk2 Das DWORD *unk2* gibt an, in welcher Größe die Miniaturbilder in dieser Datei gespeichert werden. Mögliche Werte sind in diesem Fall 0, 1, 2 und 3, die für 32x32, 96x96, 256x256 und 1024x768 Pixel stehen.

headerSize Das DWORD *headerSize* steht für die Länge des gesamten Headers in Bytes.

offsetLastEntry Das DWORD *offsetLastEntry* steht für das Offset des letzten CMMM-Eintrags und wird dafür verwendet, neue Einträge schnell und einfach an die bestehenden anzuhängen.

Gleich nach dem Header folgen die einzelnen Einträge, wobei die Anzahl im Header in *entryCount* festgelegt ist und jeder Eintrag wiederum mit der Zeichenfolge *CMMM* beginnt. Die Einträge sind folgendermaßen aufgebaut:

```
typedef struct {
    CHAR magic[4];
    DWORD sizeHeaderAndData;
    UQUAD secret<format=hex>;
    CHAR ext[8];
    DWORD huh1;
    DWORD type;
    DWORD dataSize;
    DWORD unk1;
    DWORD unk2;
    DWORD unk3;
    DWORD unk4;
    DWORD unk5;
    CHAR name[huh1];
    if( sizeHeaderAndData - size > 88 )
        CHAR padding[ sizeHeaderAndData - size - 88 ];
    if( size > 0 )
        CHAR data[ size ];
} CMMM;
```

Die wichtigsten Einträge dabei sind:

sizeHeaderAndData Das DWORD *sizeHeaderAndData* gibt die Gesamtgröße von Header und Daten an.

secret Der Eintrag *secret* ist dieselbe eindeutige ID wie sie im Indexfile *thumbcache_idx.db* zu finden ist.

name Im Character-Array *name* können die Dateieindungen der verschiedenen Dateitypen gespeichert sein. Mögliche Werte an dieser Stelle sind *jpg*, *bmp*, *text*, es kann aber auch leer (0) sein. Windows speichert nicht nur Miniaturansichten von diversen Bildformaten, sondern auch verschiedene Ordnersymbole oder Filesymbole wie zum Beispiel die von Text-Dateien.

huh1 Das DWORD *huh1* gibt die Länge des Unicode-Strings *name* an und hat einen Wert zwischen 30 und 32.

2.2.2.3 Thumbcache_sr.db

Die genaue Aufgabe dieser Datei ist nicht ganz klar, sie hat einen fixen Inhalt, der sich nicht zu ändern scheint, wenn der Inhalt der anderen Thumbcache-Dateien modifiziert wird. Bei Windows Vista hat die Datei folgenden hexadezimalen Inhalt: *43 4D 4D 4D 14 00 00 00 04 00 00 00 18 00 00 00 18 00 00 00 00 00 00 00*

Die ersten vier Bytes stehen für die Zeichenfolge *CMMM*, die auch in allen anderen *Thumbcache_*.db*-Dateien zu Beginn zu finden ist. Das folgende Byte (*0x14*) steht mit großer Wahrscheinlichkeit als Kennung für *Windows Vista*, für *Windows 7* ist der Wert *0x15* vorgesehen.

2.2.3 Suchen von Thumbnails im Cache

Das Speicherformat der *thumbcache_*.db*-Dateien ermöglicht es dem System, die richtigen Miniaturbilder für ein beliebiges Bild anzuzeigen. Wenn der Benutzer im Explorer also eine Datei in einer Miniaturansicht anzeigen möchte, wird zuerst im Index *thumbcache_idx.db* nachgesehen und bei einem vorhandenen Eintrag für diese Größe im jeweiligen Cache gesucht.

Der genaue Ablauf sieht wie folgt aus:

1. Lade die Datei *thumbcache_idx.db*
2. Finde den *IMMM-Eintrag* mit der ID *secret*, nach dem gesucht wird
3. Wenn der Offset für die jeweilige *thumbcache_*.db*-Datei nicht -1 ist, öffne diese Datei
4. Finde die Position, die im *IMMM-Eintrag* angegeben ist
5. Lese genau *dataSize* Bytes, um das Miniaturbild zu erhalten

Die beiden Abbildungen 2.4 auf Seite 11 und 2.5 auf Seite 12 verdeutlichen noch einmal die Art und Weise, wie Thumbnails gespeichert sind. In beiden Dateien (*thumbcache_idx.db* und *thumbcache_32.db*) steht die gleiche ID, was eine Suche nach einer Miniaturansicht für eine bestimmte Datei möglich macht.

Name	
▷ struct IMMMH header	
▲ struct IMMM b[101]	
▷ struct IMMM b[0]	
▷ struct IMMM b[1]	
▷ struct IMMM b[2]	
▷ struct IMMM b[3]	
▷ struct IMMM b[4]	
▷ struct IMMM b[5]	
▷ struct IMMM b[6]	
▷ struct IMMM b[7]	
▷ struct IMMM b[8]	
▷ struct IMMM b[9]	
▷ struct IMMM b[10]	
▷ struct IMMM b[11]	
▷ struct IMMM b[12]	
▷ struct IMMM b[13]	
▷ struct IMMM b[14]	
▷ struct IMMM b[15]	
▷ struct IMMM b[16]	
▷ struct IMMM b[17]	
▷ struct IMMM b[18]	
▲ struct IMMM b[19]	
UQUAD secret	E69EAB1BB16922E5h
FILETIME lastModified	11/02/2006 12:37:20
UINT unk2	0
UINT offset32	18h
UINT offset96	528D4h
UINT offset256	3B87Ch
UINT offset1024	18h
UINT offsetsr	FFFFFFFFh

Abbildung 2.4: Eindeutige ID eines Thumbnails in der Datei *thumbcache_idx.db*

Interessant an dieser Speicherart ist die Generierung der ID, da diese sowohl vom Filenamen, als auch vom Inhalt der Datei abhängt. Wenn eine Datei umbenannt wird, resultiert dies in einem neuen Eintrag mit neuer ID, für zwei idente Dateien mit verschiedenem Namen ergibt sich auch ein unterschiedlicher Bezeichner.

