

Microsoft Excel und Arbeitsspeicher

Inhaltsübersicht

- » [Einleitung und Vorwort](#)
 - [Einleitung](#)
 - [Vorwort](#)
- » [Die Excel-Spezifikation "Durch den verfügbaren Speicher begrenzt"](#)
- » [Speichernutzung von Windows](#)
 - [Speicherwerte mit einem Utility abfragen](#)
 - [Windows Utility "Windows Task-Manager"](#)
 - [Windows Utility "Microsoft Systeminfo"](#)
 - [Freeware Tool "FreeMem"](#)
 - [Weitere nützliche Programme](#)
 - [Speicherwerte mit VBA abfragen](#)
 - [Möglichkeiten in VBA](#)
 - [Memory Monitor mit VBA programmieren](#)
- » [Speicherbedarf von Excel](#)
 - [Empfohlene Grösse des Arbeitsspeichers](#)
 - [Windows 95/Windows 98](#)
 - [Windows ME/Windows NT](#)
 - [Windows 2000](#)
 - [Windows XP/Windows 2003](#)
 - [Nutzung der Speicherauslagerungsdatei](#)
- » [Struktur des von Excel verwendeten Speichers](#)
 - [Die sechs Speicherbereiche von Excel](#)
 - [Grösse der drei Speicherbereiche Program Memory, Workbook Data Memory und Workbook Memory](#)
 - [Excel 5.0](#)
 - [Excel 95/7.0](#)
 - [Excel 97](#)
 - [Excel 2000](#)
 - [Excel 2002/XP](#)
 - [Excel 2003](#)
 - [Excel V.12](#)
- » [Speichernutzung von Microsoft Excel](#)
 - [Beanspruchter Speicher von Excel \(Programm und Daten\)](#)
 - [Gesamt-Speichernutzung von Excel](#)
 - [Beanspruchter Speicher der xlb-Datei](#)
 - [Beanspruchter Speicher des VBA-Editors](#)
 - [Beanspruchter Speicher der Add-Ins](#)
 - [Verlauf der Speichernutzung während der Arbeit mit Excel](#)
- » [Speichernutzung der Daten in Excel](#)
 - [Das Speicher-Management von Excel](#)
 - [Das Data Memory von Excel](#)
 - [Das Workbook Memory von Excel](#)
 - [Zweck und Aufgabe des Workbook Memory](#)
 - [Grösse des Workbook Memory](#)
 - [Verfügbarer, benutzter und gesamter Speicher abfragen](#)
 - [Tabellenfunktion INFO](#)
 - [Application-Eigenschaften MemoryFree, MemoryUsed und MemoryTotal](#)
 - [Was die Speicherwerte bedeuten](#)
 - [Speicherbedarf einer Arbeitsmappe ermitteln](#)
- » [Der Undo-Speicher von Excel](#)
 - [Was ist der Undo-Speicher?](#)
 - [Undo-Speicher optimal verwenden](#)
 - [Anzahl Undo-Schritte reduzieren](#)
 - [Anzahl Undo-Schritte erhöhen](#)
 - [Undo-Speicher leeren](#)
 - [Undo-Speicher in Excel leeren](#)

[Undo-Speicher mit VBA leeren](#)

[Fehlermeldungen im Zusammenhang mit dem Undo-Speicher](#)

- » [Anwendungsfehler wegen Speicher-Bugs](#)
- » [Speicher-Fehlermeldungen beim Excel-Start](#)
- » [Speicher-Probleme wegen beschädigten Arbeitsblättern](#)
 - [Kurzbeschreibung des Problems](#)
 - [So erkennen Sie das Problem](#)
 - [So beheben Sie das Problem](#)
 - [Letzte Zelle eines Arbeitsblattes von Hand zurücksetzen](#)
 - [Letzte Zelle eines Arbeitsblattes mittels VBA-Makro zurücksetzen](#)
 - [Letzte Zelle eines Arbeitsblattes mittels Tool zurücksetzen](#)
- » [Speicher-Probleme wegen Speicher-Lecks \(Memory Leaks\)](#)
 - [Allgemeines über Speicher-Lecks](#)
 - [Speicher-Leck bei Arbeitsmappen mit eingebetteten Objekten](#)
 - [Weitere Speicher-Lecks in Microsoft Excel](#)
- » [Möglichkeiten zum Einsparen von Speicher](#)
 - [Allgemeines über das Einsparen von Speicher](#)
 - [Excel mehr Speicher zur Verfügung stellen](#)
 - [Excel-Speicher freigeben](#)
 - [Speicherbedarf von Arbeitsmappen reduzieren](#)
- » [Excel durch Belegen des gesamten Speichers zum Absturz bringen](#)
- » [VBA-Laufzeitfehler infolge ungenügendem Speicher](#)
 - [Allgemeines über VBA-Fehlermeldungen](#)
 - [Laufzeitfehler Nr. 7 "Nicht genügend Speicher"](#)
 - [Runtime Error No. 7 "Out of memory"](#)
 - [Laufzeitfehler Nr. 14 "Nicht genügend Zeichenfolgespeicher"](#)
 - [Runtime Error No. 14 "Out of string space"](#)
 - [Laufzeitfehler Nr. 28 "Nicht genügend Stapelspeicher"](#)
 - [Runtime Error No. 28 "Out of stack space"](#)
- » [Speichernutzung des VBA-Editors](#)
 - [Allgemeines über den VBA-Editor](#)
 - [Speicherbeanspruchung des VBA-Editors](#)
 - [VBA-Editor aus dem Speicher entfernen](#)
 - [Generierung der VBA-Projektstruktur \(Symbol Table\)](#)
 - [Entfernen der VBA-Projektstruktur](#)
 - [Speicherbeanspruchung der VBA-Projektstruktur](#)
 - [Dateigrösse mit und ohne VBA-Projektstruktur](#)
 - [Existenz einer VBA-Projektstruktur herausfinden](#)
 - [Kommunikation zwischen Excel und VBA-Editor](#)
- » [Speichernutzung von VBA-Programmen](#)
- » [Tipps und Tricks](#)
- » [Hilfsprogramme und Tools](#)
- » [Weitere Informationsquellen](#)
 - [Links zum Thema "Speicher"](#)
 - [Links zum Thema "Limitationen"](#)
 - [Links zum Thema "Bugs und Probleme"](#)



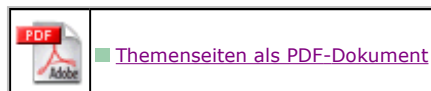
Einleitung und Vorwort

Einleitung

Auf dieser Internetseite erhalten Sie umfassende und detaillierte Informationen über Microsoft Excel und den Arbeitsspeicher. Der Inhalt der Seite ist wie ein Fachbuch aufgebaut: Die vorgestellten Themen sind in Form von Kapiteln zusammengefasst, und die Kapitel besitzen Abschnitte (Unterkapitel). Über das Inhaltsverzeichnis können die Kapitel und Abschnitte direkt aufgerufen werden. Infolge des Aufbaus als 'elektronisches' Fachbuch können Sie aber auch ganz einfach bei Kapitel 1 beginnend alle Kapitel nacheinander durchlesen. Der Text ist in einer allgemein verständlichen Sprache verfasst, sodass auch Excel-Benutzer mit nur wenig Erfahrung und geringem technischen Fachwissen verstehen werden, was gemeint ist. Alle Fach- und technischen Begriffe sowie Abkürzungen sind stets erläutert.

Kurz zusammengefasst wird auf dieser Internetseite erklärt, wie Microsoft Excel vom Arbeitsspeicher Gebrauch macht, wie es ihn verwaltet, welche Struktur der Speicher besitzt und welche Probleme, deren Lösungen, Fehlermeldungen, Optimierungsmöglichkeiten und weiteres existieren. Auch der vom VBA-Editor und von VBA-Programmen beanspruchter Speicher wird ausführlich vorgestellt.

Diese Internetseite ist auch als Acrobat PDF-Dokument erhältlich. Sie finden dieses und andere Dokumente auf dieser Seite:



Vorwort

Im Frühling des Jahres 2002 hatte ich das unbezwingbare Verlangen, mehr über die Speichernutzung von Microsoft Excel herausfinden zu müssen. Meine Arbeitsmappen wurden immer umfangreicher und komplexer, was zur Folge hatte, dass der Speicher während der Arbeit immer knapper wurde, des öfteren Fehlermeldungen wie "Nicht genügend Arbeitsspeicher" auftraten und Excel immer langsamer wurde. Ich unternahm daher mehrere Anläufe, um die Geheimnisse der Excel-Speichernutzung herauszufinden.

Nach dem Studium von unzähligen technischen Dokumentationen, tagelangen Internet-Recherchen, vielen durchgearbeiteten Nächten und ein mehreren hohen Pizza Kurier-Rechnungen hatte ich schliesslich genügend Informationen zusammengetragen. Diese konnte ich anschliessend kombinieren, sodass ein ziemlich umfassendes und detailliertes Bild des Speichermanagements von Microsoft Excel entstand. Eine grosse Herausforderung bestand darin, das Bild konsistent zu machen. Das war alles andere als einfach, da zu vielen Themen nur sehr spärlich Informationen erhältlich sind. So konnte ich einige Dinge bisher noch nicht herausfinden, beispielsweise wie der Speicher einer Pivot-Tabelle aufgebaut ist beziehungsweise welche Speicherbereiche beansprucht werden. Solche Informationen wären durchaus nützlich, da man bei deren Kenntnis eine Arbeitsmappe gezielt und noch besser optimieren könnte - und sich vor allem auch potenzielle Speicherprobleme vermeiden liessen.



TOP

Die Excel-Spezifikation "Durch den verfügbaren Speicher begrenzt"

Wenn man sich die in der Referenz/Online-Hilfe von Microsoft Excel aufgeführten Spezifikationen ein bisschen genauer anschaut, so findet man insgesamt 21 Merkmale, welche die Spezifikation "Durch den verfügbaren Speicher begrenzt" besitzen. Das ist bei allen gängigen Excelversionen, d.h. Excel 97, 2000, 2002 und 2003 der Fall.

Hier ein Ausschnitt des Hilfethemas "Angaben zur Arbeitsumgebung" aus der Online-Hilfe von Microsoft Excel 97:

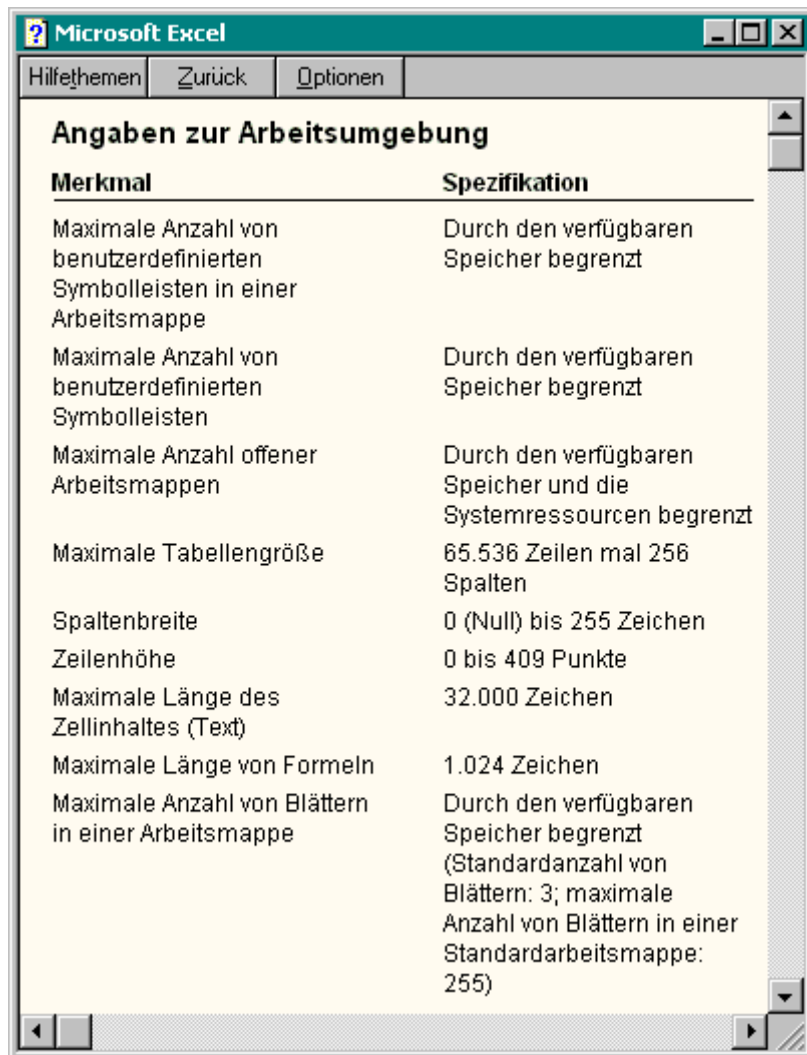


Abbildung: Angaben zur Arbeitsumgebung (Auszug aus der Excel 97-Online Hilfe)

Während meiner Tätigkeit als Berater und Trainer stellte ich des öfteren fest, dass unerwartet viele Benutzer - sowohl Einsteiger als auch Fortgeschrittene - die Spezifikation "Durch den verfügbaren Speicher begrenzt" nicht korrekt interpretieren. Excel-Anwender sind oft der Meinung, dass "Speicher" dasselbe ist wie "Arbeitsspeicher" (mit anderen Worten [RAM](#)), und dass "verfügbar" so viel wie "frei" bedeutet.

Aus diesem Grund sind folgende Annahmen bei den Usern weit verbreitet:

Annahme 1:	Wenn beim Öffnen oder Bearbeiten einer Arbeitsmappe die Fehlermeldung "Nicht genügend Arbeitsspeicher" auftritt, dann ist der Computer mit zu wenig RAM ausgerüstet.
Annahme 2:	Je mehr Arbeitsspeicher der Computer besitzt, desto grösser kann eine Excel-Arbeitsmappe sein.
Annahme 3:	Man kann Microsoft Excel mehr Arbeitsspeicher zur Verfügung stellen, indem man die Speicherauslagerungsdatei von Windows vergrössert.

Alle drei Annahmen sind leider falsch!

Wieso sie falsch sind, lässt sich am einfachsten anhand der Fehlermeldung "Nicht genügend Arbeitsspeicher" erläutern.

↑
TOP

Speichernutzung von Windows

Speicherwerte mit einem Utility abfragen

Die drei wichtigsten Kennzahlen 'gesamter Speicherplatz', 'beanspruchter Speicherplatz' und 'verfügbarer Speicherplatz' aller Anwendungen und Prozesse können in Windows sehr einfach abgefragt werden. Dazu reicht ein Standard-Utility wie beispielsweise der Task-Manager von Windows oder das Windows-Utility "Microsoft Systeminfo" (MSInfo32.exe). Der verfügbare Arbeitsspeicher für Programme lässt sich gut mit einem Tool wie beispielsweise "FreeMem" ermitteln.

Nachfolgend werden diese Tools kurz vorgestellt. Beachten Sie auch das Kapitel "[Hilfsprogramme und Tools](#)". Dort finden Sie weitere nützliche Tools und die Internetadressen, wo diese Programme heruntergeladen werden können.



Windows Utility "Windows Task-Manager"

Im Task-Manager von Windows NT wird wie in der nächsten Abbildung zu sehen ist der Gesamtspeicher unter "Realer Speicher (KB): Insgesamt" angezeigt, der verfügbare Speicher unter "Realer Speicher (KB): Verfügbar". Im Task-Manager von Windows XP ist dieser Fensterbereich mit "Physikalischer Speicher" angeschrieben.

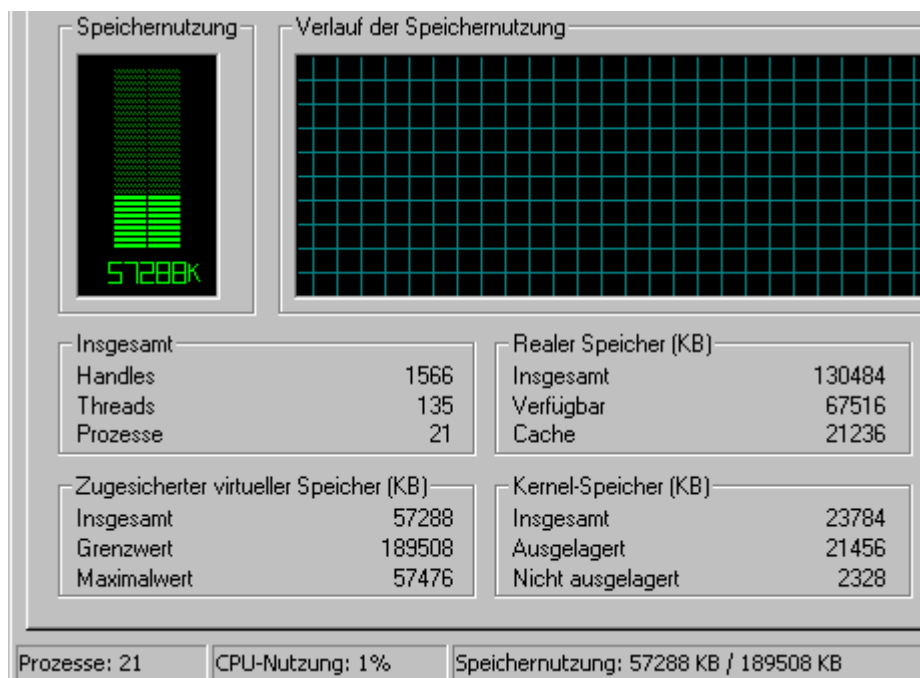


Abbildung: Kennzahlen zum Speicher (Windows NT Task-Manager)



Windows Utility "Microsoft Systeminfo"

Im Windows-Utility "Microsoft Systeminfo" wird der Gesamtspeicher unter "Gesamter physischer Speicher" angezeigt, der verfügbare Speicher unter "Verfügbarer physischer Speicher" (siehe nächste Abbildung). Je nach Windows-Version sieht die Oberfläche des Tools geringfügig anders aus.

Element	Wert
Betriebssystem:	Windows NT
Windows-Version:	4.0
Prozessor:	Pentium
Gesamter physischer Speicher:	130484 KB
Verfügbare physischer Speicher:	68624 KB*
Benutzer-Speicher:	90% frei
GDI-Speicher:	90% frei
Auslagerungsdatei:	189508 KB
Auslagerungsdatei-Nutzung:	34%
Auslagerungsdatei-Einstellungen:	Dynamisch
Verfügbare Speicher auf Laufwerk C:	17696 KB
Verfügbare Speicher auf Laufwerk D:	54033 KB
Windows-Ordner:	D:\WINNT
TEMP-Ordner:	D:\TEMP

* Siehe "Speicher" im Hilfeindex

Abbildung: Kennzahlen zum Speicher (Microsoft Systeminfo)

Freeware Tool "FreeMem"

Mit dem Freeware Tool "FreeMem" kann man den Verlauf des verfügbaren Arbeitsspeichers ansehen. Sie können natürlich auch ein anderes Tool einsetzen. Im Internet wird eine ganze Reihe vergleichbarer Programme angeboten, z.B. auf Freeware.de (www.freeware.de) oder Jumbo (www.jumbo.com).

Im Gegensatz zum Windows Task-Manager zeigt FreeMem nicht den Verlauf des **benutzten** Speichers sondern den Verlauf des **verfügbaren** Speichers, also genau die umgekehrte Darstellung. Zum Messen bzw. Kontrollieren der Speicherwerte eignet sich diese Darstellungsart besser (persönliche Meinung des Autors). FreeMem verfügt zudem über eine äusserst nützliche Funktion, die insbesondere beim Erforschen des Excelspeichers eingesetzt wird. Sie können den Arbeitsspeicher defragmentieren, indem der Speicherinhalt auf die Festplatte geschrieben und wieder eingelesen wird. Dadurch werden einerseits die im Speicher - beispielsweise infolge eines Programmabsturzes - liegengeliebene Daten gelöscht und andererseits fragmentierte Speicherblöcke eliminiert. Als Folge wird der zur Verfügung stehende Arbeitsspeicher grösser (zum Teil um mehrere MB).

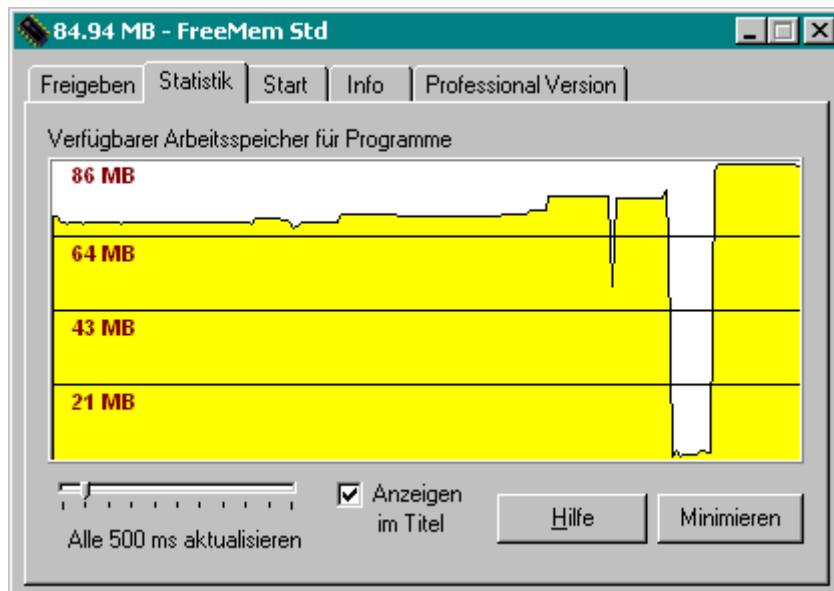


Abbildung: Verlauf des verfügbaren Speichers (Freeware Tool 'FreeMem')

Anmerkung

Die obige Abbildung zeigt die Freeware Edition von FreeMem Version 4.3, welche ich seit rund 4 Jahren einsetze. Inzwischen ist die Version 5 erhältlich, welche allerdings nicht mehr als Freeware-Programm zu haben ist.

Die Webseite mit Produkt-Informationen und einer Download-Möglichkeit finden Sie hier: www.meikel.com/de/products/freemem



Weitere nützliche Programme

Eine Liste mit weiteren interessanten Tools finden Sie hier:



Speicherwerte mit VBA abfragen

Möglichkeiten in VBA

Auch mit VBA ist es sehr einfach, verschiedene Speicherwerte zu ermitteln. Am einfachsten verwendet man dazu die Windows API-Prozedur *GlobalMemoryStatus*, die sich in der Kernel32.dll befindet. Beim Prozeduraufruf muss lediglich eine Datenstruktur übergeben werden, in welche dann die Speicherwerte abgefüllt werden. Dazu wird mittels *Type-End Type* ein benutzerdefinierter Datentyp definiert (MEMORYSTATUS), der als Prozedur-Argument *lpBuffer* verwendet wird.

Wie der benutzerdefinierte Datentyp deklariert wird und wie die Speicherwerte schliesslich ausgelesen und angezeigt werden, zeigt das nachfolgende kleine VBA-Beispiel.



Memory Monitor mit VBA programmieren

In diesem VBA-Beispiel wird ein kleiner Memory Monitor programmiert. Der Memory Monitor besitzt ein Benutzerformular, auf dem die beim Programmstart ermittelten Werte bezüglich

- Memory Load (Speicherauslastung),
 - Physical Memory (Physischer Speicher gesamt und verfügbar),
 - Page File (Speicherauslagerungsdatei gesamt und verfügbar) und
 - Virtual Memory (Virtueller Speicher gesamt und verfügbar)
- ausgegeben werden.

Das Benutzerformular des fertigen Memory Monitors sieht so aus:

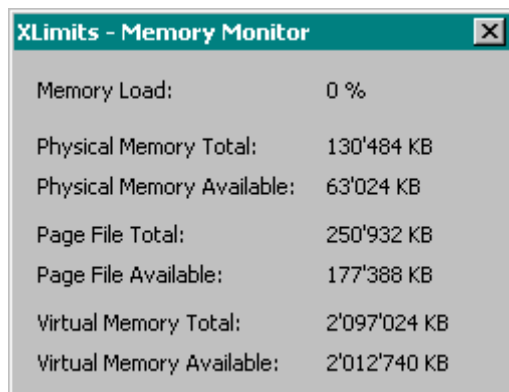


Abbildung: Dialogfenster eines selbstentwickelten Memory Monitors

Legen Sie in Microsoft Excel eine neue Arbeitsmappe an und öffnen dann den VBA-Editor. Fügen Sie dem VBA-Projekt ein neues Benutzerformular hinzu und kopieren Sie den nachstehenden Programmcode in das Codefenster des Benutzerformulars.

```
VBA-Code  
Private Type MEMORYSTATUS
```

```
dwLength As Long
dwMemoryLoad As Long
dwTotalPhys As Long
dwAvailPhys As Long
dwTotalPageFile As Long
dwAvailPageFile As Long
dwTotalVirtual As Long
dwAvailVirtual As Long
End Type
Private Declare Sub GlobalMemoryStatus Lib "kernel32" (lpBuffer As MEMORYSTATUS)

Private Sub UserForm_Initialize()
Dim MS As MEMORYSTATUS
MS.dwLength = Len(MS)
GlobalMemoryStatus MS
lblMemoryLoad.Caption = "Memory Load:" & Chr$(9) & Chr$(9) & _
Format$(MS.dwMemoryLoad, "###0") & " %"
lblTotalPhys.Caption = "Physical Memory Total:" & Chr$(9) & _
Format$(MS.dwTotalPhys / 1024, "#,##0") & " KB"
lblAvailPhys.Caption = "Physical Memory Available:" & Chr$(9) & _
Format$(MS.dwAvailPhys / 1024, "#,##0") & " KB"
lblTotalPageFile.Caption = "Page File Total:" & Chr$(9) & Chr$(9) & _
Format$(MS.dwTotalPageFile / 1024, "#,##0") & " KB"
lblAvailPageFile.Caption = "Page File Available:" & Chr$(9) & Chr$(9) & _
Format$(MS.dwAvailPageFile / 1024, "#,##0") & " KB"
lblTotalVirtual.Caption = "Virtual Memory Total:" & Chr$(9) & _
Format$(MS.dwTotalVirtual / 1024, "#,##0") & " KB"
lblAvailVirtual.Caption = "Virtual Memory Available:" & Chr$(9) & _
Format$(MS.dwAvailVirtual / 1024, "#,##0") & " KB"
End Sub
```

Dem Benutzerformular müssen noch 7 Bezeichnungsfelder (Labels) hinzugefügt werden, damit die einzelnen Speicherwerte angezeigt werden können. Die Namen der Bezeichnungsfelder lauten wie folgt (siehe auch VBA-Code):

- lblMemoryLoad
- lblTotalPhys
- lblAvailPhys
- lblTotalPageFile
- lblAvailPageFile
- lblTotalVirtual
- lblAvailVirtual

Wenn Sie den Programmcode dann ausführen, so wird das weiter oben abgebildete Benutzerformular eingeblendet und die aktuellen Speicherwerte angezeigt.

Quelle: Sam Huggill (Anpassungen durch Philipp von Wartburg)

 TOP

Speicherbedarf von Excel

Empfohlene Grösse des Arbeitsspeichers

Je nach Windows-Version wird unterschiedlich viel Arbeitsspeicher benötigt, damit Sie flüssig mit Microsoft Excel arbeiten können. Die vier nachfolgenden Tabellen zeigen Empfehlungen, wie gross der Arbeitsspeicher sein sollte, und zwar für die Bearbeitung von kleinen, mittelgrossen, grossen und sehr grossen Arbeitsmappen. Gelegentliche bzw. Durchschnitts-Anwender können die Angaben für mittelgrosse Arbeitsmappe verwenden (**grün** markiert). Der Speicherbedarf von Windows und Excel selbst ist in der empfohlenen Arbeitsspeichergösse bereits berücksichtigt. Die Angaben gelten für die gängigen Excelversionen 97, 2000, 2002/XP und 2003 gleichermassen.

Bitte beachten Sie, dass es sich hier um Empfehlungen und nicht um zwingende Voraussetzungen handelt. Die angegebenen Zahlen basieren auf Erfahrungswerten, auf Informationen von Microsoft und auf Empfehlungen der Firma Decision Models (www.decisionmodels.com).

Lesebeispiel 1

Sie verwenden Windows 98 und arbeiten mit mittelgrossen Arbeitsmappen:

» Der Arbeitsspeicher sollte nicht kleiner als 64 MB sein.

Lesebeispiel 2

Sie verwenden Windows NT und arbeiten mit grossen Arbeitsmappen:

» Der Arbeitsspeicher sollte nicht kleiner als 136 MB sein; besser wären 160 MB.

Lesebeispiel 3

Sie verwenden Windows 2000 und arbeiten mit sehr grossen Arbeitsmappen:

» Der Arbeitsspeicher sollte nicht kleiner als 304 MB sein; besser wären 320 MB.

Windows 95/Windows 98

Merkmal	Speicherbedarf
Windows (Betriebssystem)	24 MB
Excel (Anwendung)	6-12 MB
Kleine Arbeitsmappen	32 MB
Mittelgrosse Arbeitsmappen	64 MB
Grosse Arbeitsmappen	128 MB
Sehr grosse Arbeitsmappen	256 MB

Windows ME/Windows NT

Merkmal	Speicherbedarf
Windows (Betriebssystem)	32 MB
Excel (Anwendung)	6-12 MB
Kleine Arbeitsmappen	40 MB / 64 MB
Mittelgrosse Arbeitsmappen	72 MB / 96 MB
Grosse Arbeitsmappen	136 MB / 160 MB
Sehr grosse Arbeitsmappen	264 MB / 288 MB

Windows 2000

Merkmal	Speicherbedarf
Windows (Betriebssystem)	64 MB
Excel (Anwendung)	6-12 MB
Kleine Arbeitsmappen	80 MB / 96 MB
Mittelgrosse Arbeitsmappen	112 MB / 128 MB
Grosse Arbeitsmappen	176 MB / 192 MB
Sehr grosse Arbeitsmappen	304 MB / 320 MB

Windows XP/Windows 2003

Merkmal	Speicherbedarf
Windows (Betriebssystem)	128 MB
Excel (Anwendung)	6-12 MB
Kleine Arbeitsmappen	128 MB / 160 MB
Mittelgrosse Arbeitsmappen	160 MB / 192 MB
Grosse Arbeitsmappen	224 MB / 256 MB
Sehr grosse Arbeitsmappen	352 MB / 384 MB

Wenn Sie vorhaben, mehrere grössere Arbeitsmappen gleichzeitig in der gleichen Excel-Anwendung (Instanz) zu bearbeiten, so ist der Speicherbedarf (logischerweise) höher. Allerdings beträgt der Speicherbedarf bei mehreren Mappen nicht zwangsläufig das Mehrfache einer Mappe. Beispielsweise benötigen drei grosse Arbeitsmappen eher

selten dreimal so viel Speicher wie eine einzige grosse Mappe. Oft ist es so, dass nur während einer Neuberechnung sehr viel Speicher beansprucht wird. Beachten Sie, dass bei einer Neuberechnung immer sämtliche in der gleichen Excel-Instanz geöffneten Arbeitsmappen mit einbezogen werden. Sie können dies umgehen, indem Sie jede Arbeitsmappe in einer separaten Excel-Instanz öffnen. Dadurch wird zwar die Anwendung Microsoft Excel mehrmals parallel ausgeführt. Es hat aber den Vorteil, dass immer nur eine Arbeitsmappe pro Instanz geöffnet ist, und diese Mappe somit den gesamten von Excel benutzbaren Arbeitsspeicher gewissermassen für sich alleine zur Verfügung hat.

Nutzung der Speicherauslagerungsdatei

Wenn Sie insbesondere beim Öffnen und Berechnen einer Arbeitsmappe feststellen, dass die Speicherauslagerungsdatei intensiv genutzt wird, so ist dies ein Anzeichen für zu wenig physischen Arbeitsspeicher (RAM). Falls dies häufig bei mehreren Arbeitsmappen vorkommt, sollten Sie ernsthaft die Anschaffung zusätzlicher Speichermodule in Betracht ziehen, um damit Ihren Computer aufzurüsten.

Die folgende Abbildung zeigt eine Situation, in der die Speichernutzung durch Microsoft Excel sehr hoch ist:

Name	PID	CPU-Nutzung (%)	Speichernutzung	Virtueller Speicher	Basispri...
Leerlaufprozeß	0	98	16 KB	0 KB	n. zutr.
System	2	00	200 KB	36 KB	Normal
SMSS.EXE	20	00	0 KB	168 KB	Hoch
CSRSS.EXE	24	00	200 KB	1060 KB	Hoch
WINLOGON.EXE	34	00	200 KB	888 KB	Hoch
SERVICES.EXE	40	00	28 KB	1240 KB	Normal
LSASS.EXE	43	00	0 KB	1288 KB	Normal
SPOOLSS.EXE	88	00	32 KB	2796 KB	Normal
RPCSS.EXE	98	00	264 KB	712 KB	Normal
TCPSVCS.EXE	103	00	188 KB	488 KB	Normal
TAPISRV.EXE	112	00	28 KB	944 KB	Normal
PSTORES.EXE	123	00	0 KB	1568 KB	Normal
NDDEAGNT.EXE	125	00	0 KB	300 KB	Normal
RASMAN.EXE	130	00	440 KB	1028 KB	Normal
mstask.exe	134	00	200 KB	796 KB	Normal
EXPLORER.EXE	145	00	1464 KB	3588 KB	Normal
AcroTray.exe	147	00	0 KB	244 KB	Normal
CLICKER.EXE	149	00	0 KB	584 KB	Normal
TASKMGR.EXE	172	02	1056 KB	492 KB	Hoch
Calc.exe	175	00	0 KB	512 KB	Normal
fpeditor.exe	179	00	10436 KB	6344 KB	Normal
EXCEL.EXE	181	00	81980 KB	152172 KB	Normal
InCD.exe	185	00	652 KB	1380 KB	Normal
freemem.exe	189	00	1032 KB	804 KB	Normal

Prozesse: 24 CPU-Nutzung: 4% Speichernutzung: 233228 KB / 250932 KB

Abbildung: Speichernutzung von Prozessen (Windows Task-Manager)

Die blau markierte Spalte zeigt die Belegung des physischen Speichers der laufenden Prozesse, die rote Spalte den virtuellen Speicher (Auslagerungsdatei). Es ist gut zu erkennen, dass Windows bereits intensiv Speicher von mehreren Prozessen ausgelagert hat. Sowohl Systemprozesse wie unter anderem SMSS.EXE und LSASS.EXE als auch Anwendungen wie CLICKER.EXE (ein kleines Utility zum Programmieren der mittleren Maustaste) und CALC.EXE (das ist der Windows Taschenrechner) befinden sich vollständig im virtuellen Speicher, nur damit Microsoft Excel (EXCEL.EXE) möglichst viel physischer Speicher zur Verfügung gestellt werden kann. Wie in der

Abbildung zu sehen ist, benutzt Excel 81'980 KB physischen Speicher. Aber auch virtueller Speicher wird von Excel stark benutzt, nämlich 152'172 KB. Insgesamt belegt Excel somit rund 234 MB Speicher. Die mit brauner Farbe eingerahmte Information "Speichernutzung: 233228 KB / 250932 KB" zeigt, wieviel Speicher total belegt ist.

Informationen über den Zusammenhang zwischen Dateigrösse und Speicherbedarf finden Sie hier:



▲
TOP

Struktur des von Excel verwendeten Speichers

Grundsätzlich benötigt jedes auf einem Computer ausgeführte Programm eine bestimmte Menge Arbeitsspeicher. Wenn man beispielsweise den Windows Editor (Notepad) ausführt, so belegt er unmittelbar nach dem Programmstart ungefähr 1.3 MB Speicher. Dies, weil die ausführbare Datei (Notepad.exe) des Editors, welche den Programmcode enthält, geladen werden muss. Bei Microsoft Excel läuft das gleiche ab: Die Datei Excel.exe wird in den Speicher geladen und belegt dadurch ca. 7 MB Speicherplatz (je nach Excelversion verschieden).

Wenn man den Speicher etwas genauer betrachtet, handelt es sich nicht um einen einzigen grossen Speicherbereich sondern um mehrere Bereiche.

▲

Die sechs Speicherbereiche von Excel

Man kann den von Microsoft Excel beanspruchten Speicher in sechs Bereiche unterteilen:

- Speicher für **Arbeitsmappen-Daten** (Workbook Data Memory)
- Speicher für **Verarbeitung/Bearbeitung der Arbeitsmappen-Daten** ([Workbook Memory](#))
- Speicher für **Allgemeine Daten** (Data Memory)
- Speicher für **VBA-Programme** (VBA Memory)
- Speicher für den **VBA-Editor** (VBE Memory)
- Speicher für die **Excel-Anwendung** (Program Memory)

Die nachfolgende Abbildung zeigt diese Speicherbereiche in einer vereinfachten Darstellung als sechs Blöcke:

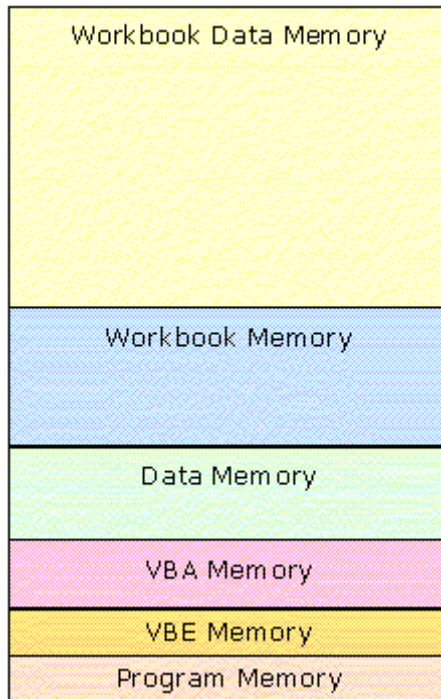


Abbildung: Die Bereiche des Excel-Speichers

-> Data Memory: Excel-Optionen, Arbeitsmappen-Optionen, Druckvorschau, Pivot-Cache (?), Dokument-Eigenschaften, Suchen-Begriff etc.

-> VBA Memory: VBA-Programme

-> VBE Memory: VBE-Optionen, Undo-Aktionen des VBA-Editors

Obwohl diese vereinfachte Darstellung nicht sonderlich viel aussagt, ist dennoch erkennbar, dass das Program Memory kleiner als das Workbook Memory ist, welches seinerseits kleiner als das Workbook Data Memory ist.

Die nächste Abbildung zeigt eine schematische Darstellung des Windows-Arbeitsspeicher (grosses schwarzes Rechteck), in welchem sich eine laufende Excel-Anwendung befindet (blaues Rechteck). Die drei oben vorgestellte Bereiche sind ebenfalls zu sehen (gelbe, blaue und grüne Fläche im blauen Rechteck):

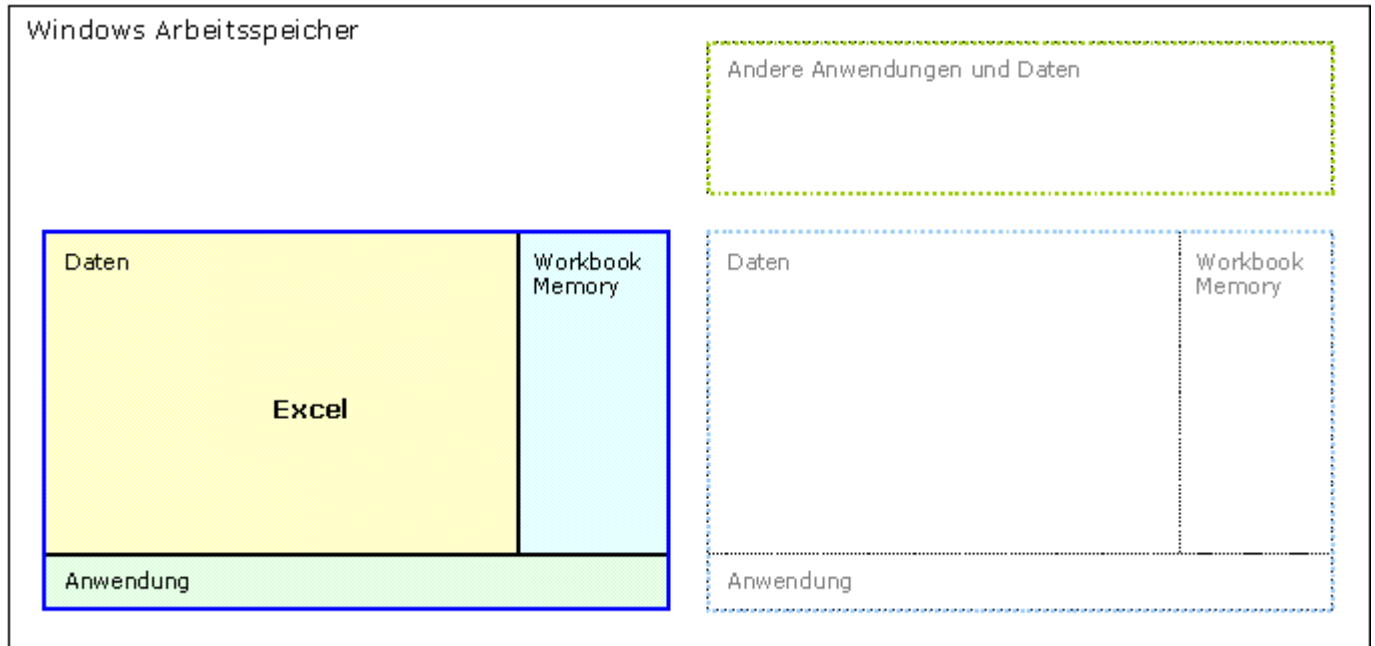


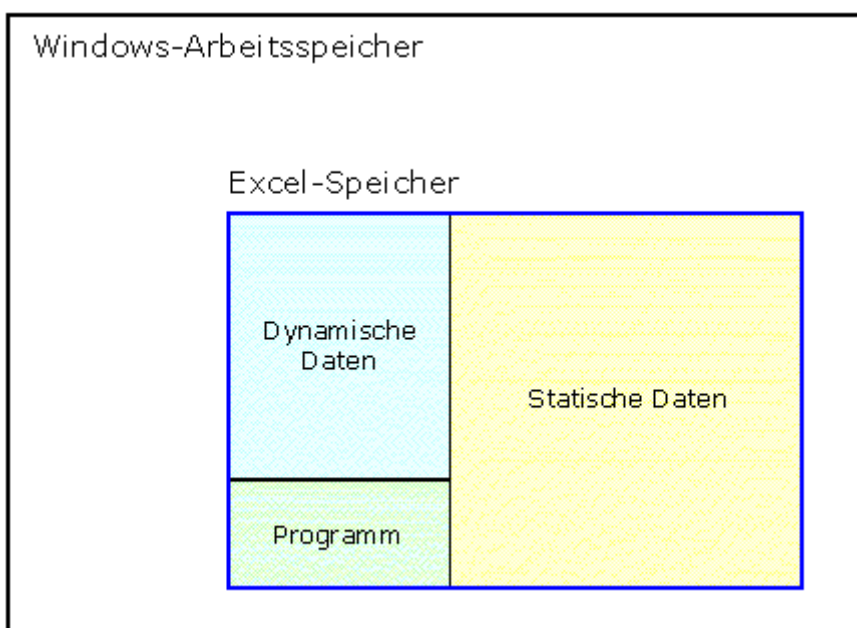
Abbildung: Microsoft Excel im Arbeitsspeicher von Windows

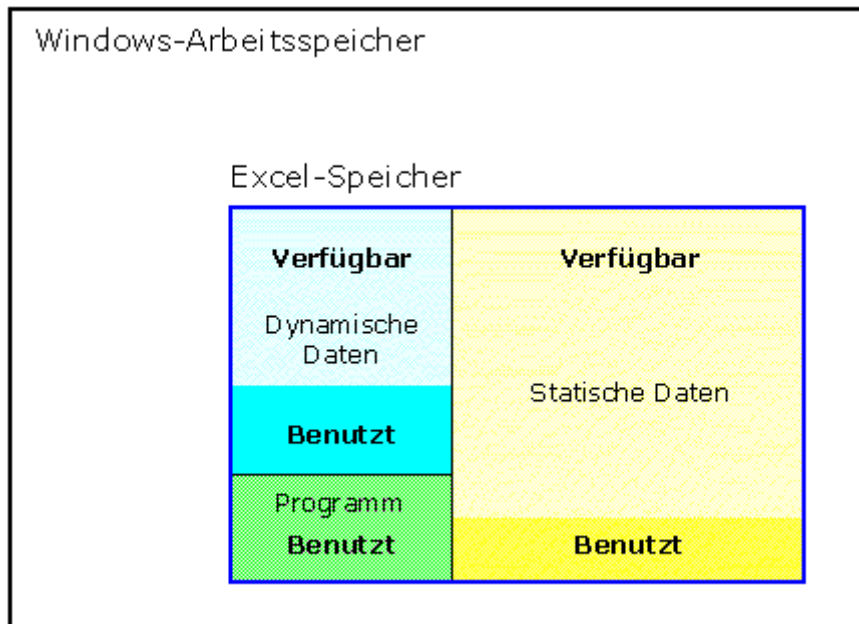
Das hellblaue Rechteck rechts davon stellt eine zweite laufende Excel-Instanz an und darüber befindet sich ein weiteres Rechteck, das mit "Andere Anwendungen und Daten" angeschrieben ist. Diese beiden Rechtecke müssen Sie nicht gross beachten.