2.2.4 Löschen von Thumbcaches

Um die Thumbcaches vorübergehend zu löschen und den Speicherplatz anderweitig zu verwenden, kann das Windows-Tool „Datenträgerbereinigung“ verwendet werden. Da-

Name	
▷ struct CMMMM header	
▲ struct CMMM c[0]	
▷ CHAR magic[4]	CMMM
DWORD sizeHeaderAndData	3216
UQUAD secret	E69EAB1BB16922E5h
▷ CHAR ext[8]	j
DWORD huh1	32
DWORD type	2
DWORD size	3126
DWORD unk6	0
DWORD unk1	459166235
DWORD unk2	408680331
DWORD unk3	1623572973
DWORD unk4	4215538706
▷ CHAR name[32]	7
▷ CHAR padding[2]	

Abbildung 2.5: Gleiche ID des Thumbnails in der Datei *thumbcache_32.db*

bei kann der Benutzer auswählen, welche Daten entfernt werden sollen und dabei eben auch den Punkt „Miniaturansichten“ anklicken. Nach der Anwendung ist ein Neustart erforderlich, da die alten Files in einen Ordner „CacheFilesToDelete“ verschoben werden und dieser erst nach einem Reboot endgültig gelöscht wird. Dabei werden gleichzeitig alle Cache-Files wieder neu angelegt.

Ein händisches Löschen der Dateien ist nicht möglich, da der Benutzer (auch als Administrator) trotz „Vollzugriffs“ auf die Datei offensichtlich nicht die entsprechenden Rechte besitzt. Vermutlich ist dies der Fall, da die Dateien ständig von einem Prozess verwendet und nicht freigegeben werden.

2.2.5 Dauerhaftes Ausschalten des Thumbnail-Caching

Unter Windows Vista erfordert das Ausschalten des automatischen Thumbnail-Cachings mehrere Schritte und ist nicht ganz so einfach durchzuführen wie beim Vorgänger Windows XP (siehe Kapitel 2.1.4 auf Seite 5). Es gibt die Möglichkeit, statt der Miniaturbilder nur die Dateisymbole anzuzeigen, möchte man aber die Erstellung des Thumbcaches unter `%homedrive%\Users\%username%\AppData\Local\Microsoft\Windows\Explorer` komplett verhindern, sind folgende Schritte durchzuführen:

1. Voraussetzung für diese Schritte ist eine Anmeldung als ein Benutzer mit Administratorrechten

2. Zuerst muss im Explorer unter *Organisieren > Ordner- und Suchoptionen > Ansicht* der Punkt *Immer Symbole statt Miniaturansichten anzeigen* aktiviert werden.
3. Nun kann der Speicherplatz mit Hilfe der *Datenträgerbereinigung* (siehe Kapitel 2.2.4 auf Seite 11) freigegeben werden.
4. Nun müssen noch die Berechtigungen des Verzeichnisses *Explorer* angepasst werden, um dem System zu verbieten, im Cache-Verzeichnis neue Dateien anzulegen. Dazu muss unter *Eigenschaften > Sicherheit > Erweitert > Bearbeiten* die Option *Vererbte Berechtigungen des übergeordneten Objektes einschließen* deaktiviert werden.
5. Nun gibt es mehrere Möglichkeiten des weiteren Vorgehens, wobei die einfachere aber etwas unsauberere Möglichkeit weiter beschrieben wird.
6. Beim erscheinenden Popup ist *Entfernen* zu wählen, um die vererbten Berechtigungen auf diesen Ordner zu löschen.
7. Alle offenen Dialoge sind mit *OK* zu bestätigen
8. Im letzten Schritt kann die in Schritt 2 deaktivierte Miniaturansicht wieder aktiviert werden. Windows Vista erstellt nun bei jedem Ordnerzugriff die Thumbnails neu und speichert diese nicht mehr im Cache-Verzeichnis.

Wenn man auf eine elegantere Lösung abzielt und etwas mehr Verständnis für das Berechtigungs-system unter Windows hat, ist es möglich, die Vererbung wie in der oben stehenden Anleitung aufzubrechen, aber anstatt alle Berechtigungen zu entfernen, die des jeweiligen Benutzers anzupassen. Es reicht aus, den Vollzugriff bzw. den schreibenden Zugriff des Users zu verbieten, um den gewünschten und gleichen Effekt zu haben, ohne dabei die Berechtigungen der anderen Benutzer und Gruppen auf diesem Verzeichnis bearbeiten zu müssen.

2.3 Microsoft Windows 7

Windows 7 unterscheidet sich hinsichtlich Organisation und Speicherung der Miniaturansichten kaum von seinem Vorgänger Windows Vista.

Auch hier gibt es die gleichen *thumbcache_*.db*-Dateien und eine Index-Datei *thumbcache_idx.db* im Verzeichnis

`%homedrive%\Users\%username%\AppData\Local\Microsoft\Windows\Explorer.`

2.3.1 Minimale Unterschiede zu Windows Vista

Minimale Unterschiede gibt es im Vergleich von Windows 7 und Windows Vista im Speicherformat der einzelnen Dateien im Thumb-Cache:

- Die IMMM-Einträge in der Index-Datei *thumbcache_idx.db* haben bei Windows 7 kein 8-Byte langes Feld *Last-Modified*
- Ein CMMM-Eintrag hat bei Windows 7 kein 8-Byte-langes Feld *Extension*
- Die Variable *unk1* im IMMM-Header ist ein Hinweis auf die Version des Betriebssystems und steht nicht für die Länge des Headers (ohne Magic Number). Für Windows Vista steht der Wert *0x14* und für Windows 7 der Wert *0x15*, ebenso wie dies der Fall in der Datei *thumbcache_sr.db* ist (siehe Kapitel 2.2.2.3 auf Seite 10).

Kapitel 3

Untersuchung von Zeitstempeln

Dieses Kapitel beschäftigt sich mit *Zeitstempeln* unter Microsoft Windows. Dabei werden fünf verschiedene Versionen von MS Windows genauer untersucht (*Windows XP*, *Windows Vista* und *Windows 7*, *Windows Server 2003*, *Windows Server 2008*).

Es wird beschrieben, wie bestimmte häufig durchgeführte Aktionen die Zeitstempel einer Datei verändern können. Außerdem wird überprüft, ob sich alle Windows-Versionen gleich verhalten und dafür eine für diesen Zweck entwickelte Software eingesetzt. Dieses Programm ist in C# implementiert und kann auf allen genannten Betriebssystemen ausgeführt werden.

3.1 Allgemeines

Ein *Zeitstempel* ist ein bestimmter Wert, der einem Ereignis einen Zeitpunkt zuordnet. Dieser Zeitpunkt hat ein einheitliches, definiertes Format und den Zweck, mit einer gewissen Fälschungssicherheit darüber Auskunft zu geben, zu welcher genauen Zeit und in welcher Reihenfolge bestimmte Ereignisse stattgefunden haben. Diese Zeitstempel werden oft in Relation zur UTC (Universal Time Coordinated) angegeben und sind daher international vergleichbar.

In einem Dateisystem unter MS Windows hat jede Datei folgende drei Zeitstempel als Attribute:

- Erstelldatum: Creation Date, „Erstellt am“
- Bearbeitungsdatum: Modification Date, „Geändert“

- Zugriffsdatum: Access Date, „*Letzter Zugriff*“

Diese drei Daten werden vom System automatisch verändert, wenn bestimmte Aktionen auf/mit der Datei durchgeführt werden. Grundsätzlich verändern sich die Zeitstempel unter folgenden Umständen:

Erstelldatum Erstellen einer neuen Datei, Kopieren und Verschieben einer bestehenden Datei

Bearbeitungsdatum Erstellen einer neuen Datei, Änderungen des Dateiinhalts

Zugriffsdatum Erstellen einer neuen Datei, Kopieren und Verschieben einer bestehenden Datei, Änderung des Inhalts einer Datei

3.2 Entwicklung des Testprogramms

Für die Sammlung der Informationen über die Änderungen der Zeitstempel wurde ein kleines Programm in C# implementiert, das automatisch die häufigsten Aktionen in einem Dateisystem ausführt und daraus einen grafischen Überblick der Resultate erstellt. Bei der Implementierung wurden der Namespace *System.IO* und die darin zur Verfügung gestellten Operationen auf Dateien und Verzeichnisse, verwendet. Das Programm sammelt Daten für folgende Aktionen:

- Das Erstellen einer neuen Datei
- Das Öffnen einer Datei
- Das Öffnen und Schließen einer Datei
- Das Umbenennen einer Datei
- Das Öffnen, Bearbeiten und Schließen einer Datei
- Das Kopieren einer Datei im aktuellen Verzeichnis
- Das Kopieren einer Datei in ein anderes Verzeichnis
- Das Kopieren einer Datei auf ein anderes Laufwerk (sofern möglich)

- Das Verschieben einer Datei in ein anderes Verzeichnis
- Das Verschieben einer Datei auf ein anderes Laufwerk (sofern möglich)
- Das Packen und Entpacken einer Datei mit *WinRAR*

Im Rahmen der Implementierung der Software muss auf zwei Details etwas genauer eingegangen werden: das *Packen und Entpacken einer Datei* sowie das *Löschen und Wiederherstellen einer Datei*.

3.2.1 Packen und Entpacken einer Datei

Um eine Datei zu packen und wieder zu entpacken, wurde das bekannte Archivierungs- und Dateikompressions-Programm *WinRAR* verwendet. WinRAR kann auch über die Kommandozeile aufgerufen werden, bei allen getesteten Windows-Versionen wurde die gleiche Version von WinRAR verwendet (3.93).

3.2.2 Löschen und Wiederherstellen einer Datei (Papierkorb)

Da sich das Löschen und Wiederherstellen einer Datei programmtechnisch sehr schwer umsetzen lässt, ist diese Aktion rein der Vollständigkeit halber händisch auf allen Systemen durchgeführt worden. Dafür wurde eine Datei händisch gelöscht und kurz darauf aus dem „Papierkorb“ wiederhergestellt. Das Resultat nach einer erfolgreichen Wiederherstellung ist folgendes:

- Erstelldatum: das Erstelldatum bleibt bei allen Windows-Versionen unverändert
- Bearbeitungsdatum: das Bearbeitungsdatum bleibt ebenso bei allen Versionen unverändert
- Zugriffsdatum: das Zugriffsdatum ändert sich bei *Windows XP* und *Windows Server 2003* automatisch auf den Zeitpunkt der Wiederherstellung, bei den anderen Versionen bleibt es unverändert

3.3 Auswertung

Es lässt sich erkennen, dass sich die etwas älteren getesteten Windows-Versionen (*XP* und *Server 2003*) etwas anders verhalten als die Versionen *Vista*, *Server 2008* und *7*. Vor allem in Hinblick auf das Zugriffsdatum erkennt man deutliche Unterschiede zwischen den früheren und den später entwickelten Betriebssystemen. Abbildung 3.1 gibt einen kurzen Überblick über das Ergebnis der Tests.