Wie in der Abbildung auf jeden Fall gut zu erkennen ist, befindet sich das blaue Rechteck (Excel-Speicher) im schwarzen Rechteck (Windows-Arbeitsspeicher) und ist somit auch kleiner als das schwarze Rechteck. Das hellblau ausgefüllte Rechteck, welches das Workbook Memory darstellt und sich im blauen Rechteck - dem Excel-Speicher - befindet, ist somit zwangsläufig kleiner als der Windows-Arbeitsspeicher.

So simpel und logisch sich das auch anhören mag. Diese Erkenntnisse sind äusserst wichtig - vor allem wenn Sie mit grossen und komplexen Arbeitsmappen arbeiten. Fehlermeldungen infolge zu wenig verfügbarem Speicher treten oft auf, weil der Bereich "Workbook Memory" vollständig belegt ist.

Gehen wir einen Schritt weiter. Die nächsten beiden Abbildungen zeigen wiederum die drei Bereiche, wobei der blaue Bereich für das Workbook Memory hier mit "Dynamische Daten" und der gelbe Bereich für das Data Memory mit "Statische Daten" angeschrieben ist.





Unterschied zwischen Workbook Data Memory und Workbook Memory

Es ist nicht immer einfach zu erklären, weshalb es einen Speicherbereich für die Arbeitsmappendaten (Workbook Data Memory) gibt und einen Speicherbereich für die Verarbeitung/Bearbeitung der Arbeitsmappendaten (Workbook Memory). Die Frage ist, warum nicht ein einziger Speicherbereich für beides verwendet wird, beziehungsweise ob die Aufteilung in zwei Bereiche tatsächlich von Bedeutung ist.

Experiment

Schritt 1: Merken Sie sich folgende fünf Zahlen:

12, 4, 17, 23, 9

Nachdem Sie sich die Zahlen gemerkt haben, sind diese gewissermassen in Ihrem Gedächtnis abgelegt - oder anders ausgedrückt in Ihrem 'Speicher' eingetragen. Bei Excel passiert eigentlich genau das gleiche: Sie tragen die fünf Zahlen in Zellen ein, oder es wird eine Arbeitsmappe geöffnet, die diese fünf Zahlen enthält.

Schritt 2: Addieren Sie nun die fünf Zahlen und merken Sie sich das Ergebnis.

Das Ergebnis müsste 65 lauten.

Unterschied zwischen Workbook Data Memory und Data Memory

Im Data Memory sind alle Daten gespeichert,



Grösse der drei Speicherbereiche Program Memory, Workbook Data Memory und Workbook Memory

Jeder der drei Speicherbereiche Program Memory, Workbook Data Memory und Workbook Memory besitzt eine bestimmte bzw. eine maximale Grösse.

Hier eine Übersicht dieser Speicherbereiche für Microsoft Excel 97, 2000, 2002/XP und 2003:

- Die Grösse von "Program Memory" zeigt den von Microsoft Excel als Programm benötigten Speicherplatz.
- Die Grösse von "Workbook Data Memory" zeigt den von den Daten benötigten Speicherplatz.
- Die Grösse von "Workbook Memory" zeigt den für die Datenverarbeitung benötigten Speicherplatz.

Excel 5.0

Speicherbereich	Speicher
Ungefähre Grösse von "Program Memory"	(Nicht bekannt)
Maximale Grösse von "Workbook Data Memory"	(Nicht bekannt)
Maximale Grösse von "Workbook Memory"	16 MB

Excel 95/7.0

Speicherbereich	Speicher
Ungefähre Grösse von "Program Memory"	(Nicht bekannt)
Maximale Grösse von "Workbook Data Memory"	(Nicht bekannt)
Maximale Grösse von "Workbook Memory"	64 MB

Excel 97

Speicherbereich	Speicher
Ungefähre Grösse von "Program Memory"	ca. 6 MB
Maximale Grösse von "Workbook Data Memory"	1'000 MB (1 GB)
Maximale Grösse von "Workbook Memory"	64 MB

Excel 2000

Speicherbereich	Speicher
Ungefähre Grösse von "Program Memory"	ca. 8 MB
Maximale Grösse von "Workbook Data Memory"	1'000 MB (1 GB)
Maximale Grösse von "Workbook Memory"	64 MB

Excel 2002/XP

Speicherbereich	Speicher
Ungefähre Grösse von "Program Memory"	ca. 10 MB
Maximale Grösse von "Workbook Data Memory"	1'000 MB (1 GB)
Maximale Grösse von "Workbook Memory"	128 MB

Excel 2003

Speicherbereich	Speicher
Ungefähre Grösse von "Program Memory"	ca. 12 MB
Maximale Grösse von "Workbook Data Memory"	1'000 MB (1 GB)
Maximale Grösse von "Workbook Memory"	1'000 MB (1 GB)

Excel V.12

Speicherbereich	Speicher
Ungefähre Grösse von "Program Memory"	(Nicht bekannt)
Maximale Grösse von "Workbook Data Memory"	Unbegrenzt
Maximale Grösse von "Workbook Memory"	Unbegrenzt

▲
TOP

Speichernutzung von Microsoft Excel

Beanspruchter Speicher von Excel (Programm und Daten)

Wer sich schon mal etwas intensiver mit der Speichernutzung von Microsoft Excel beschäftigt hat, wird die eine oder andere interessante (und auch unverständliche) Entdeckung gemacht haben. Nachfolgend ein paar Dinge, die mir bei verschiedenen Experimenten mit Microsoft Excel 97 aufgefallen sind.

▲

Gesamt-Speichernutzung von Excel

Der von Microsoft Excel insgesamt beanspruchte Speicher wird am einfachsten anhand des [Windows Task-Managers](#) ermittelt.

Auf der Registerseite "Prozesse" des Task-Managers kann die Speichernutzung der momentan ausgeführten Prozesse (bzw. Anwendungen) abgelesen werden. Die nächste Abbildung zeigt einen Ausschnitt der Prozessliste mit der Speichernutzung der Anwendung EXCEL.EXE (11'464 KB) und den durch EXCEL.EXE belegten [virtuellen Speicher](#) (3'548 KB).

			Speichernutzung		Virtueller Speicher		
Prozessname	PID	Speicher	Physisch	Virtual	Private	Shared	
EXPLORER.EXE	89	00	2712 KB	2268 KB	65	5	
EXCEL.EXE	173	00	11464 KB	3548 KB	117	4	
CSRSS.EXE	24	00	200 KB	1064 KB	190	6	

Abbildung: Nutzung des Speichers und des virtuellen Speichers von EXCEL.EXE (Ausschnitt des Task-Managers)

Wenn Sie die Werte auf Ihrer eigenen Arbeitsstation prüfen möchten, aber die Spalten "Speichernutzung" und "Virtueller Speicher" nicht zu sehen sind, müssen Sie diese zuerst über den Menübefehl *Ansicht/Spalten wählen* einblenden.

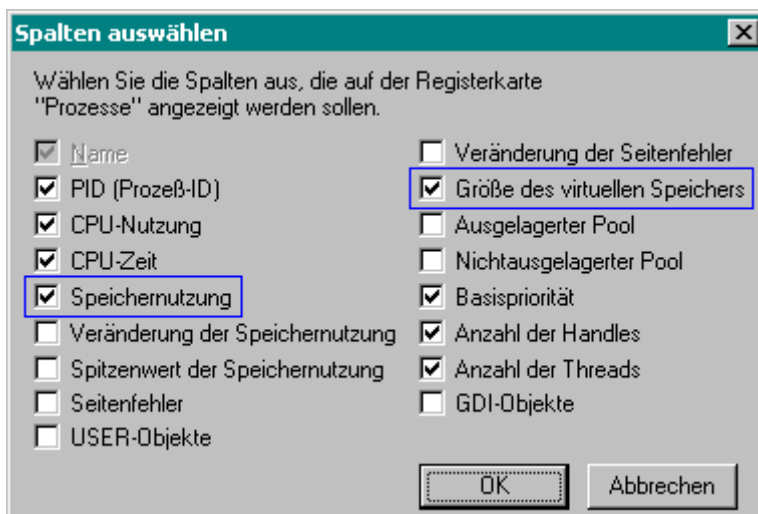


Abbildung: Dialog "Spalten auswählen" (Task-Manager)

Der in der Abbildung angezeigte Wert (11'464 KB) stellt den **gesamten** momentan von Microsoft Excel belegten physischen Speicher dar, d.h. derjenige Speicherplatz, der für

- das Programm 'Microsoft Excel',
- die Daten und

- die Verarbeitung/Bearbeitung der Daten belegt ist. Eine Beschreibung dieser Speicherbereiche finden Sie im Kapitel "[Die sechs Speicherbereiche von Excel](#)".

Die Anwendung Microsoft Excel als solche belegt zwischen 6 und 12 MB Speicher, je nach Excel-Version unterschiedlich (siehe Abschnitt "[Grösse der drei Speicherbereiche Program Memory, Workbook Data Memory und Workbook Memory](#)").

Beachten Sie bitte, dass der angezeigte Wert nicht nur den Speicher für das Programm Excel und die geöffneten Arbeitsmappen beinhaltet, sondern auch den von diesen Komponenten beanspruchten Speicher:

- [Symbolleisten-Informationen in der xlb-Datei](#)
- [VBA-Editor](#)
- [geladene Add-Ins](#)



Beanspruchter Speicher der xlb-Datei

Beim jedem Start von Microsoft Excel wird automatisch die xlb-Datei geladen. In dieser Datei sind sämtliche Informationen über benutzerdefinierte Symbolleisten und Steuerelemente gespeichert. Je grösser die xlb-Datei ist, desto mehr Speicher wird beansprucht.

Beachten Sie bitte, dass eine xlb-Datei fragmentiert sein kann. Das bedeutet, dass sich in der Datei fehlerhafte Daten befinden, d.h. Daten über Symbolleisten und Steuerelemente, die früher einmal existierten, später aber gelöscht wurden. In bestimmten Situationen kann es vorkommen, dass die Löschung nicht korrekt durchgeführt wird, sodass die Symbolleisten bzw. Steuerelemente zwar für den Benutzer nicht mehr vorhanden sind, die Informationen über diese nicht mehr existierenden Objekte jedoch weiterhin in der xlb-Datei gespeichert sind (die Datei ist somit 'fragmentiert'). Beim Start von Microsoft Excel werden diese 'Datenleichen' ebenfalls in den Speicher geladen.

Informationen über diese Problematik erhalten Sie hier:

	Weitere Informationen ■ The SOSCQ Page: Hintergrund-Informationen über die xlb-Datei
---	--

Es existiert ein Tool namens **XLB-Defragmentor**, mit dem man sämtliche fehlerhaften Daten aus der xlb-Datei entfernen kann. Detaillierte Produkt-Informationen zu diesem Tool und vieles mehr finden Sie hier:


	Informationen über den XLB-Defragmentor ■ Offizielle Produkt-Webseite des XLB-Defragmentors ■ Downloadseite: XLB-Defragmentor
---	--



Beanspruchter Speicher des VBA-Editors

Auch der VBA-Editor belegt eine bestimmte Menge Speicherplatz. Zu beachten ist, dass der VBA-Editor nicht nur in geöffnetem (sichtbarem) Zustand Speicher belegt. Es wird auch dann Speicher belegt, wenn er aktuell nicht geöffnet ist, aber im Laufe der Excel-Sitzung irgend einmal geöffnet und wieder geschlossen wurde.

Weitere interessante und hilfreiche Informationen dazu finden Sie hier:

	Weitere Informationen ■ Speichernutzung des VBA-Editors
---	---



Beanspruchter Speicher der Add-Ins

Jedes im Add-In-Manager von Microsoft Excel aktivierte Add-In wird beim Excelstart automatisch in den Arbeitsspeicher geladen. Das gilt auch für Add-Ins, die sich in einem Excel-Startordner befinden (XlStart oder zusätzlicher Startordner), oder die xla-Datei des Add-Ins auf eine andere Art und Weise geöffnet wurde (z.B. durch Doppelklick auf die Datei im Windows Explorer).

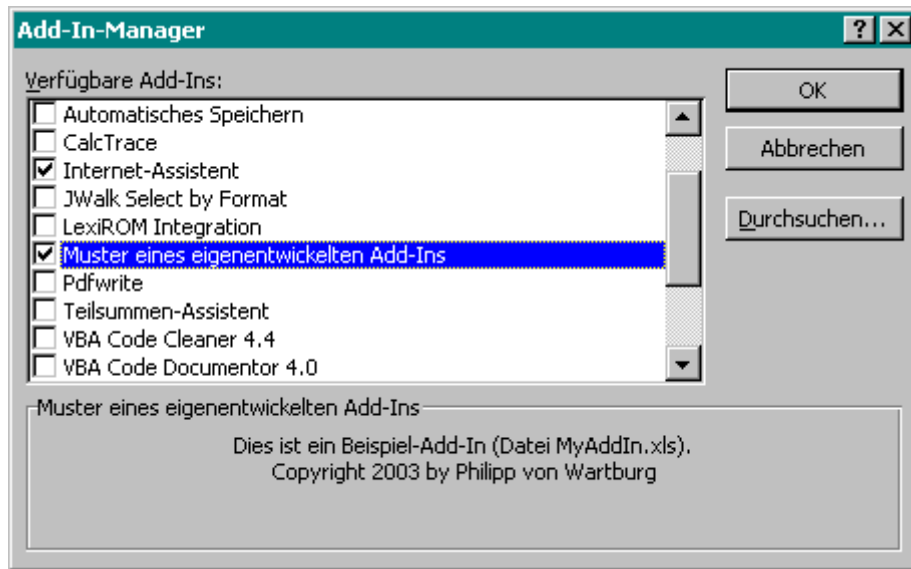



Abbildung: Add-In-Manager mit aktivierten Add-Ins

Durch das Laden eines Add-Ins belegt dieses somit Arbeitsspeicher. Vereinfacht kann man daher sagen: Je mehr Add-Ins geladen sind, desto mehr Arbeitsspeicher wird beansprucht beziehungsweise desto weniger Arbeitsspeicher steht für andere Dinge in Microsoft Excel und andere Prozesse wie beispielsweise ein anderes Windows-Programm (z.B. Microsoft Word oder Internet Explorer) zur Verfügung. Mit "andere Dinge" sind unter anderem Arbeitsmappen gemeint. Wenn zum Beispiel sehr viele grössere Add-Ins geladen sind, so kann es durchaus passieren, dass nur noch eine einzige oder sogar keine Arbeitsmappe nach dem Excelstart geöffnet werden kann.

Merken Sie sich daher folgendes:



Bitte beachten Sie, dass jedes in Microsoft Excel geladene Add-In einen bestimmten Anteil von Speicherplatz des Workbook Memory benutzt. Um möglichst viel verfügbaren Speicher zu erhalten, sollten daher nur diejenigen Add-Ins aktiviert werden, die tatsächlich zur Arbeit benötigt werden.

Verlauf der Speichernutzung während der Arbeit mit Excel

Die nachfolgend angegebenen Speichernutzungswerte wurden mit dem Task-Manager unter Windows NT 4 gemessen und stellen den gesamten von Microsoft Excel 97 benutzten Speicher dar.

Testfall 1

Aktion	Speichernutzung	Bemerkung
Excel im Safe Mode starten (Start-Parameter "/safe")	2'608 KB	Gemessen wurde unmittelbar nach vollständig aufgestartetem Programm. Eine neue leere Arbeitsmappe mit 3 Arbeitsblättern war sichtbar.
Warten (ca. 5 Minuten)	2'280 KB	-
Schliessen der leeren Arbeitsmappe	3'100 KB	-
Öffnen einer neuen leeren Arbeitsmappe	3'320 KB	-

Testfall 2

Aktion	Speichernutzung	Bemerkung
Excel starten	7'820 KB	Gemessen wurde unmittelbar nach vollständig aufgestartetem Programm. Eine neue leere Arbeitsmappe mit 1 Arbeitsblatt war sichtbar.
Excel zum Symbol verkleinern	688 KB	-
Excel zum Vollbild vergrößern	1'792 KB	-



Testfall 3

Aktion	Speichernutzung	Bemerkung
Excel starten	11'000 KB	Gemessen wurde unmittelbar nach vollständig aufgestartetem Programm. Mehrere Add-Ins wurden geladen. Eine neue leere Arbeitsmappe mit 1 Arbeitsblatt war sichtbar.
Zellbereich A1:J65536 (=10 Spalten) mit AutoFormat formatieren	23'000 KB	-
Spalten A:J löschen	29'000 KB	-
Zellbereich A1:J65536 (=10 Spalten) mit AutoFormat formatieren	35'000 KB	-



Speichernutzung der Daten in Excel

Das Speicher-Management von Excel

Microsoft Excel verwendet einen eigenen Speicherpool für die Arbeitsmappen-Daten sowie einen eigenen Speichermanager. Für den Excel Speicherpool wird gewöhnlich eine englische Bezeichnung benutzt, wobei es gleich deren drei gibt: *Excel Useable Memory*, *Workbook Memory* oder *Internal Heap Space*. Am häufigsten wird der Begriff *Workbook Memory* verwendet. In deutschen Microsoft-Dokumenten, die aus dem Englischen übersetzt wurden, wird oft der Ausdruck *Interner Heap-Speicher* verwendet (Anmerkung des Autors: Ich persönlich finde *Interner Heap-Speicher* viel zu technisch und bevorzuge daher den Begriff *Workbook Memory*).

Weitere Informationen über die Speicherbereiche von Excel finden Sie hier:

Weitere Informationen
■ [Die sechs Speicherbereiche von Excel](#)

Wie im Abschnitt "[Die sechs Speicherbereiche von Excel](#)" ausführlich beschrieben ist, kann der von Microsoft Excel benutzte Speicher in mehrere Bereiche aufgeteilt werden. Hier nochmals kurz eine schematische Darstellung dieser Bereiche:

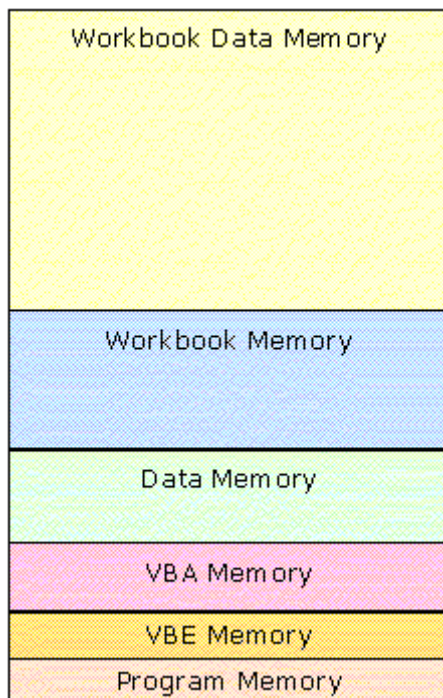


Abbildung: Die Bereiche des Excel-Speichers

In diesem Kapitel geht es um den von Microsoft Excel für **Daten** benutzbaren Speicher, also um die drei Speicherbereiche Workbook Data Memory, Workbook Memory und Data Memory. Die anderen Speicherbereiche, also VBA Memory, VBE Memory und Program Memory werden in anderen Kapiteln vorgestellt.



Das Workbook Data Memory von Excel

Alle in einer Excel-Arbeitsmappe eingetragenen Daten sind in einem Bereich des Arbeitsspeichers abgelegt, der bis zu 1 GB gross sein kann. Eine Arbeitsmappe kann somit theoretisch Daten enthalten, die bis zu 1 GB Platz im Speicher beanspruchen.

In diesem Speicherbereich sind Daten wie unter anderem Blattnamen, Zellwerte, Zelltexte, Namen von benannten Zellbereichen, Formeln und dergleichen abgelegt. Es sind also alles Daten, die einen gewissen statischen Charakter besitzen.

Man kann sich diesen Speicher - stark vereinfacht - als eine grosse Tabelle oder Liste vorstellen. In dieser steht unter anderem, dass beispielsweise in der Zelle A1 von Tabellenblatt Tabelle1 die Zeichenfolge "Kontoübersicht" eingetragen ist, dass in Zelle A3 das Wort "Nummer" steht und so weiter. Es ist quasi ein Abbild sämtlicher in den Tabellenblättern eingetragene Zellwerte. Beachten Sie, dass das Workbook Data Memory für die gesamte Excel-Instanz gilt. Es gibt somit nur ein einziges Workbook Data Memory für alle geöffneten Arbeitsmappen. Zu diesen gehören auch ausgeblendete Mappen wie die persönliche Makroarbeitsmappe Personl.xls und alle aktivierten Add-Ins (sofern diese Tabellenblätter mit Zellinhalten besitzen).

Weitere Informationen über die Speicherbereiche von Excel finden Sie hier:



Das Workbook Memory von Excel

In diesem Abschnitt geht es um den Sinn und Zweck des Workbook Memory. Wie bereits erwähnt verwendet Microsoft Excel einen eigenen Speicherpool, der von der Anwendung selbst verwaltet wird. Im Kapitel "[Grösse der drei Speicherbereiche](#)" wurden die Excel-Speicherbereiche und ihre jeweilige Grösse vorgestellt. Das Excel Workbook Memory ist gewöhnlich kleiner als der Arbeitsspeicher von Windows und gewöhnlich auch um einiges kleiner als der verfügbare physische Speicher (RAM) des Computers. Das Workbook Memory besitzt eine maximale Grösse, die je nach Excelversion verschieden ist (siehe Abschnitt "[Grösse des Workbook Memory](#)").

Weitere Informationen über die Speicherbereiche von Excel finden Sie hier:



Zweck und Aufgabe des Workbook Memory

Das Workbook Memory ist ein Bereich des von Excel zugänglichen Speichers, der vereinfacht ausgedrückt für die Verarbeitung der Arbeitsmappen-Daten benötigt wird. Es wird vorwiegend immer dann beansprucht, wenn bestimmte Operationen durchgeführt werden, wie unter anderem folgende (Aufzählung nicht abschliessend):

- Berechnen von geöffneten Arbeitsmappen
- Berechnen von Formeln und Funktionen
- Eingeben von Zellformeln
- Ausfüllen von Zellbereichen
- AutoAusfüllen von Zellbereichen
- Kopieren und Einfügen von Zellen

Das Workbook Memory wird somit insbesondere bei diesen Vorgängen verwendet:

- für das Verfolgen von Zellen und Formeln (engl. "Track cells and formulas")
- für das Bereitstellen der Kopier- und Einfügefunktion (engl. "Provide copy and paste functionality")
- für das Nachführen von Verknüpfungen zu Objekten (engl. "Track pointers to objects")

Im Workbook Memory sind keine statischen Daten wie beispielsweise Zellinhalte (Konstanten, d.h. Zahlen und Texte), eingefügte Grafiken, Kommentare und andere statische Daten abgelegt. Das Workbook Memory ist vielmehr eine Art "Arbeitsspeicher", d.h. Speicher, der zum Arbeiten benötigt wird.

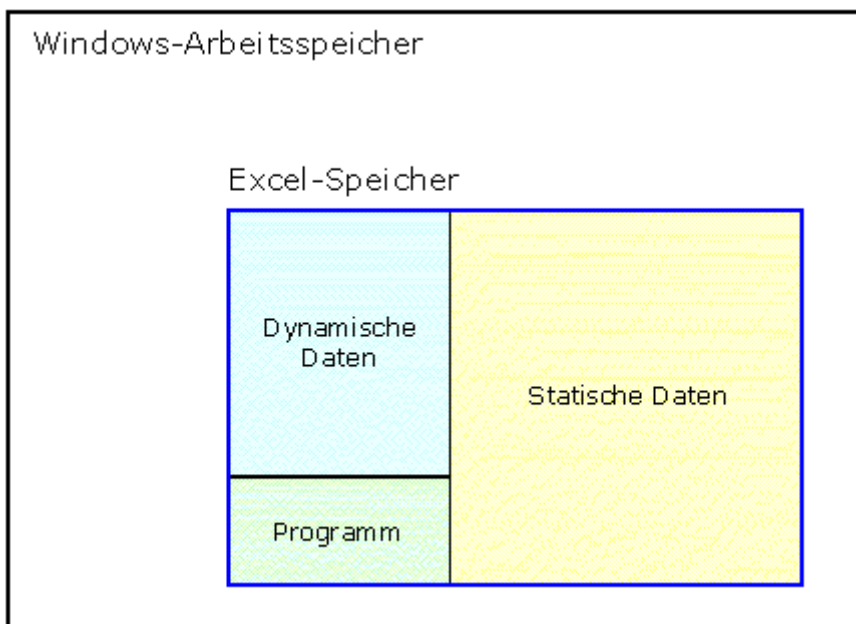


Abbildung: Die Speicherbereiche von Microsoft Excel

Das Workbook Memory wird von der gesamten Excel-Anwendung benutzt. Jede Excel-Instanz besitzt ein eigenes Workbook Memory. Alle in einer Instanz geöffneten Arbeitsmappen benutzen das Workbook Memory gemeinsam.

Aus Sicht des Workbook Memory ist ein aktiviertes Add-In eine Art Arbeitsmappe. Merken Sie sich daher folgendes:



Auch die in der Excel-Instanz geladenen Add-Ins belegen Speicherplatz des Workbook Memory. Wenn sich zum Beispiel eine Zellformel mit einer volatilen Funktion (z.B. ZUFALLSZAHL) auf einem Tabellenblatt eines Add-Ins befindet, so wird diese Formel bei jeder Änderung irgend einer Arbeitsmappe ebenfalls neu berechnet.

Grösse des Workbook Memory

Diese Tabelle zeigt die maximale Grösse des Workbook Memory der gängigen Excelversionen:

Excelversion	Workbook Memory
Excel 5.0	16 MB
Excel 95/7.0	64 MB (inoffiziell 80 MB)
Excel 97	64 MB (inoffiziell 80 MB)
Excel 2000	64 MB (inoffiziell 80 MB)
Excel 2002/XP	128 MB (inoffiziell 160 MB)
Excel 2003	1 GB
Excel V.12	Unbegrenzt

Das Workbook Memory von Excel 2002 ist also doppelt so gross wie das von Excel 97 und Excel 2000. Die Frage ist nun, was das in der Praxis bedeutet. Die Antwort ist relativ einfach: Da das Workbook Memory unter anderem zum Berechnen einer Arbeitsmappe benötigt wird, kann in Excel 2002 eine Arbeitsmappe grösser bzw. komplexer sein als in den Vorgängerversionen. Beispielsweise darf die Mappe mehr blattübergreifende Zellbezüge enthalten.

Neuer Sub-Titel (TODO)

Mappe mit 60 MB Workbook Memory öffnen

Kopie der gleichen Mappe zusätzlich öffnen -> nur 10 MB mehr Speicher belegt

Zweite Kopie öffnen -> Fehlermeldung "Ohne Rückgängig weiter?" erscheint beim Öffnen!

Verfügbarer, benutzter und gesamter Speicher abfragen

Wir wissen nun, dass es in Microsoft Excel das so genannte Workbook Memory gibt, welchen Zweck es besitzt und wie gross es maximal sein kann. Jetzt geht es darum, während der Arbeit mit Excel die Auslastung des Workbook Memory herauszufinden.

In Excel kann man anhand der Tabellenfunktion INFO drei Kennzahlen des Arbeitsspeichers abfragen. In einem VBA-Makro erhält man die gleichen Kennzahlen mit den Eigenschaften *MemoryFree*, *MemoryUsed* und *MemoryTotal* des *Application*-Objektes.

Tabellenfunktion INFO


Die Info-Typen der Tabellenfunktion INFO

Die Excel-Tabellenfunktion INFO nimmt als Argument folgende Info-Typen in Bezug auf Speicher entgegen:

Info-Typ	Info-Typ (engl.)	Beispiel	Ergebnis
BenutztSpeich	MemUsed	=INFO("BenutztSpeich")	Zeigt die Grösse des momentan für Daten benötigten Arbeitsspeichers in Bytes.
VerfSpeich	MemAvail	=INFO("VerfSpeich")	Zeigt den verfügbaren Arbeitsspeicher in Bytes.
GesamtSpeich	TotMem	=INFO("GesamtSpeich")	Zeigt den gesamten verfügbaren Arbeitsspeicher in Bytes einschliesslich der bereits belegten Teile des Arbeitsspeichers.

Es ist unerheblich, ob der Info-Typ in Gross- und/oder Kleinschrift geschrieben wird.

Tipp!

	Ich persönlich kann mir den Info-Typ "BenutztSpeich" irgendwie schlecht merken. Oft schreibe ich "BenutzerSpeicher", "BenutztSpeicher", "BenutzerSpeich" oder "BenSpeicher", was Excel natürlich nicht akzeptiert. Ich verwende daher statt "BenutztSpeich" die englische Bezeichnung "MemUsed", die ebenfalls den benutzten Speicher liefert (obwohl dies in der Excel-Dokumentation nicht erwähnt ist). Auch für "VerfSpeich" und "GesamtSpeich" kann man stattdessen die englischen Bezeichnungen verwenden. Diese heissen "MemAvail" (liefert den verfügbaren Speicher) und "TotMem" (liefert den gesamten Speicher).
---	---

Anzeige und Formatierung der Speicherwerte

Die wichtigste und zugleich einzige korrekt funktionierende Kennzahl ist die Grösse des benutzten Speichers, die man mit dem Info-Typ "BenutztSpeich" erhält. Die beiden anderen Werte für "Verfügbarer Speicher" und "Gesamter Speicher" können Sie vergessen, da sie schlicht und einfach unbrauchbar sind. Das Problem ist nämlich, dass die mit dem Info-Typ "VerfSpeicher" ermittelte Zahl immer 1'048'576 lautet. Wird diese Zahl durch 1024 geteilt, erhält man 1024, d.h. 1'024 KB. 1'024 KB sind genau 1 MB. Wieso Excel den verfügbaren Speicher generell mit 1 MB angibt, ist leider nicht bekannt.

Die nächste Abbildung zeigt einen Ausschnitt eines Excel-Tabellenblattes mit der INFO-Funktion in der Praxis. Die dritte Tabellenspalte zeigt die Formel, die zum Ermitteln der in der zweiten Spalte angegebenen Zahl verwendet wurde:

Verfügbarer Speicher:	1048576	=INFO("VerfSpeich")
Benutzer Speicher:	684244	=INFO("BenutztSpeich")
Gesamter Speicher:	1732820	=INFO("GesamtSpeich")

Abbildung: Die INFO-Tabellenfunktion in der Praxis

Da die Werte in der Masseinheit Bytes geliefert werden, empfehle ich zwecks besserer Lesbarkeit eine Umrechnung in Kilobytes (KB) durchzuführen. Dazu müssen Sie ein Formelergebnis lediglich durch die Zahl 1024 dividieren. Das Ergebnis kann ruhig auf eine Ganzzahl gerundet werden, weil allfällige Dezimalstellen 'nur' Bytes darstellen und kaum ins Gewicht fallen. Sie können bei Bedarf das Resultat nochmals durch 1024 dividieren, damit Sie Megabytes (MB) erhalten.

Hier ein paar Beispiele:

- Beispiel 1 (Berechnen in KB und MB):
 $684244 / 1024 = 668.20703125 = 668 \text{ KB}$
 $668 / 1024 = 0.65 \text{ MB}$
- Beispiel 2 (Berechnen in KB und MB):
 $11218884 / 1024 = 10955.94140625 = 10956 \text{ KB}$
 $10956 / 1024 = 10.69 \text{ MB}$
- Beispiel einer Zellformel, die den benutzten Speicher in KB liefert:
 $=\text{INFO}(\text{"BenutztSpeich"})/1024$
- Beispiele von Zellformeln, die den benutzten Speicher in MB liefern:
 $=\text{INFO}(\text{"BenutztSpeich"})/1024/1024$
 $=\text{INFO}(\text{"BenutztSpeich"})/(1024*1024)$
 $=\text{INFO}(\text{"BenutztSpeich"})/(1024^2)$
 $=\text{INFO}(\text{"BenutztSpeich"})/1048576$


Spezialitäten der INFO-Tabellenfunktion

1) Neben den drei oben vorgestellten Info-Typen "BenutztSpeich", "VerfSpeich" und "GesamtSpeich" kennt die INFO-Tabellenfunktion noch 7 weitere Typen:

Info-Typ	Info-Typ (engl.)	Ergebnis
Dateizahl	numfile	Zeigt die Anzahl Tabellen- und Dialogblätter aller geöffneten Arbeitsmappen.
Rechenmodus	recalc	Zeigt den aktuellen Berechnungsmodus "Automatisch" oder "Manuell".
System	system	Zeigt den Namen des aktuellen Betriebssystems.
Sysversion	osversion	Zeigt die Version des aktuellen Betriebssystems als Text.
Ursprung	origin	Zeigt den Zellbezug der sichtbaren, obersten linken Zelle im aktuellen Fensterbereich zurück.
Version	version	Zeigt die Version von Microsoft Excel als Text.
Verzeichnis	directory	Zeigt den Pfad des aktuellen Verzeichnisses.

In der deutschsprachigen Ausgabe von Microsoft Excel können sowohl die deutschen als auch die englischen Info-Typen verwendet werden. Wenn die Arbeitsmappe in verschiedensprachigen Ausgaben von Excel benutzt werden soll, sollten Sie möglichst nur die englischen Info-Typen einsetzen, da diese in jeder Ausgabe funktionieren. Anderenfalls könnten Probleme auftreten, weil die deutschen Info-Typen ausschliesslich in der deutschen Excel-Ausgabe funktionieren. Es ist möglich, dass beispielsweise die Zellformel "=INFO("BenutztSpeich")" in der französischen Excel-Ausgabe nicht berechnet werden kann, d.h. in der Zelle der Fehler "#WERT!" angezeigt wird.

2) Wenn die INFO-Tabellenfunktion mit dem Info-Typ "BenutztSpeich", "VerfSpeich" oder "GesamtSpeich" benutzt wird, so benötigt sie zum Berechnen viel Prozessorleistung. Beachten Sie daher folgendes:

	<p>Die INFO-Tabellenfunktion benötigt zum Ermitteln eines Speicherwertes sehr viel Rechenpower. Die CPU wird während der Berechnung erheblich belastet. Verwenden Sie die Funktion mit einem bestimmten Info-Typ wenn möglich nur einmal in einer Arbeitsmappe. Wird das Formelergebnis noch in weiteren Zellen benötigt, sollte anstelle der Funktion ein Zellbezug auf die bereits vorhandene Zellformel verwendet werden. Wenn beispielsweise der Wert "Benutzter Speicher" sowohl in Zelle A1 als auch in Zelle B2 benötigt wird, so ist in Zelle A1 die Formel "=INFO("BenutztSpeich")" und in Zelle B2 der Bezug "=A1" einzutragen. Dadurch wird die INFO-Funktion nur einmal berechnet.</p>
---	--

3) Das Berechnen der INFO-Tabellenfunktion mit einem der drei Info-Typen für Speicher-Kennzahlen kann im Vergleich zu anderen Tabellenfunktionen bzw. Info-Typen sehr viel länger dauern. Der Grund liegt am aufwändigen Vorgehen. Die INFO-Tabellenfunktion wird im Prinzip wie jede andere volatile Tabellenfunktionen berechnet. Volatile Funktionen werden immer neu berechnet, wenn in einer beliebigen Zelle des Tabellenblattes eine Änderung bzw. Berechnung durchgeführt wird. Nicht flüchtige Funktionen werden dagegen nur dann neu berechnet, wenn sich die Eingabevariablen ändern. Da es irrelevant ist, wann die Funktion innerhalb des Berechnenvorganges neu berechnet wird (als volatile Funktion wird sie sowieso neu berechnet), kann die Funktion zu einem beliebigen Zeitpunkt verarbeitet werden:

- Wenn Sie beispielsweise die Tabellenfunktion ZUFALLSZAHL in drei Zellen verwenden, so werden diese drei Funktionen nacheinander und völlig unabhängig voneinander berechnet und als Folge drei unterschiedliche Formelergebnisse ausgegeben. Die Reihenfolge der Berechnung spielt keine Rolle, da die Funktionen nichts weiter als eine zufällige Zahl liefern müssen.

- Wenn die Tabellenfunktion JETZT in sehr vielen Zellen verwendet wird, so kann es ohne weiteres vorkommen, dass nicht alle Formelzellen die gleiche Zeitangabe anzeigen. Dies ist dann der Fall, wenn die Neuberechnung aller JETZT-Tabellenfunktionen länger als eine Sekunde dauert. Die letzte berechnete JETZT-Funktion zeigt dann eine Zeit, die eine Sekunde oder mehr von der Zeit der zuerst berechneten JETZT-Funktion abweicht. Das ist nicht weiter schlimm, **XXXXXXXXXX**

- Wenn man die Tabellenfunktion INFO zum Ermitteln von Speicher-Kennzahlen einsetzt und die Funktion in sehr vielen Zellen verwendet, so müssen die Zellen mehrmals neu berechnet werden. **XXXXXXXXXX** -> Weil alle Zellformeln den gleichen Wert anzeigen müssen -> Der Speicherwert kann im Laufe des Berechnenvorganges ändern -> Dann müssen die bereits berechneten Formeln nochmal neu berechnet werden

4) Volatile INFO-Tabellenfunktion zusätzlich volatil machen:

$$=(\text{INFO}(\text{"memused"})/1024)+(\text{JETZT}()*0)$$



Application-Eigenschaften MemoryFree, MemoryUsed und MemoryTotal

Die drei Speicherwerte können auch ohne die Excel-Tabellenfunktion ermittelt werden. Und zwar indem Sie mit einem VBA-Makro die drei Eigenschaften "MemoryFree", "MemoryUsed" und "MemoryTotal" des *Application*-Objektes abfragen.

Eigenschaft	Ergebnis
Application.MemoryFree	Zeigt den verfügbaren Arbeitsspeicher in Bytes.
Application.MemoryUsed	Zeigt die Grösse des momentan für Daten benötigten Arbeitsspeichers.
Application.MemoryTotal	Zeigt den gesamten verfügbaren Arbeitsspeicher in Bytes einschliesslich der bereits belegten Teile des Arbeitsspeichers.



Was die Speicherwerte bedeuten

Die mit der Tabellenfunktion INFO bzw. den oben vorgestellten *Application*-Eigenschaften ermittelten Werte sind leider nur bedingt nützlich.

Beachten Sie bitte, dass der Wert für "Benutzer Speicher" nicht nur den von der aktiven Arbeitsmappe für Daten benötigten Speicher zeigt, sondern die Speichernutzung der gesamten Excel-Instanz. In diesem Wert sind sämtliche

- geöffneten Arbeitsmappen (auch ausgeblendete),
 - aktivierte bzw. geladene Add-Ins und
 - Symbolleisten
- enthalten.



Speicherbedarf einer Arbeitsmappe ermitteln

Insbesondere bei einer grösseren Arbeitsmappe ist es von Vorteil, wenn man ihren Speicherbedarf kennt.

Es existieren mehrere Möglichkeiten, wie man die Grösse des von einer bestimmten Arbeitsmappe belegten Speichers herausfinden kann. Mit 'Bordmitteln', also mit den in Windows und Excel standardmässig vorhandenen Möglichkeiten, ist dies jedoch alles andere als einfach. Grundsätzlich aber ist es machbar. Mit einem Spezial-Tool könnten Sie allerdings detaillierte Analysen durchführen, und erhalten die Speicherwerte mehrheitlich auf Knopfdruck. Daher möchte ich gleich vorneweg darauf hinweisen, dass die Anschaffung eines derartigen Tools auf jeden Fall sinnvoll ist, insbesondere wenn Sie häufig grössere Arbeitsmappen erstellen oder ganze Anwendungen mit Excel/VBA realisieren.

Bei Microsoft Excel 97 belegt eine neue, leere Mappe mit einem einzelnen Arbeitsblatt ca. 4 KB Speicher.

Speicherbedarf von Arbeitsmappen reduzieren



Möglichkeit 1: Task-Manager

Man kann mit Hilfe des Windows Task-Managers herausfinden, wieviel Speicher eine bestimmte Arbeitsmappe ungefähr beansprucht. Dazu müssen Sie die Arbeitsmappe mehrmals hintereinander öffnen und schliessen, und dabei ständig die im Task-Manager angezeigte Speichernutzung des Prozesses EXCEL.EXE überwachen.

Gehen Sie wie folgt vor:

- Starten Sie Microsoft Excel.
- Öffnen Sie den Task-Manager und wechseln zur Registerseite "Prozesse". Suchen Sie in der Prozessliste den Eintrag EXCEL.EXE.



Möglichkeit 2: INFO-Tabellenfunktion

INFO-Tabellenfunktion



Möglichkeit 3: Spezialtools

Mit geeigneten Spezialtools lässt sich die Speichernutzung am besten feststellen.

[FastExcel](#)

Decision Models: FastExcel

www.decisionmodels.com/fastexcel.htm



Der Undo-Speicher von Excel

Was ist der Undo-Speicher?

In Microsoft Excel können standardmässig die letzten 16 Aktionen rückgängig gemacht werden. Damit dies ermöglicht wird, werden die Aktionen im Undo-Speicher abgelegt. Für jede Aktion muss festgehalten werden:

- Art der Aktion (z.B. "Eingabe")
- Betroffenes Objekt (z.B. "Zelle B5")
- Wert (z.B. "=SUMME(B1:B4)")

Je nach Art der Aktion sind ganz unterschiedliche Informationen enthalten. Beim Verschieben einer Grafik beispielsweise wird folgendes registriert:

- Art der Aktion: Objekt verschieben
- Betroffenes Objekt: Bild 2
- Wert: (Neue Bild-Position)

Berücksichtigen Sie, dass nicht nur Tätigkeiten wie das Eingeben, Bearbeiten und Löschen von Zell-Werten, -Texten und -Formeln als Aktionen im Undo-Speicher eingetragen werden sondern auch Aktionen wie das Formatieren von Zellen und die Benutzung von AutoFormat. Dies ist wichtig zu wissen, da es nämlich ein kleines Problem bei der AutoFormat-Funktion gibt, wenn der zu formatierende Zellbereich sehr gross ist.

Der Undo-Speicher befindet sich im Speicherbereich des [Workbook Memory](#). Für jeden Undo-Schritt wird bereits beim Excelstart eine bestimmte Menge Speicherplatz reserviert, damit sichergestellt werden kann, dass auch bei knappem Speicher sämtliche zuletzt ausgeführten Aktionen rückgängig gemacht werden können. Der Undo-Speicher bzw. die Undo-Aktionen beziehen sich auf die gesamte Excel-Instanz und werden somit von allen geöffneten Arbeitsmappen gemeinsam benutzt. Das bedeutet, dass sich die Arbeitsmappen die verfügbaren Rückgängig-Schritte teilen. Es stehen nicht für jede geöffnete Mappe 16 Rückgängig-Schritte zur Verfügung.

Tipp!



Wenn Sie eine Aktion ausgeführt haben, diese jedoch nicht das gewünschte Ergebnis lieferte, sollten Sie die Aktion rückgängig machen und nicht anhand einer anderen Aktion den Ursprungszustand wiederherstellen.

Beispiel:

Sie markieren die Spalten A:J und formatieren sie mit dem AutoFormat-Format "Standard 1". Das Resultat gefällt Ihnen aber nicht. Sie möchten lieber das Format "Standard 2" verwenden, und müssen daher zuerst

die Formatierung entfernen. Führen Sie dazu unbedingt den Befehl "Rückgängig: AutoFormat" aus (oder Strg+Z). Sie sollten nicht die Spalten A:J selektieren und den Menübefehl "Bearbeiten/Löschen/Formate" ausführen.