	Erstelldatum					Bearbeitungsdatum					Letzer Zugriff				
	XP	2003	Vista	2008	7	XP	2003	Vista	2008	7	XP	2003	Vista	2008	7
Neue Datei erstellen	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
Datei öffnen	U	U	U	U	U	U	U	U	U	U	M	M	U	U	U
Datei öffnen und schließen	U	U	U	U	U	U	U	U	U	U	M	M	U	U	U
Datei umbenennen	U	U	U	U	U	U	U	U	U	U	M	M	U	U	U
Datei öffnen, ändern und schließen	U	U	U	U	U	M	M	M	M	M	M	M	U	U	U
Datei in gleichen Ordner kopieren	M	M	M	M	M	U	U	U	U	U	M	M	M	M	M
Datei in anderen Ordner kopieren	M	M	M	M	M	U	U	U	U	U	M	M	M	M	M
Datei auf anderes Laufwerk kopieren	M	M	M	M	M	U	U	U	U	U	M	M	M	M	M
Datei in anderen Ordner verschieben	U	U	U	U	U	U	U	U	U	U	M	M	U	U	U
Datei auf anderes Laufwerk verschieben	U	U	U	U	U	U	U	U	U	U	M	M	M	M	M
Datei packen und entpacken (*.rar)	M	M	M	M	M	U	U	U	U	U	M	M	M	M	M

Abbildung 3.1: Vergleich der Ergebnisse bei allen getesteten Windows-Versionen

Weiters ist festzuhalten, dass sich interessanterweise unterschiedliches Verhalten beobachten lässt, je nachdem wie die Operationen auf und mit den Dateien durchgeführt werden. In dieser Arbeit sind die Zeitstempel auf folgende Varianten getestet worden:

- Mit *Maus* und *Kontextmenü* auf der Benutzeroberfläche im Windows Explorer („händischer Test“)
- Mit der *MS-DOS-Eingabeaufforderung* (*move*, *copy*,...))
- Mit der in C# implementierten Software, also den .NET-Klassenbibliotheken

Die dabei festgestellten Unterschiede in der Veränderung der Zeitstempel einer Datei werden in den folgenden Kapiteln im Detail für jede untersuchte Windows-Version beschrieben. Dabei wurde bei allen Versionen das Dateisystem *NTFS* (New Technology File System) verwendet.

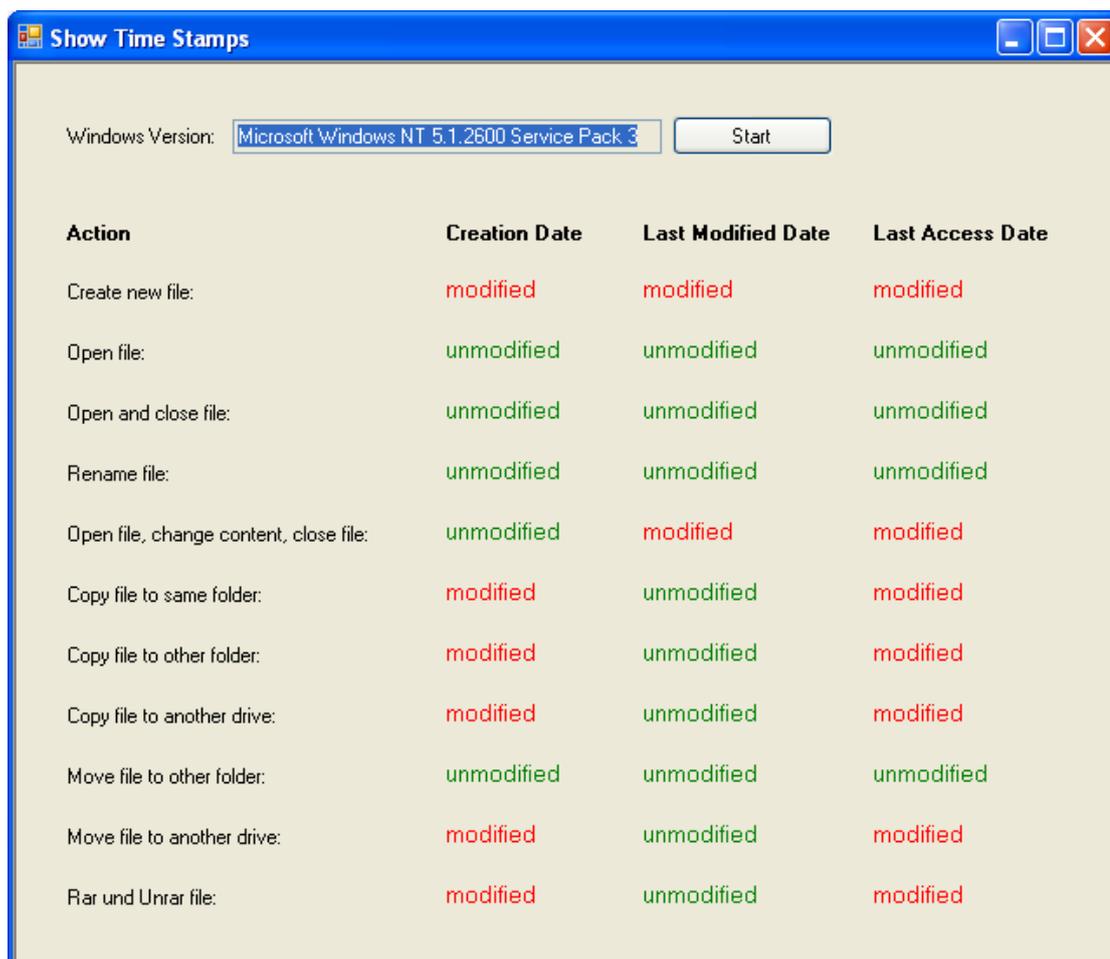


Abbildung 3.2: Ergebnis der Dateioperationen unter Windows XP

Windows XP			
	Erstelldatum	Bearbeitungsdatum	Letzer Zugriff
Neue Datei erstellen	M	M	M
Datei öffnen	U	U	M
Datei öffnen und schließen	U	U	M
Datei umbenennen	U	U	M
Datei öffnen, ändern und schließen	U	M	M
Datei in gleichen Ordner kopieren	M	U	M
Datei in anderen Ordner kopieren	M	U	M
Datei auf anderes Laufwerk kopieren	M	U	M
Datei in anderen Ordner verschieben	U	U	M
Datei auf anderes Laufwerk verschieben	U	U	M
Datei packen und entpacken (*.rar)	M	U	M

Abbildung 3.3: Ergebnis der Dateioperationen unter Windows XP - Manueller Test

3.3.1 Microsoft Windows XP

Wie man aus den beiden Abbildungen 3.2 und 3.3 erkennen kann, gibt es Unterschiede zwischen den händischen Tests und den Resultaten der implementierten Software.

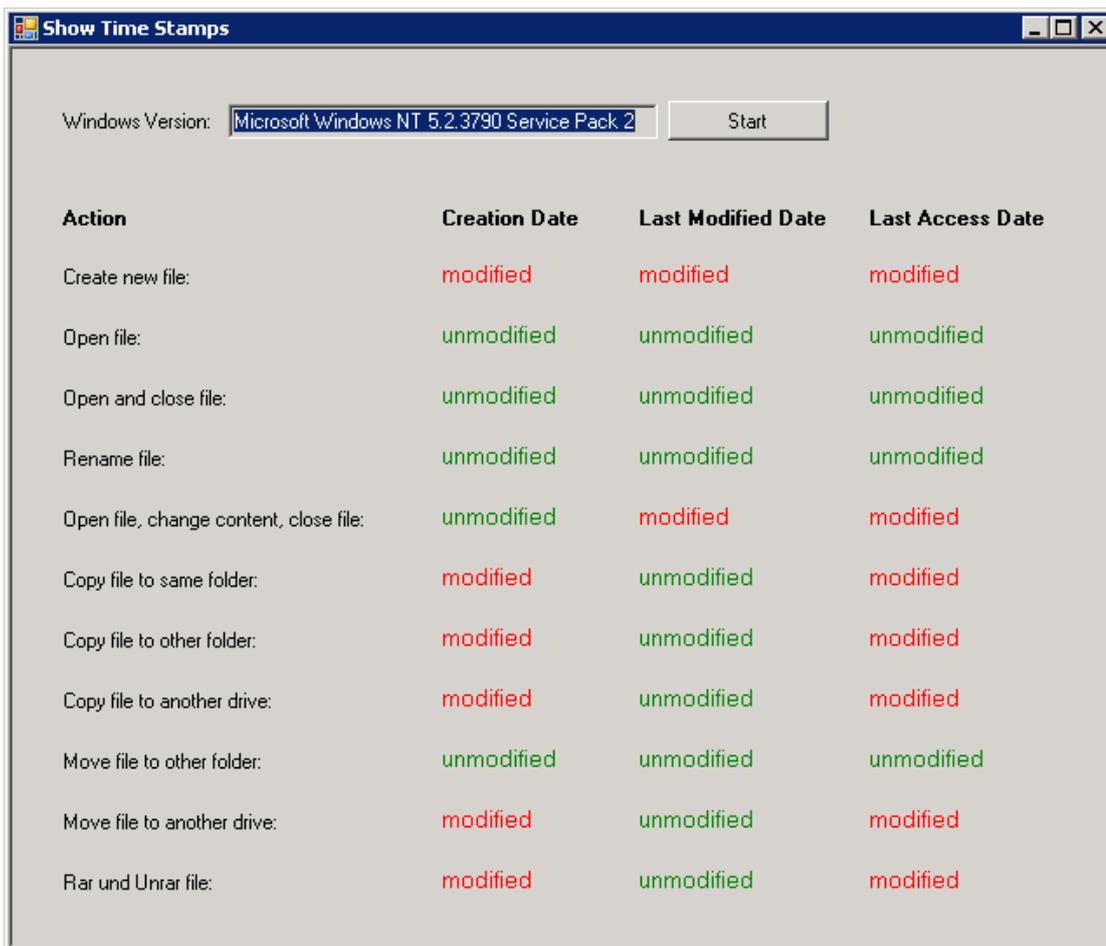
Das Öffnen einer Datei verändert unter *Windows XP* das Zugriffsdatum, allerdings nicht immer. Aufgrund eingehender Tests ist davon auszugehen, dass diese Änderung beim ersten lesenden Zugriff auf die Datei passiert. Der Grund dafür, dass der automatische Test ein anderes Ergebnis zeigt, ist vermutlich der, dass nur ein `FileStream` mit der Methode `File.OpenRead(filePath)` geöffnet wird und die Datei nicht mit einem externen Programm geöffnet wird.

Das Umbenennen und das Verschieben der Datei auf ein anderes Laufwerk weist unterschiedliche Ergebnisse im Zugriffsdatum auf. Für beide Tests wurde die Funktion `File.Move(oldPath, newPath)` aufgerufen, die offensichtlich keinen Einfluss auf das Datum des letzten Dateizugriffs hat.