Undo-Speicher optimal verwenden

Wie weiter oben erwähnt wurde, ist der Undo-Speicher ein Teil des [Workbook Memory](#). Im Workbook Memory werden unter anderem die Informationen über die Aktionen abgelegt, die rückgängig gemacht werden können. Das bedeutet folglich, dass bei vielen und speicherintensiven Undo-Aktionen für andere Daten weniger Platz im Workbook Memory zur Verfügung steht, beispielsweise zum Berechnen von Formeln. Sie sollten daher darauf achten, dass aufwändige Aktionen möglichst bald aus dem Undo-Speicher entfernt werden, was man durch Speichern der Arbeitsmappe erledigen kann. Eine aufwändige Aktion ist unter anderem das Formatieren eines grösseren Zellbereiches anhand der AutoFormat-Funktion. Beispielsweise werden beim Auto-Formatieren von (nur) 10 Spalten ungefähr 5 MB Daten im Undo-Speicher abgelegt.

Wenn eine Aktion nicht zum erhofften Resultat geführt hat, sollten Sie die Aktion tatsächlich auch anhand der Undo-Funktion rückgängig machen. Damit verwenden Sie nicht nur ein nützliches Excel-Feature, sondern sparen je nach Situation erheblich Arbeitsspeicher.

Beispiel 1

Aktion	Speichernutzung
Sie starten Microsoft Excel mit einer neuen Arbeitsmappe	11 MB
Sie markieren die Spalten A:J und formatieren sie mit der AutoFormat-Funktion	23 MB
Sie markieren die Spalten A:J und führen den Menübefehl <i>Bearbeiten/Löschen/Formate</i> aus	27 MB

Beispiel 2

Aktion	Speichernutzung
Sie starten Microsoft Excel mit einer neuen Arbeitsmappe	11 MB
Sie markieren die Spalten A:J und formatieren sie mit der AutoFormat-Funktion	23 MB
Sie führen den Menübefehl <i>Bearbeiten/Rückgängig: AutoFormat</i> aus	17 MB

Weitere Informationen zum Thema "Verwenden der Rückgängig-Funktion":

OFFICE97: Verwendung der Funktion RÜCKGÄNGIG bei eingefügten Objekten

<http://support.microsoft.com/?scid=kb;de;172542>

OFFICE97: Using the Undo Feature with Inserted Objects

<http://support.microsoft.com/?scid=kb;EN-US;172542>

OFFICE98: Verwendung der Funktion "Rückgängig" bei eingefügten Objekten

<http://support.microsoft.com/?scid=kb;de;183279>

OFFICE98: Using the Undo Feature with Inserted Objects

<http://support.microsoft.com/?scid=kb;EN-US;183279>

OFFICEXP: HOW TO: Use the Undo Feature with Inserted Objects in Office XP Programs

<http://support.microsoft.com/?scid=kb;en-us;290148>



Anzahl Undo-Schritte reduzieren

Wenn Sie mit einer sehr grossen Arbeitsmappe arbeiten und praktisch auf jedes Byte Speicher angewiesen sind, können Sie mit einem Trick der Arbeitsmappe mehr Speicher zur Verfügung stellen. Dies lässt sich bewerkstelligen, indem Sie die Anzahl Rückgängig-Aktionen verkleinern. Standardmässig besitzt Microsoft Excel

16 Rückgängig-Ebenen (Undo Levels). Diese Anzahl kann anhand eines Eintrages in der Windows Registry geändert werden.

Vorgehen

So reduzieren Sie die Anzahl Rückgängig-Schritte:

1. Führen Sie das Windows-Utility 'RegEdit' aus (regedit.exe)
2. Selektieren Sie den Zweig "HKEY_CURRENT_USER\Software\Microsoft\Office\8.0\Excel\Microsoft Excel"
3. Legen Sie einen neuen 'DWORD'-Wert an (Menübefehl "Bearbeiten/Neu/DWORD-Wert")
4. Geben Sie dem neuen Schlüssel den Namen "UndoHistory"
5. Tragen Sie als Wert eine Zahl kleiner als 16 ein (Menübefehl "Bearbeiten/Ändern", Option "Dezimal")



Anzahl Undo-Schritte erhöhen

Der Undo-Speicher kann natürlich auch vergrößert werden. Wie erwähnt besitzt Microsoft Excel standardmässig 16 Rückgängig-Schritte (Undo Levels). Diese Anzahl kann anhand eines Eintrages in der Windows Registry geändert werden.

Vorgehen

So erhöhen Sie die Anzahl Rückgängig-Schritte:

1. Führen Sie das Windows-Utility 'RegEdit' aus (regedit.exe)
2. Selektieren Sie den Zweig "HKEY_CURRENT_USER\Software\Microsoft\Office\8.0\Excel\Microsoft Excel"
3. Legen Sie einen neuen 'DWORD'-Wert an (Menübefehl "Bearbeiten/Neu/DWORD-Wert")
4. Geben Sie dem neuen Schlüssel den Namen "UndoHistory"
5. Tragen Sie als Wert eine Zahl grösser als 16 ein (Menübefehl "Bearbeiten/Ändern", Option "Dezimal")

Die Abbildung zeigt eine Rückgängig-Liste, welche auf 300 Einträge erweitert wurde:

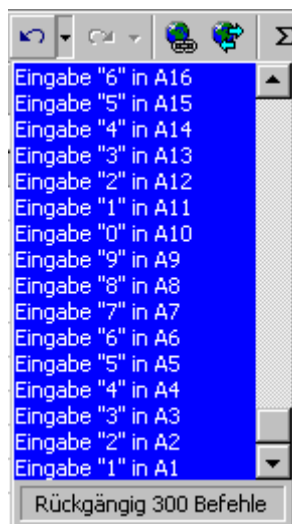


Abbildung: Die Rückgängig-Liste

Anmerkung

Microsoft empfiehlt, den Wert nicht grösser als 100 zu wählen. Laut einem Knowledge Base-Artikel wird für jede Undo-Aktion bereits beim Excelstart Arbeitsspeicher reserviert, damit sichergestellt werden kann, dass auch bei knappem Speicher sämtliche zuletzt durchgeführten Aktionen rückgängig gemacht werden können. Dieses Verhalten konnte ich allerdings nicht nachvollziehen. Ich führte mehrere Tests durch und stellte die Anzahl Rückgängig-Schritte unter anderem auf 5'000'000 (5 Million!). Beim nächsten Excelstart war die Speichernutzung von Excel jedoch nicht erkennbar höher als z.B. bei nur 30 Rückgängig-Schritten. Von einer Speicherplatz-Reservierung kann daher meiner Meinung nach nicht die Rede sein.

Übrigens: Der grösste erlaubte Wert liegt bei ungefähr 2 Milliarden.



Undo-Speicher leeren

Undo-Speicher in Excel leeren

Durch das Speichern einer Arbeitsmappe wird der von dieser Mappe belegte Undo-Speicher geleert. Dadurch werden die Informationen über die verfügbaren Rückgängig-Befehle für diese Mappe gelöscht. Es werden keinerlei Informationen in die Arbeitsmappendatei geschrieben. Diejenigen Aktionen, die sich auf andere geöffnete Arbeitsmappen beziehen, verbleiben im Undo-Speicher und können somit weiterhin rückgängig gemacht werden. Der beim Speichern verwendete Dateityp besitzt keinen Einfluss. Auch wenn Sie die Mappe beispielsweise als txt- oder csv-Datei speichern, wird der Undo-Speicher geleert. Auch das Speichern über den Menübefehl "Speicher unter" verhindert das Leeren des Undo-Speichers nicht.

Der Undo-Speicher wird bei diesen Aktionen geleert:

- Speichern der Arbeitsmappe
- Schliessen der Arbeitsmappe
- Ausführen eines Makros
- Aktualisieren der Daten einer Pivot-Tabelle



Undo-Speicher mit VBA leeren

Sobald ein VBA-Makro eine Anweisung enthält, die eine Aktion ausführt, welche gewöhnlich im Undo-Speicher festgehalten wird, so wird der Undo-Speicher geleert.

```
ActiveSheet.UsedRange
```

Undo-Methode zweimal aufrufen:

```
Application.Undo  
Application.Undo
```

```
ActiveWorkbook.SaveCopyAs "C:\Daten\Mappe1-Kopie.xls"
```

```
ActiveSheet.Range("Z1").Value = "" -> leert Undo-Speicher nicht  
ActiveSheet.Range("Z1").Value = 0 -> leert Undo-Speicher
```

```
ActiveSheet.Range("A99").EntireRow.Hidden = True -> leert Speicher
```



Fehlermeldungen im Zusammenhang mit dem Undo-Speicher

Beim Ausführen einer Aktionen, die im Undo-Speicher festgehalten und somit rückgängig gemacht werden kann, kann in bestimmten Situationen die Fehlermeldung "Nicht genügend Arbeitsspeicher. Ohne Rückgängig weiter?" (engl. "Not enough memory. Continue without Undo?") auftreten.

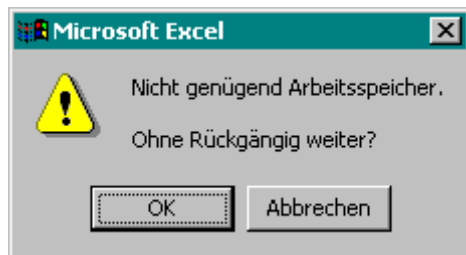


Abbildung: Fehlermeldung "Nicht genügend Arbeitsspeicher. Ohne Rückgängig weiter?"

Beachten Sie bitte, dass die obige Fehlermeldung nicht ausschliesslich bedeutet, dass der Undo-Speicher voll ist. Der Meldungstext lautet "Nicht genügend Arbeitsspeicher"; es wird also mit keinem Wort der Undo-Speicher erwähnt. Es ist nämlich durchaus möglich - wenn nicht sogar der Normalfall - dass die auszuführende Aktion nicht vollständig und korrekt abgeschlossen werden kann.

Die oben abgebildete Fehlermeldung kann sehr einfach erzeugt werden, wie die beiden nachfolgenden Beispiele zeigen.

Beispiel 1

1. Öffnen Sie eine neue Arbeitsmappe
2. Drücken Sie die Tastenkombination *Strg+A* um alle Zellen des aktiven Tabellenblattes zu markieren
3. Führen Sie den Menübefehl *Format/AutoFormat*
4. Klicken Sie im Dialogfenster "AutoFormat" auf OK

Beispiel 2

1. Öffnen Sie eine neue Arbeitsmappe
2. Geben Sie die Matrixformel " $=\text{SUMME}(\text{WENN}(\text{B1:DZ65535}=0;1;0))$ " in die Zelle A1 des aktiven Tabellenblattes ein (Formeleingabe mit *Strg+Umschalt+Eingabe* abschliessen)

Excel versucht die Matrixformel zu berechnen, was enorm viel Speicher benötigt. Nach ein bis zwei Minuten (die Dauer ist unter anderen abhängig von der Geschwindigkeit des Computers) erscheint die Fehlermeldung "Nicht genügend Arbeitsspeicher. Ohne Rückgängig weiter?". Wenn Sie die Meldung mit OK bestätigen, tritt der Fehler "Nicht genügend Arbeitsspeicher" auf. Nach Schliessen dieser Meldung wird der belegte Speicherbereich geleert und freigegeben, wodurch die Excel-Speichernutzung wieder auf ein normales Niveau zurückgeht. In der Zelle mit der Matrixformel wird die Zahl 0 angezeigt und in der Statusleiste ist der Hinweis "Berechnen" eingeblendet.

Achtung!

Der Hinweis "Berechnen" bedeutet, dass sich Excel in einem speziellen Zustand befindet, bei dem **sämtliche** Zellen neu berechnet werden, sobald man eine Zelle oder eine Spaltenbreite ändert. Das bedeutet, dass die obige Matrixformel erneut berechnet wird, was verständlicherweise nochmal zur Fehlermeldung "Nicht genügend Arbeitsspeicher. Ohne Rückgängig weiter?" führt. Sie sollten daher die Matrixformel löschen oder dann den in der WENN-Funktion angegebenen Zellbereich so stark verkleinern, dass die Formel korrekt berechnet werden kann. Sie können auch den Berechnen-Modus (Optionen-Dialog, Registerseite "Berechnen") auf "Manuell" stellen, damit generell keine Neuberechnung mehr stattfindet. Sobald Sie jedoch den Modus auf "Automatisch" wechseln, wird die Formel neu berechnet, und die bekannte Fehlermeldung erscheint.

Maximale Arraygrösse wird überschritten



Anwendungsfehler wegen Speicher-Bugs

In seltenen Fällen stürzt Excel mit einem Anwendungsfehler komplett ab, weil eine Excel-interne Programmroutine einen ungültigen Speicherzugriff ausführt oder einen Speicherplatz irrtümlich nicht freigibt.

Derartige "schwerwiegende" Programmfehler sind allerdings wirklich selten, und werden gewöhnlich automatisch im nächsten Service Pack behoben.

In der Microsoft Knowledge Base existiert nur ein einziger Artikel über einen solchen Fehler:

XL2002: Anwendungsfehler in Modul "Excel.exe" bei Offset 0001d05f

<http://support.microsoft.com/default.aspx?scid=kb;de;321312>

XL2002: Application Error in Module Excel.exe at Offset 0001d05f

<http://support.microsoft.com/default.aspx?scid=kb;en-us;321312>

» Behoben im Microsoft Office XP Service Pack 2

▲
TOP

VBA-Laufzeitfehler infolge ungenügendem Speicher

Allgemeines über VBA-Fehlermeldungen

In VBA-Programmen können vor allem diese drei Laufzeitfehler wegen ungenügendem Speicher auftreten:

- » [Laufzeitfehler Nr. 7 "Nicht genügend Speicher"](#)
- » [Laufzeitfehler Nr. 14 "Nicht genügend Zeichenfolgespeicher"](#)
- » [Laufzeitfehler Nr. 28 "Nicht genügend Stapelspeicher"](#)

Nachfolgend werden diese drei Fehlermeldungen im Detail vorgestellt, wobei sowohl der deutsche als auch der englische Text aus der Visual Basic-Referenz angegeben ist, weil die Fehlerbeschreibungen nicht immer identisch sind.

▲

Laufzeitfehler Nr. 7 "Nicht genügend Speicher"



Abbildung: Laufzeitfehler 7 "Nicht genügend Speicher"

"Nicht genügend Speicher (Fehler 7)" (Auszug aus der Visual Basic-Referenz)

Es ist mehr Speicher erforderlich, als verfügbar ist, oder eine 64 KB-Segmentgrenze wurde erreicht. Dieser Fehler hat die folgenden Ursachen und Lösungen:

Es sind zu viele Anwendungen, Dokumente oder Quelldateien geöffnet.

» Schliessen Sie alle geöffneten Anwendungen, Dokumente oder Quelldateien, die Sie nicht benötigen.

Ein Modul oder eine Prozedur ist zu gross.

» Teilen Sie sehr grosse Module oder Prozeduren in kleinere Komponenten auf. Dadurch wird zwar kein Speicher eingespart, aber das Überschreiten von 64 KB-Segmentgrenzen kann vermieden werden.

Sie haben Microsoft Windows im Standard-Modus gestartet.

» Starten Sie Microsoft Windows erneut im erweiterten Modus.

Sie haben Microsoft Windows im erweiterten Modus gestartet, aber es ist kein virtueller Speicher mehr verfügbar.

» Vergrössern Sie den virtuellen Speicher, indem Sie zusätzlichen Festplattenspeicher bereitstellen, oder stellen Sie sicher, dass freier Speicherplatz vorhanden ist.

Es sind TSR-Programme (TSR=terminate-and-stay-resident) aktiv.

» Entfernen Sie die TSR-Programme.

Es wurden zu viele Gerätetreiber geladen.

» Entfernen Sie alle nicht benötigten Gerätetreiber.

Es ist kein Speicherplatz mehr für Public-Variablen verfügbar.

» Verringern Sie die Anzahl von Public-Variablen.

Man kann den Laufzeitfehler 7 ganz einfach auch absichtlich erzeugen, indem man im Direktfenster des VBA-Editors die Anweisung

?Cells

eingibt und die *Eingabe*-Taste drückt. Die Fehlermeldung "Nicht genügend Speicher" wird sofort erscheinen. Das funktioniert jedoch nur bei Microsoft Excel 97.

Achtung: In Microsoft Excel 2000 oder neuer sollten Sie diese Anweisung nicht ausprobieren, da sich bei diesen Excelversionen Ihr Rechner "zu Tode quälen" wird. Genauer gesagt wird Excel so lange den Arbeitsspeicher füllen und nicht benötigte Speichersegmente in die Auslagerungsdatei schreiben, bis der gesamte Speicher voll ist. Das kann mehrere Minuten dauern, wobei das System während dieser Zeit voll ausgelastet ist, nicht mehr reagiert und instabil wird, weil für Systemprozesse nicht genügend Speicher zur Verfügung steht. Oft stürzt Excel sogar komplett ab.

Weitere Informationen dazu finden Sie in Kapitel "[Excel durch Belegen des gesamten Speichers zum Absturz bringen](#)".

Hinweis

Beachten Sie bitte, dass es eine fast identische Fehlermeldung mit dem gleichen Meldungstext gibt, die jedoch kein Laufzeitfehler ist. Der Fehlermeldungsdialog sieht so aus:



Abbildung: Fehler "Nicht genügend Speicher"

Dieser Fehler "Nicht genügend Speicher" kann unter anderem auftreten, wenn im VBA-Editor mehrere Benutzerformular-Fenster eines VBA-Projektes geöffnet sind und dann die Arbeitsmappe geschlossen wird (unabhängig davon, ob der VBA-Editor vorher geschlossen wird oder nicht). Anmerkung des Autors: Die genaue Ursache für das Erscheinen der Fehlermeldung konnte ich noch nicht herausfinden.



Runtime Error No. 7 "Out of memory"

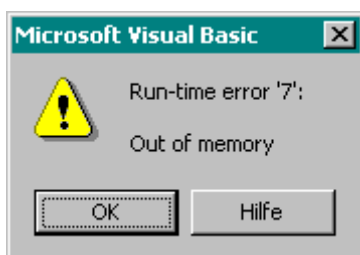


Abbildung: Laufzeitfehler 7 "Out of memory"

"Out of memory (error 7)" (Auszug aus der Visual Basic-Referenz)

More memory was required than is available or a 64K segment boundary was encountered. To prevent this error, try the

following:

- » Close any unnecessary applications, documents, or source files that are in memory.
- » If you have extremely large modules or procedures, consider breaking them into smaller ones. This procedure doesn't save memory, but it can prevent hitting 64K segment boundaries.
- » If you are running Microsoft Windows in standard mode on an 80386 or 80486 computer, try running it in enhanced mode.
- » If you are running Microsoft Windows in enhanced mode, free up some disk space, or at least ensure that some space is available.
- » Eliminate terminate-and-stay-resident (tsr) programs.
- » Eliminate unnecessary device drivers.
- » Reduce the number of global variables.



Laufzeitfehler Nr. 14 "Nicht genügend Zeichenfolgenspeicher"

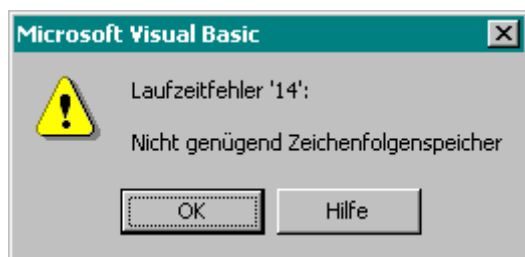


Abbildung: Laufzeitfehler 14 "Nicht genügend Zeichenfolgenspeicher"

"Nicht genügend Zeichenfolgenspeicher (Fehler 14)" (Auszug aus der Visual Basic-Referenz)

In Visual Basic können Sie sehr lange Zeichenfolgen verwenden. Dieser Fehler kann jedoch auftreten, wenn andere Programme Zeichenfolgenspeicher in einer bestimmten Weise benötigen oder wenn Sie Zeichenfolgen in Ihrem Programm in bestimmter Weise verändern. Dieser Fehler hat die folgenden Ursachen und Lösungen:

Ausdrücke, die das Erzeugen temporärer Zeichenfolgen erfordern, können diesen Fehler auslösen. Der folgende Code erzeugt zum Beispiel bei manchen Betriebssystemen die Fehlermeldung "Nicht genügend Zeichenfolgenspeicher":

```
TestZF = "Hallo"  
For Anzahl = 1 To 100  
    TestZF = TestZF & TestZF  
Next Anzahl
```

- » Weisen Sie die Zeichenfolge einer Variablen mit einem anderen Namen zu.

Für Ihr System steht kein Speicher mehr zur Verfügung, wodurch kein Speicherplatz mehr für eine Zeichenfolge reserviert werden konnte.

- » Stellen Sie Speicherplatz zur Verfügung, indem Sie nicht länger benötigte Anwendungen aus dem Arbeitsspeicher entfernen.

Hinweis

Sowohl VBA-Editor als auch VBA-Laufzeitumgebung laufen quasi als Sub-Prozess innerhalb des Office-Programmes (Excel, Word, PowerPoint etc.) und müssen sich daher den Arbeitsspeicher mit dem Office-Programm teilen. Wenn ein VBA-Programm exzessiv Gebrauch vom Speicher macht, wird somit automatisch auch der Speicher des Office-Programmes beansprucht.

Anmerkung des Autors

Das obige, aus der Visual Basic-Referenz stammende Codebeispiel erzeugt bei meinem Excel 97 keinen Laufzeitfehler 14 "Nicht genügend Zeichenfolgenspeicher", da schon lange vor dem Auftreten dieser Fehlermeldung der gesamte Speicher des Computers aufgebraucht ist. Nach dem 22. Schleifendurchlauf besitzt die in der Variable *TestZF* abgelegte Zeichenfolge eine Länge von 20'971'520 Zeichen ($5 \cdot 2^{22}$), was sehr viel Arbeitsspeicher erfordert, sodass Windows bereits Speicherbereiche in die Auslagerungsdatei schreiben muss. Der nächste Schleifendurchlauf ist für meinen Testrechner zu viel des Guten und wird mit der Systemfehlermeldung "Nicht genügend virtueller Speicher" reklamiert.

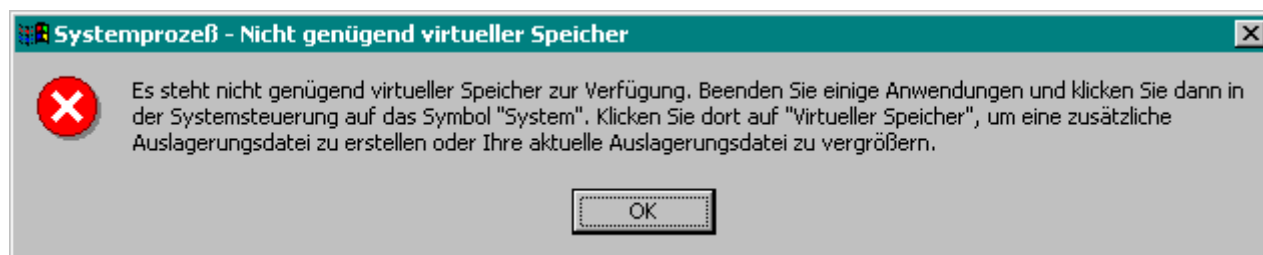


Abbildung: Fehlermeldung "Systemprozess - Nicht genügend virtueller Speicher"

Es gibt jedoch eine andere, einfachere Methode, den Laufzeitfehler "Nicht genügend Zeichenfolgenspeicher" zu erzeugen. Schreiben Sie die Anweisung

```
x = String(40000000, "x")
```

in das Direktfenster des VBA-Editors und drücken Sie die *Eingabe*-Taste. Die Fehlermeldung müsste sofort erscheinen, wobei VBA schon gar nicht erst versucht, Speicherplatz zu reservieren.



Runtime Error No. 14 "Out of string space"

"Out of string space (error 14)" (Auszug aus der Visual Basic-Referenz)

Your system may have run out of memory, which has prevented a string from being allocated. Similarly, expressions requiring that temporary strings be created for evaluation may be causing this error. For example, the following code will cause an "Out of string space" error:

```
MyString = "Hello"
For Count = 1 To 100
    MyString = MyString & MyString
Next Count
```

Visual Basic lets you allocate a string with a maximum length of 65,535 characters. However, in executing statements at run time, the host application needs to use some string space for overhead. The amount of overhead varies among hosts, but should not exceed 50 bytes. If you need to allocate a string of the maximum length your host can support, reduce the string size by 50 bytes, then increase the length incrementally until this error is generated again. The value immediately preceding the error represents the host's maximum string length.

```
'Start with (65535-50).
'On successive runs, increment length until "Out of string space" error occurs.
Dim MyString As String * 65485
Sub MySub
    MyString = "string" 'Error occurs here when the maximum length is exceeded.
End Sub
```



Laufzeitfehler Nr. 28 "Nicht genügend Stapelspeicher"

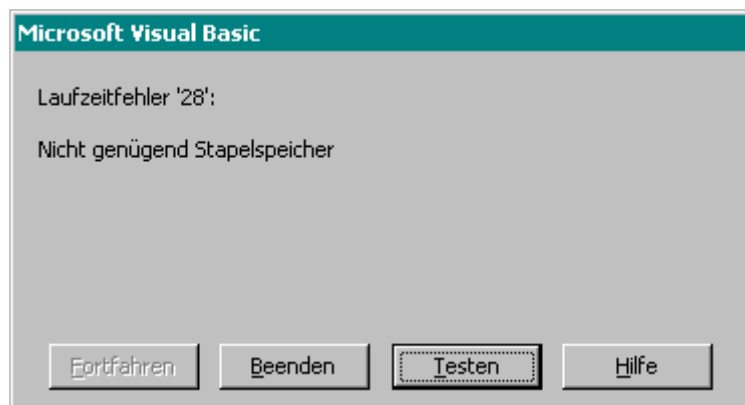


Abbildung: Laufzeitfehler 28 "Nicht genügend Stapelspeicher"

"Nicht genügend Stapelspeicher (Fehler 28)" (Auszug aus der Visual Basic-Referenz)

Der Stapelspeicher ist ein Bereich des Arbeitsspeichers, der je nach Anforderungen des momentan ausgeführten Programms dynamisch seine Grösse verändert. Dieser Fehler hat die folgenden Ursachen und Lösungen:

Es sind zu viele Aufrufe von Function-, Sub- oder Property-Prozeduren aktiv.

» Überprüfen Sie, ob Prozeduren eventuell zu tief verschachtelt sind. Dies kann insbesondere bei rekursiven Prozeduren (also bei Prozeduren, die sich selbst aufrufen) auftreten. Sorgen Sie dafür, dass rekursive Prozeduren ordnungsgemäss beendet werden. Mit dem Dialogfeld Aufrufeliste können Sie überprüfen, welche Prozeduren momentan aktiv sind (also Stapelspeicher benötigen).

Für lokale Variablen steht nicht genügend lokaler Variablenspeicher zur Verfügung.

» Versuchen Sie statt dessen, einige Variablen auf Modulebene zu deklarieren. Sie können auch alle Variablen in der betreffenden Prozedur als statisch deklarieren, indem Sie vor die Schlüsselwörter Property, Sub oder Function das Schlüsselwort Static setzen. Sie können mit der Static-Anweisung auch einzelne Variablen innerhalb einer Prozedur als Static-Variablen deklarieren.

Es sind zu viele Zeichenfolgen fester Länge vorhanden.

» Zwar erfolgen in einer Prozedur Zugriffe auf Zeichenfolgen fester Länge schneller als auf Zeichenfolgen variabler Länge, aber Zeichenfolgen fester Länge benötigen mehr Stapelspeicher, da die Daten der Zeichenfolge auf dem Stapel abgelegt werden. Versuchen Sie, einige Zeichenfolgen fester Länge als Zeichenfolgen variabler Länge zu deklarieren, da bei diesen nur der Zeichenfolgendeskriptor (also der Zeiger auf die Zeichenfolge und nicht die Daten selbst) auf dem Stapel abgelegt wird. Sie können die Zeichenfolge auch auf Modulebene definieren. Sie benötigt dann keinen Stapelspeicher, da auf Modulebene deklarierte Variablen standardmässig den Typ Public haben, d.h. auf die Zeichenfolge kann von allen Prozeduren im Modul zugegriffen werden.

Es gibt zu viele verschachtelte Aufrufe der DoEvents-Funktion.

» Überprüfen Sie mit dem Dialogfeld Aufrufeliste, welche Prozeduren zur Zeit aktiv sind (also Stapelspeicher benötigen).

Ihr Code hat eine Ereigniskette ausgelöst.

» Eine Ereigniskette entsteht, wenn ein Ereignis ausgelöst wird, das eine weitere Ereignisprozedur aufruft, die sich bereits auf dem Stapel befindet. Eine Ereigniskette ähnelt einem nicht abgebrochenen Aufruf einer rekursiven Prozedur, ist jedoch weniger offensichtlich, da der Aufruf durch Visual Basic erfolgt und nicht explizit durch Ihren Code. Überprüfen Sie mit dem Dialogfeld Aufrufeliste, welche Prozeduren zur Zeit aktiv sind (also Stapelspeicher benötigen).

Hinweis

Das Dialogfeld Aufrufeliste wird angezeigt, wenn Sie im Testfenster auf die Schaltfläche mit den Auslassungspunkten rechts neben dem Feld Prozedur klicken oder den Befehl Aufrufeliste wählen.

Beispiel

Der Laufzeitfehler 28 "Nicht genügend Stapelspeicher" kann ganz einfach erzeugt werden. Man braucht lediglich in einer Prozedur die gleiche Prozedur aufzurufen. Dies führt zu einer unendlichen Rekursion:

```
Sub MyProcedure ()
    Call MyProcedure
End Sub
```



Runtime Error No. 28 "Out of stack space"

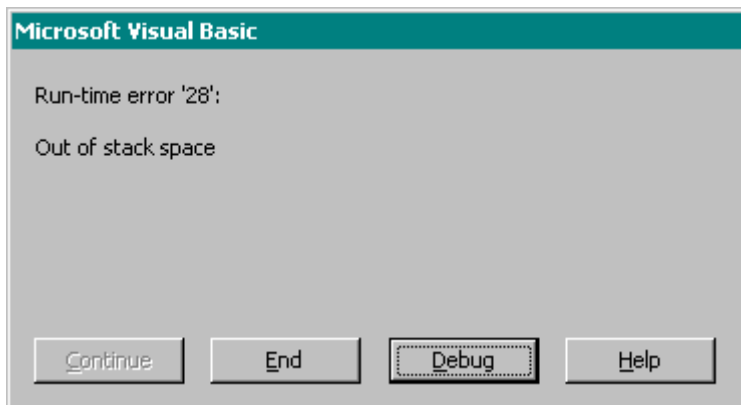


Abbildung: Laufzeitfehler 28 "Out of stack space"

"Out of stack space (error 28)" (Auszug aus der Visual Basic-Referenz)

Possible causes for this error are:

- » Too many active Function or Sub calls. Check that general recursive procedures are not nested too deeply and that they terminate properly.
- » Local variables require more local variable space than is available. Try declaring some variables at the module level instead. You can also declare all variables in the procedure static by preceding the Property, Sub, or Function keyword with Static. Or, you can use the Static statement to declare individual static variables within procedures.
- » Fixed-length strings use more stack space than variable-length strings. Try redefining some of your fixed-length strings as variable-length strings.
- » Too many nested DoEvents statements.

Info

Use the Calls dialog box to view which procedures are active (on the stack). To display the Calls dialog box, select the button to the right of the Procedures box in the Debug window.

▲
TOP

Speicher-Fehlermeldungen beim Excel-Start

When you start any of the Microsoft Office 2000 programs, for example, Microsoft Excel 2000, you receive an error message similar to the following:

```
C:\Program Files\Microsoft Office\Office\EXCEL.EXE
There is not enough free memory to run this program.
Quit one or more programs, and then try again.
```

If you click OK, you receive the following error message:

```
Error Starting Program
There is not enough memory to start EXCEL.EXE.
Quit some programs, and then try again.
```

This behavior occurs when you attempt to start any of the programs included in Office on a computer running Microsoft Windows 95 or Windows 98 and you have either, turned off virtual memory, or set the maximum size of the virtual memory too low. You may also receive these error messages when you start an Office program on a

computer running Microsoft Windows NT if the paging file is too small.

To resolve this behavior, turn on virtual memory or increase the maximum size of the virtual memory setting, under Windows 95 and Windows 98, or increase the size of the Windows NT paging file. Office programs require that virtual memory be turned on when you run them under Windows 95 or Windows 98.

In Windows 95, Windows 98 and Windows NT, virtual memory allows the computer to use hard disk space as random access memory (RAM). The computer uses virtual memory to augment the normal RAM that is installed on the computer.

If for some reason you turned off virtual memory in Windows 95 or Windows 98, or the maximum setting for virtual memory is too low, or if you reduced the size of the paging file in Windows NT, Office programs may not start correctly or at all. This behavior may occur even if the computer has a large amount of installed, RAM 32 megabytes (MB) or more. The amount of virtual memory required varies depending on the amount of installed RAM. For a system that has 16 MB of RAM, a minimum of approximately 16 MB of virtual memory is required to start an Office program. However, increasing the amount of virtual memory will also greatly increase the performance of your computer under Windows.

WIN95/WIN98: How Windows 95/98 Manages Virtual Memory

<http://support.microsoft.com/?scid=kb;en-us;128327>

▲
TOP

Speicher-Probleme wegen beschädigten Arbeitsblättern

Inhalt

- » [Kurzbeschreibung des Problems](#)
- » [So erkennen Sie das Problem](#)
- » [So beheben Sie das Problem](#)
- » [Letzte Zelle eines Arbeitsblattes von Hand zurücksetzen](#)
- » [Letzte Zelle eines Arbeitsblattes mittels VBA-Makro zurücksetzen](#)
- » [Letzte Zelle eines Arbeitsblattes mittels Tool zurücksetzen](#)

▲

Kurzbeschreibung des Problems

In diesem Kapitel möchte ich ein Phänomen in Excel vorstellen, das gewöhnlich auf ein ernstes Problem im Arbeitsspeicher hindeutet. Ich selbst bin diesem Problem schon mehrmals begegnet. Es ist durchaus möglich, dass auch Sie es schon gesehen aber vielleicht nicht darauf geachtet haben.

Es kann vorkommen, dass die sogenannte *ActiveCell Table* eines Tabellenblattes beschädigt ist. Es handelt sich hierbei um eine Excel-interne Tabelle, die im Arbeitsspeicher geführt wird und Informationen über die Zellen des benutzten Bereiches enthält. Wenn sich in dieser Tabelle fehlerhafte Informationen befinden (beispielsweise wenn steht, dass sich die letzte Zelle ausserhalb des benutzten Zellbereiches befindet), so werden schwerwiegende Probleme auftreten. Das Arbeitsblatt reagiert oft sehr träge, Excel stürzt unerwartet ab und die Arbeitsmappe kann defekt werden.

Wie Sie dieses Problem erkennen und beheben können, wird in den folgenden Abschnitten detailliert vorgestellt.

Weitere Informationen

Informationen von Microsoft zu dem hier beschriebenen Problem finden Sie hier:

XL2000: Excel Considers Cell Outside Data Area As Last Cell in the Worksheet

<http://support.microsoft.com/?scid=kb;en-us;230998> (Dead Link!)

XL: How to reset the last cell in Excel (ex "XL2000: Resetting the Last Cell Fixes Memory and Printing Problems")

<http://support.microsoft.com/?scid=kb;en-us;244435>

XL97: Deleting a Cell Changes the UsedRange Property

<http://support.microsoft.com/?scid=kb;en-us;232094>



So erkennen Sie das Problem

Wie gesagt reagiert Microsoft Excel bei einem bestimmten Tabellenblatt sehr träge. Oft ist nur ein einziges Blatt der Arbeitsmappe betroffen.

Alle diese Symptome sind praktisch immer anzutreffen:

- » Das Navigieren des Zellcursors anhand den Pfeiltasten oder der Eingabetaste ist sehr langsam.
- » Je weiter unten man sich im Tabellenblatt befindet, desto langsamer reagiert Excel.
- » Das Scrollen mit der horizontalen oder vertikalen Bildlaufleiste ist sehr langsam.
- » Die Aktualisierung der Anzeige ist generell sehr langsam.
- » Das Verschieben des Excel-Anwendungsfensters läuft ruckartig ab, da der Bildaufbau des Fensters langsam ist.
- » Die CPU-Belastung ist beim Selektieren einer Zelle anhand einer Pfeiltaste oder der Eingabetaste extrem hoch, d.h. 95 % oder mehr. Normalerweise beträgt die CPU-Belastung bei diesem Vorgang höchstens 6 %.
- » Durch Drücken der Tastenkombination *Strg+Ende* (oder Menübefehl *Bearbeiten/Gehe zu*, Schaltfläche *Inhalte*, Option *Letzte Zelle*) wird die letzte Zelle des Tabellenblattes selektiert. Dies ist jedoch oft nicht diejenige Zelle, die Sie eigentlich erwarten würden. Beispielsweise wird eine Zelle der letzten Spalte IV selektiert, was viel zu weit rechts der Tabelle ist (Anmerkung des Autors: Vermutlich würde Excel noch viel weiter nach rechts gehen, würde die Tabelle nicht bei 256 Spalten aufhören).
- » Der benutzte Zellbereich ist viel zu gross. Oft umfasst er die gesamte Tabelle, also sämtliche Zellen des Tabellenblattes. Die Grösse des benutzten Zellbereiches können Sie im VBA-Editor im Direktfenster anhand der Anweisung `"?ActiveSheet.UsedRange.Cells.Count"` abfragen. Ist der Bereich so gross wie das ganze Blatt, gibt die Abfrage die Zahl 16'777'216 zurück.
- » Das Kopieren des Tabellenblattes führt zum sofortigen Excel-Absturz.

Das Arbeitsblatt kann - abgesehen vom Kopieren - ganz normal bearbeitet werden, ohne dass dabei Fehlermeldungen oder Abstürze auftreten. Das Tabellenblatt kann oft noch problemlos verschoben werden. Sie können somit das defekte Tabellenblatt in eine neue oder andere Mappe verschieben. Allerdings wird dadurch das Problem nicht behoben. Das heisst, dass das Tabellenblatt auch in der anderen Arbeitsmappe sehr langsam reagieren wird bzw. die *ActiveCell Table* immer noch defekt ist.

Weitere Informationen

Umfassende Informationen zum Vorgang "Kopieren eines Arbeitsblattes" gibt es auf folgender Seite:



Anmerkung des Autors

Die bisher von mir untersuchten defekten Tabellenblätter waren alle sehr ähnlich aufgebaut. Oft enthielt das Blatt einen fixierten Bereich (Menübefehl *Fenster/Fixieren*) und viele ausgeblendete Zeilen. Das kann Zufall sein; man weiss es nicht so genau.



So beheben Sie das Problem

Leider kann das Problem nicht immer behoben werden. Unter Umständen muss das fehlerhafte Tabellenblatt von Grund auf neu erstellt werden. Wie weiter oben erwähnt wurde, glaubt Excel irrtümlich, dass sich die letzte Zelle ausserhalb des Datenbereiches befindet. Durch explizites Löschen von ganz bestimmten Zellen kann man Excel dazu veranlassen, die Adresse der letzten Zelle neu zu bestimmen. Aus technischer Sicht ist das keine grosse Sache. Mit anderen Worten ist die Lösung dieses Problems eigentlich sehr einfach, würden während der Problembhebung nicht häufig Fehlermeldungen wegen ungenügendem Arbeitsspeicher auftreten (mehr dazu

weiter unten).

Das Problem mit der fehlerhaften Tabelle lässt sich möglicherweise durch Zurücksetzen der letzten Zellen lösen, wobei man dies von Hand oder automatisiert mittels Programm machen kann. Bei der automatisierten Variante gibt es wiederum zwei Wege: Einsetzen eines eigenen VBA-Programmes oder eines Tools. Es stehen somit drei Lösungsmöglichkeiten zur Verfügung:

- 1) [Letzte Zelle eines Arbeitsblattes von Hand zurücksetzen](#)
- 2) [Letzte Zelle eines Arbeitsblattes mittels VBA-Makro zurücksetzen](#)
- 3) [Letzte Zelle eines Arbeitsblattes mittels Tool zurücksetzen](#)

Wichtig!

Bitte beachten Sie folgende Punkte:

- » Erstellen Sie unbedingt eine Sicherungskopie der Arbeitsmappe, bevor Sie mit der Problembhebung beginnen. Falls die Arbeitsmappe durch das Zurücksetzen der letzten Zelle vollständig defekt wird, besitzen Sie wenigstens eine aktuelle Version der Mappe.
- » Stellen Sie Excel möglichst viel Arbeitsspeicher zur Verfügung. Schliessen Sie daher alle geöffneten Arbeitsmappen (natürlich abgesehen die vom Problem betroffene Mappe). Deaktivieren Sie auch alle Add-Ins, die nicht zwingend geladen sein müssen.



Letzte Zelle eines Arbeitsblattes von Hand zurücksetzen

Wenn Sie die letzte Zelle des betroffenen Tabellenblattes von Hand zurücksetzen wollen, gehen Sie wie folgt vor:

1. Selektieren Sie auf dem fehlerhaften Tabellenblatt diejenige Zelle, die Ihrer Meinung nach die letzte Zelle darstellt. Diese befindet sich in der rechten unteren Ecke des benutzten Zellbereiches. Beachten Sie, dass durch die Tastenkombination *Strg+Ende* (oder Menübefehl *Bearbeiten/Gehe zu*, Schaltfläche *Inhalte*, Option *Letzte Zelle*) die aktuelle letzte Zelle markiert wird. Diese ist jedoch meistens nicht die effektiv von Ihnen gewünschte letzte Zelle.
2. Merken Sie sich die Adresse der selektierten Zelle (sie steht im Namenfeld, welches sich ganz links in der Bearbeitungsleiste befindet).
3. Geben Sie im Namenfeld den Adressbereich der **Spalten** ein, die sich **rechts** von der letzten Zelle befinden.
Beispiele:
 - Wenn die Adresse der letzten Zelle T245 lautet, geben Sie im Namenfeld U:IV ein.
 - Wenn die Adresse der letzten Zelle CZ3469 lautet, geben Sie im Namenfeld DA:IV ein.
4. Führen Sie den Menübefehl *Bearbeiten/Löschen/Alles* aus.
5. Führen Sie den Menübefehl *Bearbeiten/Zellen löschen* aus.
6. Geben Sie im Namenfeld den Adressbereich der **Zeilen** ein, die sich **unterhalb** der letzten Zelle befinden.
Beispiele:
 - Wenn die Adresse der letzten Zelle T245 lautet, geben Sie im Namenfeld 246:65536 ein.
 - Wenn die Adresse der letzten Zelle CZ3469 lautet, geben Sie im Namenfeld 3470:65536 ein.
7. Führen Sie den Menübefehl *Bearbeiten/Löschen/Alles* aus.
8. Führen Sie den Menübefehl *Bearbeiten/Zellen löschen* aus.
9. Drücken Sie die Tastenkombination *Strg+Pos1* oder selektieren Sie von Hand die Zelle A1.
10. Speichern Sie die Arbeitsmappe **unter einem anderen Dateinamen**.

Nach diesen Arbeitsschritten müsste die letzte Zelle neu gesetzt sein. Da die durchgeführten Aktionen sehr speicherintensiv sind, sollten Sie die Mappe zum Schluss schliessen und neu öffnen.

Mögliche Probleme beim Zurücksetzen der letzten Zelle

Es kann sein, dass beim Schritt 4, 5, 7 oder 8 eine Fehlermeldung infolge ungenügendem Speicher erscheint. Erfahrungsgemäss können insbesondere diese Fehlermeldungen auftreten:

Fehlermeldung "Nicht genügend Arbeitsspeicher. Ohne Rückgängig weiter?"

Diese Fehlermeldung können Sie einfach mit OK bestätigen.