Außerdem ist festzuhalten, dass sich beim Verschieben der Datei auf ein anderes Laufwerk mit der *MS-DOS-Eingabeaufforderung* zusätzlich das *Erstelldatum* ändert. Dieses Verhalten lässt sich auf der Benutzeroberfläche nicht reproduzieren.

Beim Packen und Entpacken der Datei mit *WinRAR* ist zu beobachten, dass sich das *Erstelldatum* nur ändert, wenn die Datei in ein neues Verzeichnis entpackt wird. Sollte beim Extrahieren eine vorhandene Datei (mit gleichem Namen) ersetzt werden, so ändert sich nur das Zugriffsdatum.

3.3.2 Microsoft Windows Server 2003



Action	Creation Date	Last Modified Date	Last Access Date
Create new file:	modified	modified	modified
Open file:	unmodified	unmodified	unmodified
Open and close file:	unmodified	unmodified	unmodified
Rename file:	unmodified	unmodified	unmodified
Open file, change content, close file:	unmodified	modified	modified
Copy file to same folder:	modified	unmodified	modified
Copy file to other folder:	modified	unmodified	modified
Copy file to another drive:	modified	unmodified	modified
Move file to other folder:	unmodified	unmodified	unmodified
Move file to another drive:	modified	unmodified	modified
Rar und Unrar file:	modified	unmodified	modified

Abbildung 3.4: Ergebnis der Dateioperationen unter Windows Server 2003

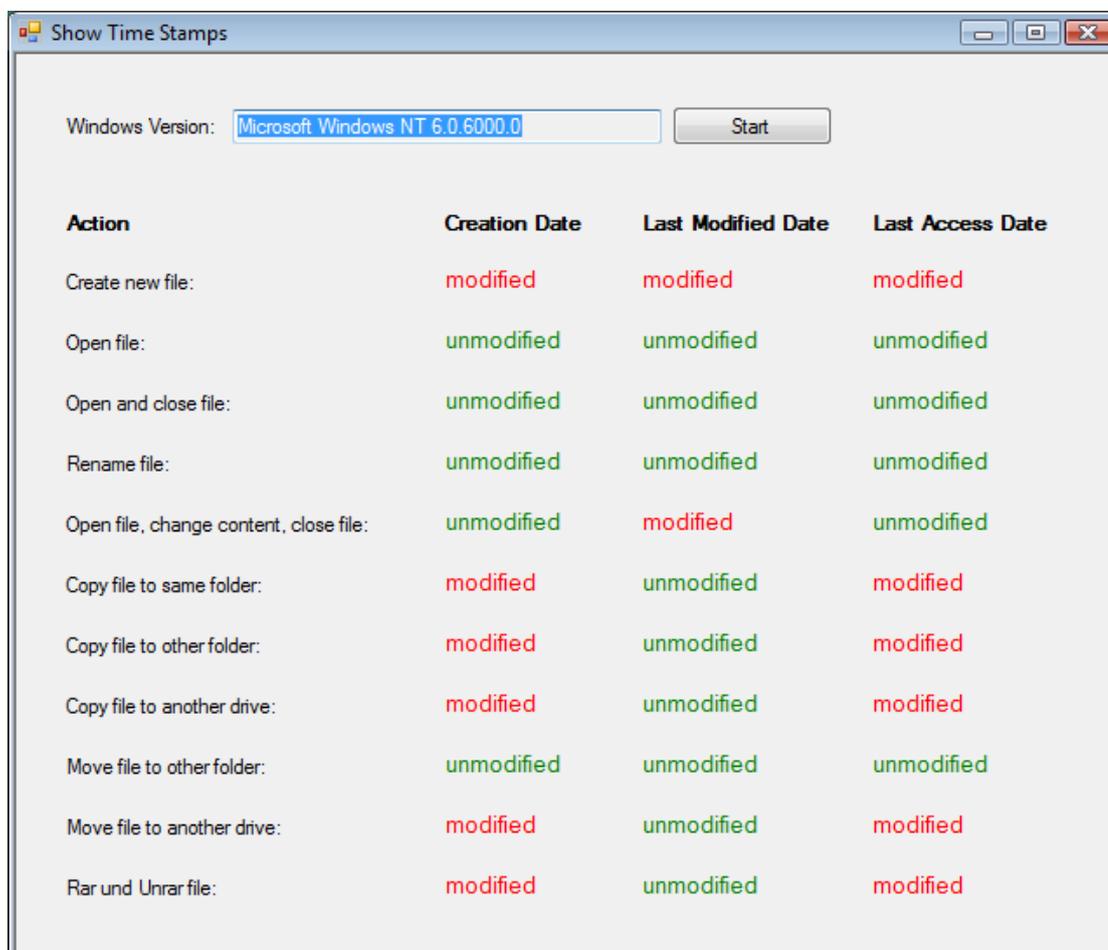
Windows Server 2003			
	Erstelldatum	Bearbeitungsdatum	Letzer Zugriff
Neue Datei erstellen	M	M	M
Datei öffnen	U	U	M
Datei öffnen und schließen	U	U	M
Datei umbenennen	U	U	M
Datei öffnen, ändern und schließen	U	M	M
Datei in gleichen Ordner kopieren	M	U	M
Datei in anderen Ordner kopieren	M	U	M
Datei auf anderes Laufwerk kopieren	M	U	M
Datei in anderen Ordner verschieben	U	U	M
Datei auf anderes Laufwerk verschieben	U	U	M
Datei packen und entpacken (*.rar)	M	U	M

Abbildung 3.5: Ergebnis der Dateioperationen unter Windows Server 2003 - Manueller Test

Wie in Abbildung 3.4 und 3.5 zu erkennen ist, verhält sich *Windows Server 2003* bei allen durchgeführten Tests exakt gleich wie *Windows XP*. Die Erklärungen für abweichende Resultate bei den verschiedenen Testvarianten sind ebenso die gleichen wie in Kapitel 3.3.1 auf Seite 20.