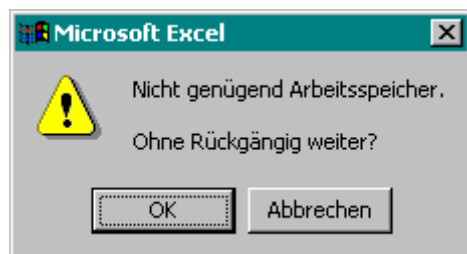


Abbildung: Fehlermeldung "Nicht genügend Arbeitsspeicher. Ohne Rückgängig weiter?"

Fehlermeldung "Nicht genügend Arbeitsspeicher."

Diese Fehlermeldung können Sie einfach mit OK bestätigen.



Abbildung: Fehlermeldung "Nicht genügend Arbeitsspeicher"

Fehlermeldung "Nicht genug Speicher für eine vollständige Anzeige"

Bei dieser Fehlermeldung sollten Sie den Vorgang des Zurücksetzens abbrechen. Schliessen Sie die Mappe, beenden Sie Excel, starten Sie Excel neu und öffnen Sie dann die Mappe wieder.

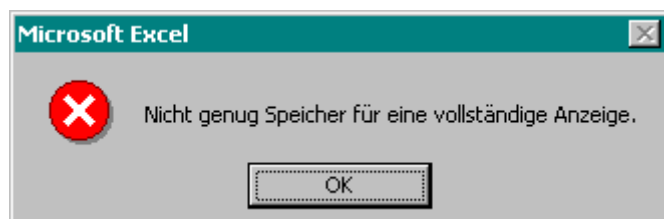


Abbildung: Fehlermeldung "Nicht genug Speicher für eine vollständige Anzeige"

Letzte Zelle eines Arbeitsblattes mittels VBA-Makro zurücksetzen

Mit dem unten vorgestellten VBA-Makro wird die letzte Zelle eines Tabellenblattes automatisch zurückgesetzt.

Bitte beachten Sie unbedingt folgende Punkte:

- Erstellen Sie zuerst eine Sicherungskopie Ihrer Arbeitsmappe.
- Entfernen Sie vor dem Makrostart einen allfällig vorhandenen Blatt-Schutz (Menübefehl *Extras/Schutz/Blattschutz aufheben*).
- Das Makro berücksichtigt keine eingefügten Objekte (AutoFormen, Steuerelemente, Bilder, eingebettete Objekte etc.), die sich rechts oder unterhalb der letzten Zelle befinden oder grösser als der benutzten Zellbereich sind. Diese Objekte werden entweder gelöscht oder 'gestaucht' dargestellt (d.h. die Objektgrösse wird geändert).
- Leere Zellen, die Funktionalitäten wie Gültigkeitsprüfungen, bedingte Formatierungen, Namen oder dergleichen besitzen, sich aber ausserhalb des benutzten Zellbereiches befinden, werden unwiderruflich gelöscht.
- Benutzte Zellen, die sich in einer ausgeblendeten Spalte oder Zeile befinden, werden vom Makro nicht erkannt. Ausgeblendete Zeilen und Spalten sollten daher **vor dem Makrostart** eingeblendet werden.
- Das Makro funktioniert nicht, wenn die letzte Zelle des benutzten Zellbereiches eine verbundene Zelle ist. Es tritt in diesem Fall der Laufzeitfehler 1004 auf.
- Sie sollten das Makro mehrmals nacheinander ausführen, am besten vier Mal. Aufgrund des verwendeten Verfahrens ist es möglich, dass sich die letzte Zelle bei wiederholter Codeausführung ändert. (Anmerkung des Autors: Gemäss meinen Erfahrungen findet nach vier Ausführungen keine Änderung mehr statt.)
- Kontrollieren Sie nach der Makroausführung, ob das Tabellenblatt immer noch wie erwartet aussieht (sprich die Darstellung).
- Nachdem Sie das Makro so lange ausgeführt haben, bis sich die Adresse der letzten Zelle nicht mehr ändert, müssen Sie die Arbeitsmappe speichern. Erst durch das Speichern erstellt Microsoft Excel die interne *Active Cell Table* neu und korrigiert dadurch ein Speicherproblem, vorher nicht. Schliessen und öffnen Sie danach die Arbeitsmappe.

Nachfolgend der VBA-Programmcode. Fügen Sie ihn in ein Codemodul eines beliebigen VBA-Projektes ein, aktivieren Sie das betroffene Tabellenblatt der Arbeitsmappe und führen Sie den Code aus.

```
Public Sub ResetLastCell()
'-----
' Diese Prozedur setzt die letzte Zelle des aktiven Tabellenblattes zurück
' Original-Code: Microsoft Corporation
' Anpassungen: Philipp von Wartburg
'-----

Dim rngLastCell As Range
Dim rowstep As Integer
Dim colstep As Integer
'Aktuelle letzte Zelle des aktiven Blattes merken
Set rngLastCell = Cells.SpecialCells(xlLastCell)
'Adresse der aktuellen letzten Zelle in der Statusleiste anzeigen
Application.StatusBar = "Letzte Zelle: " & rngLastCell.Address
'Variablen rowstep und colstep initialisieren
rowstep = -1
colstep = -1
'Schleife durchlaufen bis letzte Zelle gefunden oder Zelle A1 erreicht ist
While (rowstep + colstep <> 0) And (rngLastCell.Address <> "$A$1")
'Prüfen ob die momentane Spalte eine Zelle mit Daten enthält
If Application.CountA(Range(Cells(1, rngLastCell.Column), rngLastCell)) > 0 Then _
    colstep = 0
'Prüfen ob die momentane Zeile eine Zelle mit Daten enthält
If Application.CountA(Range(Cells(rngLastCell.Row, 1), rngLastCell)) > 0 Then _
    rowstep = 0
'Objektpointer rngLastCell auf die neue Position setzen
Set rngLastCell = rngLastCell.Offset(rowstep, colstep)
'Adresse der "neuen" letzten Zelle in der Statusleiste anzeigen
Application.StatusBar = "Letzte Zelle: " & rngLastCell.Address
Wend
'Unbenutzte Spalten leeren und dann löschen
With Range(Cells(1, rngLastCell.Column + 1), "IV65536")
    Application.StatusBar = "Lösche Spalten: " & .Address
    .Clear
    .Delete
End With
'Unbenutzte Zeilen leeren und dann löschen
With Rows(rngLastCell.Row + 1 & ":65536")
    Application.StatusBar = "Lösche Zeilen: " & .Address
    .Clear
    .Delete
End With
```

```
Set rngLastCell = Nothing
'Neue letzte Zelle selektieren
Cells.SpecialCells(xlLastCell).Select
'Statusleiste zurücksetzen und neue letzte Zelle ausgeben
Application.StatusBar = False
MsgBox "Die neue letzte Zelle besitzt die Adresse " & _
Cells.SpecialCells(xlLastCell).Address(False, False) & ".", vbInformation
End Sub
```



Letzte Zelle eines Arbeitsblattes mittels Tool zurücksetzen

Beschreibung folgt...



Tip: Setzen Sie ein geeignetes Tool ein

Mit dem Freeware Tool **UsedRange Optimizer** kann man ebenfalls die letzte benutzte Zelle zurücksetzen, und zudem den benutzten Zellbereich auf ein Optimum reduzieren. Das Tool verwendet ausgeklügelte Mechanismen, die garantieren, dass auch ausgeblendete Zellen, Objekte wie AutoFormen und ActiveX-Steuerelemente, verbundene Zellen und andere Spezialitäten wie Zellen mit Gültigkeitsprüfungen und bedingten Formatierungen korrekt berücksichtigt werden.

Der UsedRange Optimizer ist ein Tool aus der [CEFA-Toolbox](#):

» [UsedRange Optimizer](#)



TOP

Speicher-Probleme wegen Speicher-Lecks (Memory Leaks)

Allgemeines über Speicher-Lecks

In Microsoft Excel gibt es ein paar Bugs, die dazu führen, dass der Speicher beispielsweise beim Schliessen einer Arbeitsmappe nicht wie erwartet freigegeben wird. In den nachfolgenden Abschnitten werden die grössten Probleme vorgestellt.



Speicher-Leck bei Arbeitsmappen mit eingebetteten Objekten

Ein meiner Meinung nach ernstzunehmendes Problem existiert bei Arbeitsmappen, welche eingebettete Objekte enthalten. Als eingebettetes Objekt wird unter anderem ein ActiveX-Control, d.h. beispielsweise das Steuerelement "Schaltfläche" (CommandButton) aus der Steuerelement-Toolbox, eine in ein Tabellenblatt eingefügte ClipArt-Grafik oder ein eingebettetes Word-Dokument bezeichnet. Diese Objekte belegen Speicherplatz, wenn die Arbeitsmappe in Microsoft Excel geöffnet ist.

Normalerweise wird der von einer Arbeitsmappe beanspruchte Speicher beim Schliessen der Mappe wieder freigegeben. Wegen eines Bugs ist dies bei Arbeitsmappen mit eingebetteten Objekten nicht der Fall: Der Speicher dieser Arbeitsmappen wird nicht geleert. Wenn man nun eine Mappe mit eingebetteten Objekten mehrmals nacheinander öffnet und schliesst, ohne dabei Microsoft Excel zwischendurch zu beenden und neu zu starten, wird immer mehr Arbeitsspeicher beansprucht. Das geht so lange gut bis der gesamte Speicher verbraucht ist. Als Folge erscheint oft diese Fehlermeldung:

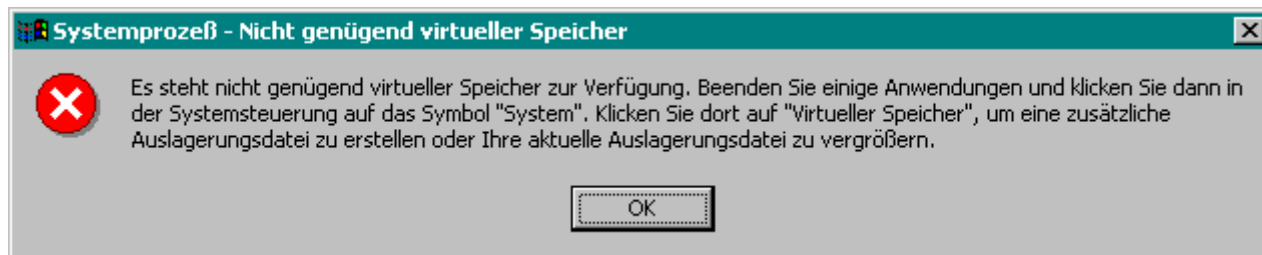


Abbildung: Fehlermeldung "Systemprozess - Nicht genügend virtueller Speicher"

Bei meinen mit Microsoft Excel 97 durchgeführten Tests kam es des öfteren vor, dass keine Fehlermeldung erschien. Das Fehlen von verfügbarem Arbeitsspeicher war jedoch gut zu erkennen, da beim Öffnen einer Arbeitsmappe einzelne Elemente der Excel-Arbeitsoberfläche nicht mehr korrekt dargestellt werden konnten. Beispielsweise fehlten mehrere Symbole von Symbolleisten-Schaltflächen und von Menübefehlen.

Wie weiter oben erwähnt wurde, müsste eigentlich der von einer Mappe beanspruchte Speicher beim Schliessen der Mappe geleert und freigegeben werden, was jedoch nicht korrekt funktioniert. Es kann sogar passieren, dass selbst beim Beenden von Microsoft Excel der Speicher nicht freigegeben wird. Die folgende Abbildung zeigt dies deutlich anhand des Utilities FreeMem:

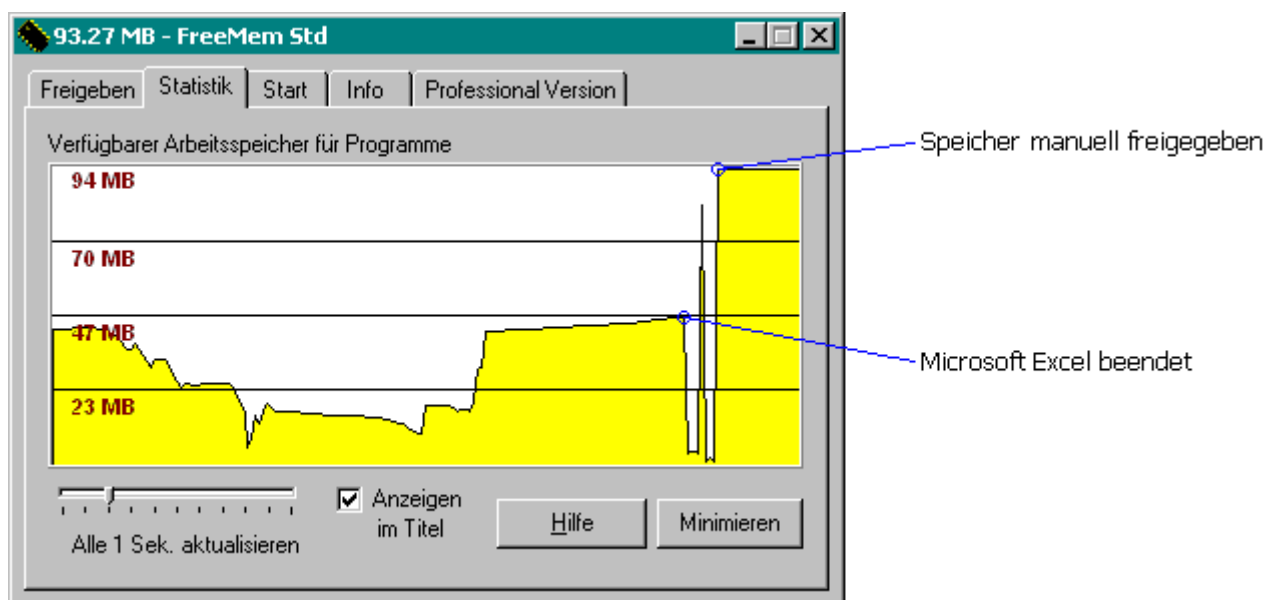


Abbildung: Utility FreeMem: Verfügbarer Arbeitsspeicher nach Beenden von Excel

Nachdem Microsoft Excel beendet wurde, war lediglich ungefähr 47 MB Arbeitsspeicher frei. Erst als der Speicher manuell freigegeben wurde, stand wieder so viel Arbeitsspeicher zur Verfügung, wie es vor dem Start von Excel der Fall war. Der Speicher wurde mit Hilfe von FreeMem freigegeben, einem kleinen Utility zum Belegen und Freigeben von Arbeitsspeicher (weitere Informationen zu FreeMem finden Sie [hier](#)).

In einem englischen Artikel von Microsoft wird das Problem folgendermassen beschrieben:

"When opening a workbook that contains an ActiveX/OCX control, the Excel process acquires a minimum of 20 private bytes. It does not release this memory when it closes a workbook, but only releases it when the Excel process ends. The amount of the private byte leak can be greater, depending on which ActiveX/OCX control(s) you use in a workbook. You can observe this behavior within Windows NT Performance Monitor or other similar tool."

Dieses Speicherleck-Problem kann nur durch einen speziellen Hotfix behoben werden. Der Fehler wurde in keinem verfügbaren Service Pack von Microsoft Office 97 korrigiert.

Mehr zu diesem Problem und dem Hotfix erfahren Sie hier:

XL97: "Out of Virtual Memory" Opening Many Workbooks with ActiveX or OCX Controls
<http://support.microsoft.com/?scid=kb;en-us;238570>

Anmerkung des Autors

Mehrere Benutzer von neueren Excelversionen wie u.a. Microsoft Excel 2000 haben mir mitgeteilt, dass auch diese

Versionen gewisse Probleme mit dem freigeben von Arbeitsspeicher besitzen.



Weitere Speicher-Lecks in Microsoft Excel

Hier ein paar Links zu Microsoft Knowledge Base-Artikeln mit Informationen zu weiteren Speicher-Lecks:

XL97/XL2000/XL2002: Memory Leak Occurs When You Query an Open Excel Worksheet Using ADO

<http://support.microsoft.com/?scid=kb;en-us;319998>

XL97/XL2000: Memory Leak When Excel Calculates User-Defined Functions

<http://support.microsoft.com/?scid=kb;en-us;265023>

» Dieses Problem wurde im Microsoft Excel 97 Service Release 2 und im Microsoft Office 2000 Service Pack 2 behoben.

OFFICE: PageSetup Object Causes Memory Leak and Hangs System

<http://support.microsoft.com/?scid=kb;en-us;192869>

OFFICE97: Speicherprobleme bei Verwenden von Hyperlinks zwischen Programmen

<http://support.microsoft.com/?scid=kb;de;157736>

OFFICE97: Memory Problems When Hyperlinking Between Programs

<http://support.microsoft.com/?scid=kb;en-us;157736>

OFFICE2000/OFFICEXP: A GDI Resource Leak Occurs When a Visual Basic for Applications UserForm Is Displayed

<http://support.microsoft.com/?scid=kb;en-us;321242>



ToDo...

- Replacing external link formulae
- Some external data retrieval actions, particularly repeatedly reading text files. Excel seems to store only unique strings rather than multiple occurrences of the same string, but does not seem to remove a unique string when it is no longer in use.
- Inserting and deleting graphic objects such as bitmaps or JPEGs.
- Not destroying objects in VBA by setting the object variables to nothing in the correct sequence so that you avoid "orphaned" objects. In theory this is not necessary, but sometimes problems occur when this is not done. Destroy in inside-out container sequence, for example Range then Worksheet then Workbook.



TOP

Möglichkeiten zum Einparen von Speicher

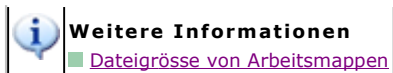
Allgemeines über das Einsparen von Speicher

Es gibt eine Vielzahl verschiedener Möglichkeiten, wie man Microsoft Excel dazu bewegen kann, weniger Speicher zu belegen beziehungsweise wie man Excel mehr Speicher zur Verfügung stellen kann. Diese Möglichkeiten lassen sich in drei Gruppen einteilen:

- 1) [Man stellt Microsoft Excel mehr Speicher zur Verfügung](#)
- 2) [Man gibt belegten Excel-Speicher frei](#)
- 3) [Man reduziert den Speicherbedarf der Arbeitsmappen](#)

Wie man die Größe einer Arbeitsmappendatei reduzieren kann, erfahren Sie auf dieser Seite:





Excel mehr Speicher zur Verfügung stellen

In diesem Abschnitt geht es darum, wie man Microsoft Excel mehr Speicher zur Verfügung stellen kann. Die hier vorgestellten Möglichkeiten zielen darauf ab, Excel so zu konfigurieren, dass möglichst wenig Speicher nach dem Excelstart belegt ist.

VBA-Editor geschlossen lassen

Vermeiden Sie es, den VBA-Editor nach dem Excelstart zu öffnen. Lassen Sie ihn so lange geschlossen wie Sie ihn nicht zwingend benötigen. Sobald der VBA-Editor einmal geöffnet ist, verbleibt er im Arbeitsspeicher. Der vom VBA-Editor belegte Speicher wird erst freigegeben, wenn die Excel-Sitzung beendet wird.

Weitere Informationen dazu finden Sie im Kapitel [Speicherbeanspruchung des VBA-Editors](#).



Anzahl Symbolleisten und Symbolleisten-Steuerelemente reduzieren

Beschreibung folgt...



xlB-Datei defragmentieren

In der xlB-Datei speichert Excel alle Informationen über benutzerdefinierte Symbolleisten, benutzerdefinierte Symbolleisten-Steuerelemente sowie angepasste Symbolleisten und Steuerelemente. Mit der Zeit nimmt die Grösse der xlB-Datei zu, da immer mehr fehlerhafte bzw. irrtümlich liegende Daten in der Datei mitgeführt werden. Dies wegen eines wenig bekannten Bugs. Bekanntlich wird die xlB-Datei automatisch beim Excelstart geladen. Je grösser die Datei ist, desto mehr Arbeitsspeicher wird benötigt. Beispielsweise beansprucht eine 5 MB grosse xlB-Datei ca. 8 MB Speicher. Eine praktisch leere xlB-Datei hingegen belegt sozusagen keinen Speicher (bzw. eine kaum messbare Menge).



Nicht benötigte Add-Ins deaktivieren

Deaktivieren Sie im Add-In-Manager (Menübefehl *Extras/Add-In-Manager*) alle diejenigen Add-Ins, welche Sie gerade nicht unbedingt benötigen. Je weniger Add-Ins aktiviert sind, desto weniger Speicher benötigt Excel. Nach dem Schliessen des Add-In-Managers sollten Sie Excel beenden und neu starten. Der von einem zuvor aktivierten Add-In belegte Speicherplatz wird erst durch das Beenden von Excel freigegeben, und nicht bereits beim Deaktivieren des Add-Ins.



Nur benötigte Arbeitsmappen beim Excelstart öffnen

Beschreibung folgt...



Makro-Arbeitsmappe Personl.xls optimieren

Beschreibung folgt...

Verweise auf DLLs vermeiden, da diese DLL-Dateien in den Speicher geladen werden.



Standard-Mustervorlage für Arbeitsmappen

Beschreibung folgt...

Verweise auf



Excel-Speicher freigeben

Hier geht es darum, wie man die Grösse des von Microsoft Excel belegten Speichers reduzieren kann. Gewöhnlich fordert Excel während der Arbeit weiteren Speicherplatz an, der dann mit irgend welchen Daten belegt wird. Oftmals ist es allerdings der Fall, dass eigentlich gar kein zusätzlicher Speicher benötigt würde, wäre da nicht ein Excel-Bug oder unsachgemäss programmierter VBA-Code. Die hier vorgestellten Möglichkeiten bezwecken, von Excel belegten Speicher freizugeben, damit er anderweitig benutzt werden kann.

Excel beenden und neu starten (1)

Microsoft Excel besitzt ein paar Speicher-Bugs (sogenannte Memory Leaks), die dazu führen, dass unter anderem beim Schliessen einer Arbeitsmappe nicht der gesamte von der Mappe belegte Speicherplatz freigegeben wird (siehe auch Kapitel [Speicher-Probleme wegen Speicher-Lecks](#)). Dies ist insbesondere bei Arbeitsmappen mit eingebetteten ActiveX-Steuerelementen oder anderen Objekten der Fall. Wenn man beispielsweise 200 Schaltflächen in ein Arbeitsblatt einfügt und die Mappe speichert, und anschliessend diese Mappe 10 Mal hintereinander schliesst und neu öffnet, erhöht sich der von Microsoft Excel belegte Speicher laufend.