3.3.3 Microsoft Windows Vista

Die Abbildungen 3.6 und 3.7 zeigen die Resultate des automatischen und händischen Tests unter *Windows Vista*.



Action	Creation Date	Last Modified Date	Last Access Date
Create new file:	modified	modified	modified
Open file:	unmodified	unmodified	unmodified
Open and close file:	unmodified	unmodified	unmodified
Rename file:	unmodified	unmodified	unmodified
Open file, change content, close file:	unmodified	modified	unmodified
Copy file to same folder:	modified	unmodified	modified
Copy file to other folder:	modified	unmodified	modified
Copy file to another drive:	modified	unmodified	modified
Move file to other folder:	unmodified	unmodified	unmodified
Move file to another drive:	modified	unmodified	modified
Rar und Unrar file:	modified	unmodified	modified

Abbildung 3.6: Ergebnis der Dateioperationen unter Windows Vista

Ab *Windows Vista* ändert sich das Zugriffsdatum viel seltener als bei den zuvor beschriebenen Windows-Versionen. Weder beim Öffnen, Umbenennen oder Verschieben (am gleichen Laufwerk) sind Veränderungen dieses Datums zu beobachten.

Die einzige Anomalie tritt - wie bei *Windows XP* und *Windows Server 2003* - beim Verschieben auf ein anderes Laufwerk auf. Auch hier wird das Erstelldatum aktualisiert, wenn die Datei per Konsolenbefehl (`move`) auf ein anderes Laufwerk verschoben wird.

Beim Packen und Entpacken ist ein kleiner Unterschied zu den Vorgängerversionen zu beobachten und zwar insofern, als bei einem Extrahieren mit Ersetzen einer Datei

Windows Vista			
	Erstelldatum	Bearbeitungsdatum	Letzer Zugriff
Neue Datei erstellen	M	M	M
Datei öffnen	U	U	U
Datei öffnen und schließen	U	U	U
Datei umbenennen	U	U	U
Datei öffnen, ändern und schließen	U	M	U
Datei in gleichen Ordner kopieren	M	U	M
Datei in anderen Ordner kopieren	M	U	M
Datei auf anderes Laufwerk kopieren	M	U	M
Datei in anderen Ordner verschieben	U	U	U
Datei auf anderes Laufwerk verschieben	U	U	M
Datei packen und entpacken (*.rar)	M	U	M

Abbildung 3.7: Ergebnis der Dateioperationen unter Windows Vista - Manueller Test

durch die Datei im Archiv alle Zeitstempel übernommen werden, sich also kein Datum ändert.

3.3.4 Microsoft Windows Server 2008

Die Abbildungen 3.8 und 3.9 zeigen die Resultate des automatischen und händischen Tests unter *Windows Server 2008*.

Windows Server 2008 verhält sich bei allen Tests genauso wie *Windows Vista*.



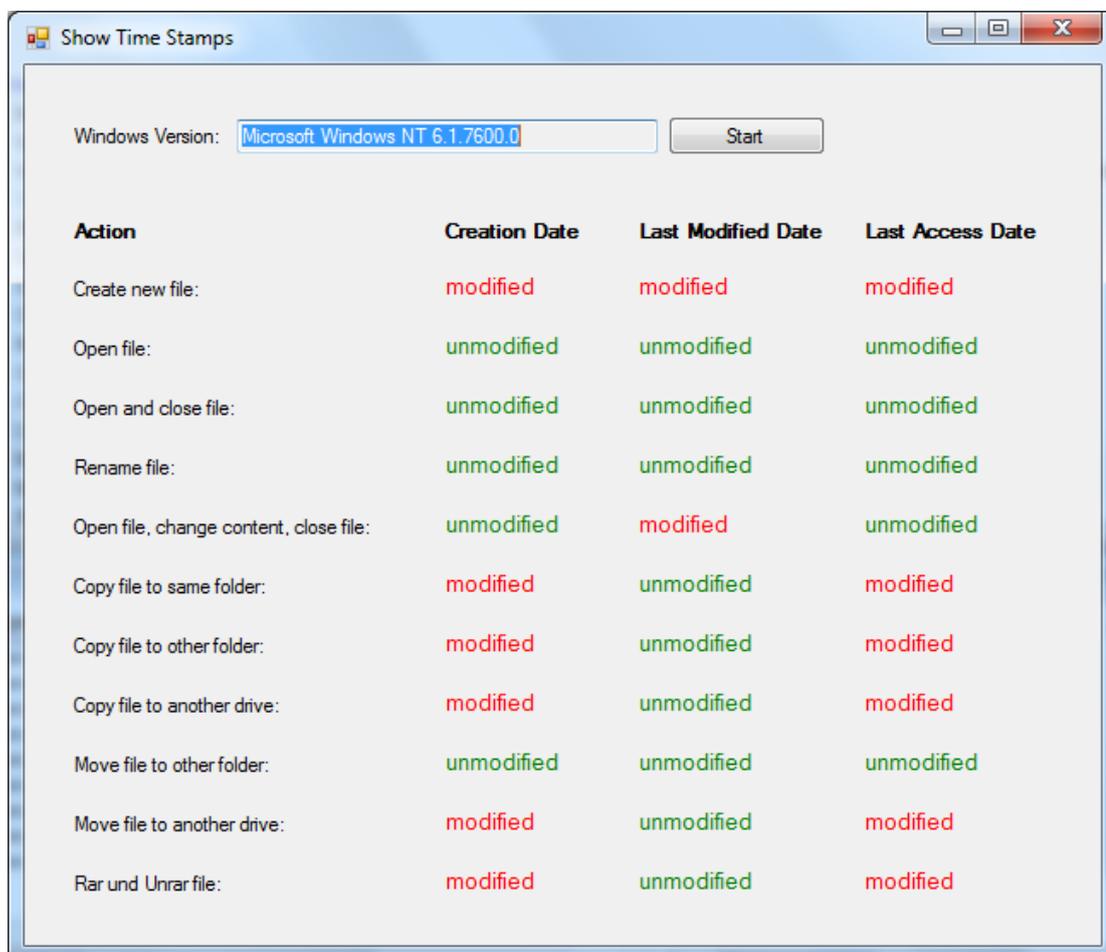
Abbildung 3.8: Ergebnis der Dateioperationen unter Windows Server 2008

Windows Server 2008			
	Erstelldatum	Bearbeitungsdatum	Letzer Zugriff
Neue Datei erstellen	M	M	M
Datei öffnen	U	U	U
Datei öffnen und schließen	U	U	U
Datei umbenennen	U	U	U
Datei öffnen, ändern und schließen	U	M	U
Datei in gleichen Ordner kopieren	M	U	M
Datei in anderen Ordner kopieren	M	U	M
Datei auf anderes Laufwerk kopieren	M	U	M
Datei in anderen Ordner verschieben	U	U	U
Datei auf anderes Laufwerk verschieben	U	U	M
Datei packen und entpacken (*.rar)	M	U	M

Abbildung 3.9: Ergebnis der Dateioperationen unter Windows Server 2008 - Manueller Test

3.3.5 Microsoft Windows 7

Die Abbildungen 3.10 und 3.11 zeigen die Resultate des automatischen und händischen Tests unter *Windows 7*.



Action	Creation Date	Last Modified Date	Last Access Date
Create new file:	modified	modified	modified
Open file:	unmodified	unmodified	unmodified
Open and close file:	unmodified	unmodified	unmodified
Rename file:	unmodified	unmodified	unmodified
Open file, change content, close file:	unmodified	modified	unmodified
Copy file to same folder:	modified	unmodified	modified
Copy file to other folder:	modified	unmodified	modified
Copy file to another drive:	modified	unmodified	modified
Move file to other folder:	unmodified	unmodified	unmodified
Move file to another drive:	modified	unmodified	modified
Rar und Unrar file:	modified	unmodified	modified

Abbildung 3.10: Ergebnis der Dateioperationen unter Windows 7

Auch *Windows 7* ist genau gleich implementiert wie *Windows Vista* und *Windows Server 2008*, wodurch keine weiteren Interpretationen der Versuchsergebnisse notwendig sind.

Windows 7			
	Erstelldatum	Bearbeitungsdatum	Letzer Zugriff
Neue Datei erstellen	M	M	M
Datei öffnen	U	U	U
Datei öffnen und schließen	U	U	U
Datei umbenennen	U	U	U
Datei öffnen, ändern und schließen	U	M	U
Datei in gleichen Ordner kopieren	M	U	M
Datei in anderen Ordner kopieren	M	U	M
Datei auf anderes Laufwerk kopieren	M	U	M
Datei in anderen Ordner verschieben	U	U	U
Datei auf anderes Laufwerk verschieben	U	U	M
Datei packen und entpacken (*.rar)	M	U	M

Abbildung 3.11: Ergebnis der Dateioperationen unter Windows 7 - Manueller Test

Literaturverzeichnis

- [1] Dustin Hurlbut, *Thumbs DB Files Forensic Issues*, Access Data Corporation, 2005

- [2] Christian Stobitzer, *Thumbs.db - Was sie ist und wofür man sie braucht*, Computer Blog, URL: <http://computer.meinwissen.info/thumbsdb-was-sie-ist-und-wofuer-man-sie-braucht>, Stand: 07.05.2010

- [3] Wikipedia, *Windows Thumbnail Cache*, URL: http://en.wikipedia.org/wiki/Windows_thumbnail_cache, Stand: 12.05.2010

- [4] Windows Power, *Wenn die Thumbs.db-Dateien zu viel Platz verbrauchen*, Windows Tipps Tricks Computer PC Hilfe, URL: http://www.windowspower.de/Wenn-die-Thumbs.db-%E2%80%93-Dateien-zu-viel-Platz-verbrauchen_840.html, Stand: 14.05.2010

- [5] TLab404, *How To Change Thumbnail Size and Quality?*, URL: <http://v2.tlab404.com/articles/detail.asp?iFaq=412&iType=13>, Stand: 05.05.2010

- [6] Markus Baumgartner, *Thumbs.db - eine platzfressende Datei beseitigen*, URL: <http://www.markusbaumi.ch/tipps/thumbs.html>, Stand: 10.05.2010

- [7] Forensics Wiki, *Vista Thumbcache*, URL: http://www.forensicswiki.org/wiki/Vista_thumbcache, Stand: 14.05.2010

- [8] GreenSpot Technologies Ltd, *dmThumbs*, URL: <http://www.dmthumbs.com/>, Stand: 16.05.2010

- [9] Wilders Security Forums, *Vistas Thumbcache.db Files*, URL: <http://www.wilderssecurity.com/showthread.php?t=224926>, Stand: 16.05.2010

- [10] Ben Vanik, *Vista Thumbnail Cache*, URL: <http://www.noxa.org/blog/?p=5>, Stand: 17.05.2010

- [11] SweetScape Software, *Home of 010 Editor*, URL: <http://www.sweetscape.com/>, Stand: 17.05.2010