XL97: "Out of Virtual Memory" Opening Many Workbooks with ActiveX or OCX Controls

<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q238570>

>> Dieses Problem kann mittels Hotfix behoben werden.



Excel beenden und neu starten (2)

DDE-Kanäle, die irrtümlich offengelassen oder wegen eines Programmfehlers nicht geschlossen wurden, werden nur durch Beenden von Excel oder des DDE-Servers geschlossen. Das Beenden des VBA-Makros oder das Schliessen der Arbeitsmappe alleine nützt nichts. Am besten gelegentlich mit "Application.DDETerminateAll" alle DDE-Links trennen.

Nebenbei: Es ist meiner Meinung nach unmöglich herauszufinden, ob DDE-Kanäle offen sind, wieviele offen sind und welche Channel-ID sie besitzen.

-> Behebt gewöhnlich Memory Leaks, die infolge Excel- oder Windows-Bugs bei der Arbeit entstehen (z.B. Freigeben der sogenannten 'Private Bytes' von ActiveX-Controls, Memory Leak bei UserForms, bei PageSetup-Objekt, bei benutzerdefinierten Funktionen und weiteres).



Excel beenden und neu starten (3)

Hinzugefügte Grafiken



Speicherbedarf von Arbeitsmappen reduzieren

Es existiert eine ganze Reihe verschiedener Möglichkeiten, wie man den von Arbeitsmappen beanspruchten Speicher reduzieren kann, ohne dass man auf verwendete Funktionalitäten und Darstellungselemente verzichten muss. Es geht in diesem Abschnitt somit darum, eine Arbeitsmappe so zu gestalten, damit sie möglichst wenig Arbeitsspeicher belegt.

Anzahl Schriften reduzieren



Zellformate vereinheitlichen



Grafiken verkleinern



Eingefügte Hyperlinks durch Tabellenfunktion HYPERLINK ersetzen

Ein in ein Arbeitsblatt eingefügter Hyperlink (Menübefehl *Einfügen/Hyperlink* oder *Strg+K*) beansprucht bedeutend mehr Arbeitsspeicher als ein anhand der Tabellenfunktion HYPERLINK erstellter Hyperlink; nach meinen Erfahrungen ca. 11 mal mehr! Wenn Sie also in einer Arbeitsmappe viele Hyperlinks benutzen, sollten Sie wenn immer möglich die HYPERLINK-Tabellenfunktion verwenden.



Benutzter Zellbereich verkleinern

Die letzte benutzte Zelle des Tabellenblattes kann mit der Tastenkombination *Strg+Ende* selektiert werden, oder über das Dialogfenster "Gehe zu", "Inhalte auswählen", Option "Letzte Zelle".



TÖP

Excel durch Belegen des gesamten Speichers zum Absturz bringen

ACHTUNG

Die hier vorgestellte VBA-Anweisung kann zum Systemabsturz führen. Es werden jedoch keine Hardware-Teile beschädigt, keine Office-Installationen zerstört oder keine Dateien von der Festplatte gelöscht. Dennoch übernehmen Sie selbst die volle Verantwortung, wenn Sie die Anweisung ausführen.

Möchten Sie mal Ihren Computer leiden sehen?

Starten Sie Microsoft Excel 2000 oder Excel 2002/XP und öffnen Sie den VBA-Editor. Geben Sie im Direktfenster folgende Anweisung ein:

?Cells

Nach etwa einer halben Minute wird sich Ihr Computer zu Tode quälen. Microsoft Excel 97 ist ironischerweise immun und zeigt lediglich den Laufzeitfehler Nr. 7 "Nicht genügend Speicher".



Speichernutzung des VBA-Editors

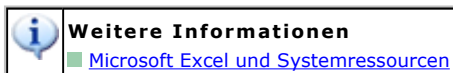
Allgemeines über den VBA-Editor

Der VBA-Editor selbst belegt nicht sehr viel Speicher. In Bezug auf Systemressourcen ist er dagegen einiges anspruchsvoller. Bedingt durch die aufwändige Gestaltung der grafischen Arbeitsoberfläche, die

- viele Fenster,
- Symbolleisten,
- Baumstruktur-Ansichten (Projekt-Explorer, Lokalfenster),
- Tabellenstruktur-Ansichten (Eigenschaftenfenster),
- Symbole (Überwachungsfenster, Objektkatalog) und
- andere grafische Elemente

enthält, wird eine grössere Menge GDI-Speicher benötigt. Beachten Sie daher, dass möglichst nur wenige Code- und Benutzerformular-Fenster gleichzeitig geöffnet sind. Wie viele bzw. welche Fenster geöffnet sind, sehen Sie im VBA-Editor in der Fensterliste des Menüs *Fenster*.

Ausführliche Informationen über Microsoft Excel und die Systemressourcen finden Sie hier:



Speicherbeanspruchung des VBA-Editors

Wie gesagt beansprucht der VBA-Editor als Teil-Anwendung nicht sehr viel Speicher. Die im VBA-Editor dargestellten Informationen jedoch beanspruchen unter Umständen viel Speicher. Bei verschiedenen Tests konnte ich feststellen, dass insbesondere die Anzeige der Projekt-Elemente (Arbeitsmappe, Blätter, Module etc.) im Projekt-Explorer alles andere als sparsam mit Speicherplatz umgehen. Allerdings ist dies nicht die 'Schuld' der Projektfensters, wie Sie später in diesem Kapitel erfahren werden.

Undo-Speicher des VBA-Editors beschreiben...



VBA-Editor aus dem Speicher entfernen

Sobald der VBA-Editor das erste Mal geöffnet wird, wird er in den Speicher geladen und belegt dadurch einen Teil des von Microsoft Excel benutzten Speichers. Durch das Schliessen des VBA-Editors (z.B. über den Menübefehl *Datei/Schliessen und zurück zu Microsoft Excel* oder mit der Tastenkombination *Strg+Q*) wird der belegte Speicherplatz nicht freigegeben. Er bleibt so lange belegt bis Sie Microsoft Excel beenden.

Das Schliessen des VBA-Editors führt somit nicht zu einer Reduktion des benutzten Speichers. Die einzige Möglichkeit zum Einsparen von Arbeitsspeicher bewirkt das Schliessen von Code- und Benutzerformular-Fenstern innerhalb des VBA-Editors.



Generierung der VBA-Projektstruktur (Symbol Table)

Wie oben erwähnt werden im Projekt-Explorer die Elemente des VBA-Projektes angezeigt, was nichts anderes heisst wie, dass die sogenannte "Symbol Table" (auch "Project Table" genannt) grafisch dargestellt wird. Das VBA-Projekt (d.h. die Symbol Table) einer neu angelegten Arbeitsmappe wird nur dann generiert, wenn der VBA-Editor geöffnet ist oder war, während die Exceldatei mit dem VBA-Projekt bearbeitet wurde.

Es gelten folgende Regeln:

- Wenn eine neue Arbeitsmappe angelegt wird, während der **VBA-Editor geschlossen** ist, wird die VBA-Projektstruktur zu dieser Mappe **nicht generiert**.
- Wenn der **VBA-Editor geöffnet** ist und dann eine neue Arbeitsmappe angelegt wird, wird die Projektstruktur **generiert**.
- Wenn der **VBA-Editor geschlossen** ist, dann eine neue Arbeitsmappe angelegt und anschliessend der **VBA-Editor geöffnet** wird, wird die Projektstruktur beim Öffnen des VBA-Editors **generiert**.
- Wenn der **VBA-Editor geöffnet** ist, dann eine neue Arbeitsmappe angelegt und anschliessend der **VBA-Editor geschlossen** wird, wird die Projektstruktur beim Anlegen der neuen Mappe **generiert**, da zu diesem Zeitpunkt der **VBA-Editor geöffnet** war.

Wurde die VBA-Projektstruktur einer Arbeitsmappe einmal generiert, so bleibt sie für immer bestehen. Man kann die Generierung beim Öffnen des VBA-Editors weder verhindern noch rückgängig machen.

Es existiert jedoch eine Möglichkeit, wie man eine Projektstruktur aus einer Arbeitsmappe entfernen kann. Das geht natürlich nur, wenn keine einzige Codezeile dem VBA-Projekt hinzugefügt wurde. Siehe dazu nächster Abschnitt "[Entfernen der VBA-Projektstruktur](#)".



Entfernen der VBA-Projektstruktur

Falls das VBA-Projekt einer Arbeitsmappe noch keine Codezeile enthält, so kann eine bereits generierte Projektstruktur entfernt werden.

Restliche Beschreibung folgt...



Speicherbeanspruchung der VBA-Projektstruktur

Wie bereits erwähnt werden die VBA-Projektstrukturen sämtlicher geöffneten Arbeitsmappen beim Öffnen des VBA-Editors angelegt. Die Struktur wird auch dann generiert, wenn das Projekt Explorer-Fenster beim Öffnen des VBA-Editors geschlossen ist.

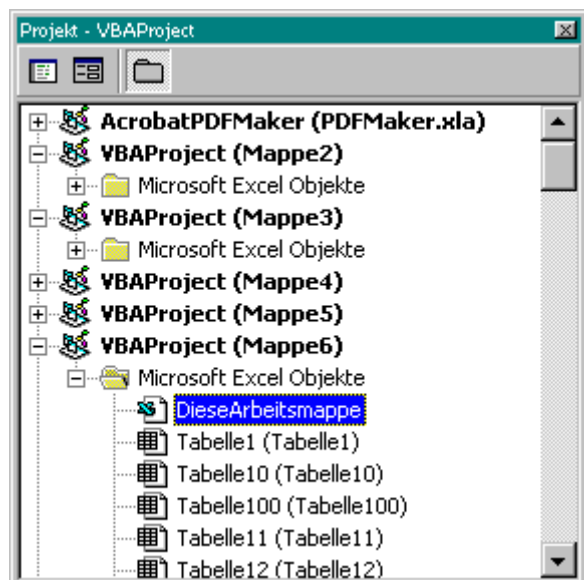


Abbildung: Der Projekt Explorer des VBA-Editors

Je mehr Elemente ein VBA-Projekt enthält, desto mehr Speicher wird für Excel-interne 'Symbol Table' und für die Anzeige des VBA-Projektes im Projekt-Explorer des VBA-Editors benötigt. Die folgenden Testergebnisse verdeutlichen dies:

Arbeitsmappe mit 3 Blättern

Der VBA-Editor mit einer neuen Arbeitsmappe mit 3 Arbeitsblättern beansprucht ca. 1.4 MB Speicher.

Arbeitsmappe mit 255 Blättern

Der VBA-Editor mit einer neuen Arbeitsmappe mit 255 Arbeitsblättern beansprucht ca. 3.4 MB Speicher.

Beachten Sie bitte, dass es keine Rolle spielt, ob die Projektelemente im Listenfenster sichtbar sind oder nicht. Die Speichernutzung ist immer gleich.

Hinweis

Die obigen Tests wurden mit Microsoft Excel 97 durchgeführt und die Speicherwerte mit dem Windows NT Task-Manager ermittelt.



Dateigrösse mit und ohne VBA-Projektstruktur

Die Struktur des VBA-Projektes wird in der Arbeitsmappendatei gespeichert. Wird eine neu angelegte Arbeitsmappe gespeichert, und die Projektstruktur wurde noch nicht generiert, so wird die Arbeitsmappe ohne Projektstruktur gespeichert.

Hier die Werte zweier Beispielmappen (ermittelt unter Microsoft Excel 97):

Beispiel 1: Arbeitsmappe mit 3 Blättern

- Die Grösse einer neuen Arbeitsmappendatei mit 3 Arbeitsblättern und **ohne** Projektstruktur beträgt 14 KB.
- Die Grösse einer neuen Arbeitsmappendatei mit 3 Arbeitsblättern und **mit** Projektstruktur beträgt 17 KB (3 KB mehr als ohne Projektstruktur).

Beispiel 2: Arbeitsmappe mit 255 Blättern

- Die Grösse einer neuen Arbeitsmappendatei mit 255 Arbeitsblättern und **ohne** Projektstruktur beträgt 82 KB.
- Die Grösse einer neuen Arbeitsmappendatei mit 255 Arbeitsblättern und **mit** Projektstruktur beträgt 460 KB (378 KB mehr als ohne Projektstruktur).

Wie man am zweiten Beispiel gut erkennen kann, ist die xls-Datei einer 255 Blätter enthaltenden Arbeitsmappe ohne VBA-Projektstruktur ca. **5.6 mal grösser** als die gleiche Arbeitsmappe ohne VBA-Projektstruktur.

Tipp: VBA-Editor wenn möglich geschlossen lassen

Solange man keine Makros, benutzerdefinierte Excelfunktionen oder anderen VBA-Programmcode in der Arbeitsmappe benötigt, sollte man den VBA-Editor schliessen, bevor man die Arbeitsmappe erstellt oder öffnet. Vor allem Arbeitsmappen mit vielen Blättern behalten auf diese Weise eine kompakte Dateigrösse.



Existenz einer VBA-Projektstruktur herausfinden

Ob eine Arbeitsmappendatei eine Projektstruktur enthält oder nicht, kann man weder in Excel direkt noch im VBA-Editor feststellen. Es gibt jedoch zwei einfache Möglichkeiten. Die erste Möglichkeit stellt das Öffnen der xls-Datei mit einem gewöhnlichen Texteditor dar, beispielsweise mit dem Editor von Windows (Notepad). Die folgende Abbildung zeigt eine Arbeitsmappe, die keine VBA-Projektstruktur enthält:

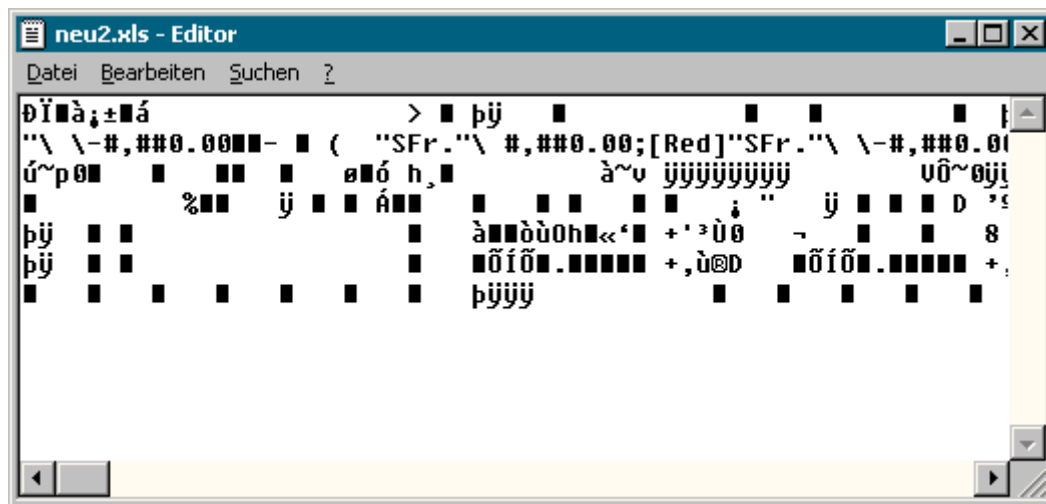


Abbildung: Arbeitsmappe ohne VBA-Projektstruktur (geöffnet mit dem Windows Editor)

Die nächste Abbildung zeigt die gleiche Arbeitsmappendatei, in der die VBA-Projektstruktur enthalten ist:

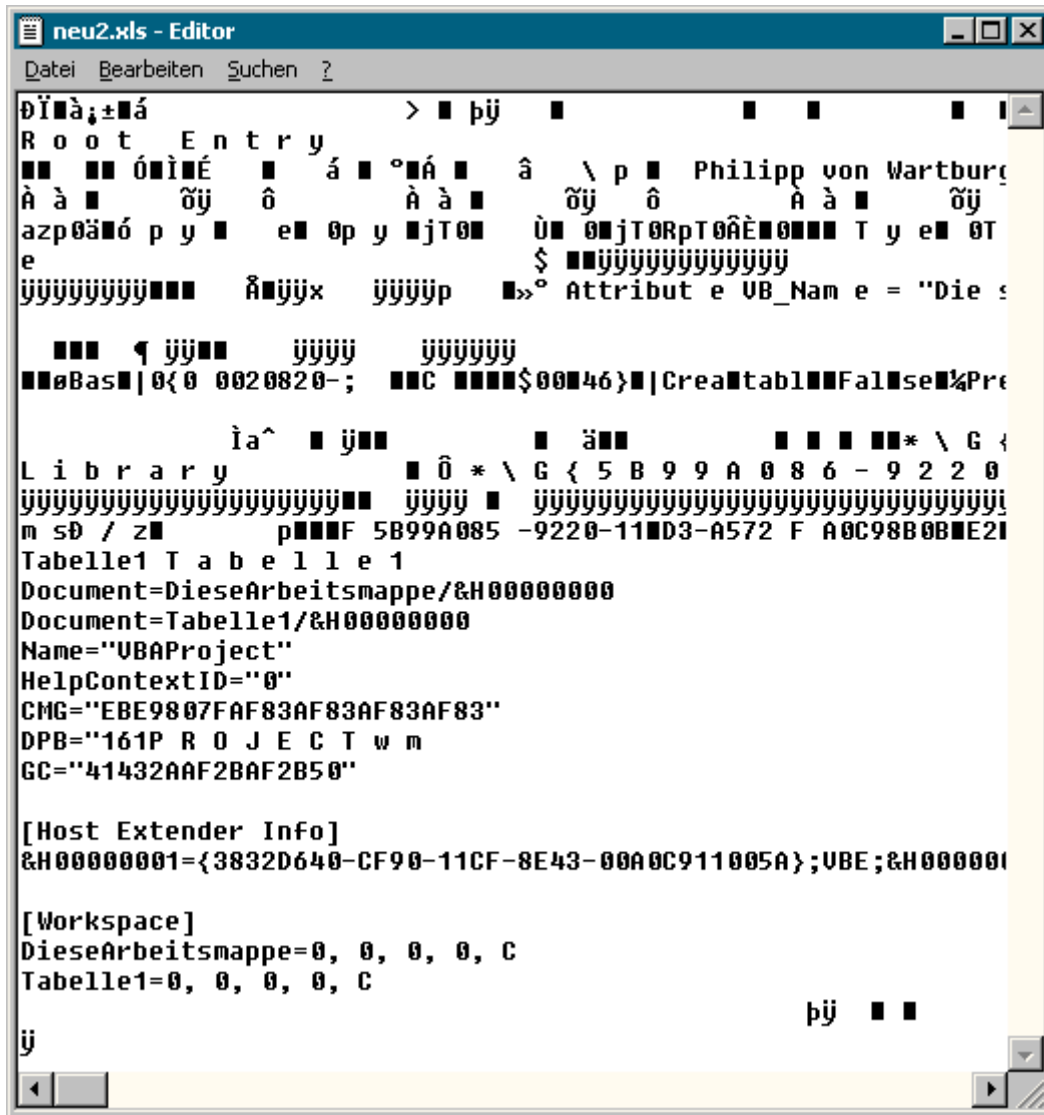


Abbildung: Arbeitsmappe mit VBA-Projektstruktur (geöffnet mit dem Windows Editor)

Die Projektstruktur wurde generiert, indem in Excel mit der Tastenkombination *Alt+F11* der VBA-Editor geöffnet wurde, als die Arbeitsmappe offen war, und dann die Mappe gespeichert wurde.

In welchen Situationen und Fällen die Projektstruktur generiert wird, ist im Abschnitt "[Generierung der VBA-Projektstruktur \(Symbol Table\)](#)" ausführlich erläutert. Wie man eine vorhandene Struktur wieder entfernen kann, erfahren Sie in Abschnitt "[Entfernen der VBA-Projektstruktur](#)".

Weitere Informationen
■ [Dateigröße von Arbeitsmappen reduzieren: VBA-Projektstruktur löschen](#)

Kommunikation zwischen Excel und VBA-Editor

Wenn der VBA-Editor geöffnet ist, müssen sämtliche verfügbaren Objekte und Funktionen ständig aktiviert und deaktiviert werden. Wird beispielsweise gerade eine Zelle bearbeitet (Bearbeiten-Modus), so sind alle Module des VBA-Projektes gesperrt. Wird die Zellbearbeitung abgeschlossen, werden die Objekte wieder freigegeben. Allgemein betrachtet sind die Objekte immer dann gesperrt, wenn sich Excel im sogenannten 'Modal State' befindet (z.B. während der Zellbearbeitung oder solange ein modales Dialogfenster geöffnet ist).

Die Sperrung und Entsperrung übernimmt der VBA Component Manager, eine interne Funktionskomponente des VBA-Editors. Jedes Objekt besitzt ein 'Freeze Bit', d.h. eine Markierung die besagt, ob es gerade "eingefroren" ist oder nicht. Der VBA Component Manager setzt die Freeze Bits aller betroffenen Objekte, sobald in den Modal State übergegangen wird. Wird der Modal State verlassen, werden die Freeze Bits wieder gelöscht. Aufgrund des Design des VBA-Editors müssen die Freeze Bits einzeln gesetzt bzw. gelöscht werden, indem auf sämtliche Objekte, ihre Sub-Objekte, die Sub-Objekte dieser Sub-Objekte usw. nacheinander zugegriffen wird. Dieser Vorgang benötigt natürlich Zeit, was bei sehr grossen Arbeitsmappen mit vielen Objekten (z.B. 500 Tabellenblättern) zu einer spürbaren Wartezeit führen kann.

▲
TOP

Speichernutzung von VBA-Programmen

In Bezug auf die Speichernutzung von VBA-Programmen existieren noch keine detaillierten Informationen.

VBA-Programmcode wird vom Prozess Excel.exe ausgeführt.

Der vom VBA-Programm benötigte Speicher ist somit Teil des vom Excel-Prozess belegten Speichers.

PSTORES.EXE	153	00	0:00:00	28 KB	1568 KB
mstask.exe	159	00	0:00:00	200 KB	940 KB
EXCEL.EXE	172	01	0:00:01	24656 KB	18832 KB
TASKMGR.EXE	193	00	0:00:01	1760 KB	480 KB
freditor.exe	343	00	0:00:37	3516 KB	5680 KB

Abbildung: Excel-Prozess im Task-Manager (Ausschnitt der Registerseite "Prozesse")

▲

Speicher von Variablen

Globale Variablen

Variablen mit *Dim* oder *Private* statt *Public* deklarieren.

Objektvariablen

Wichtig: Objektvariablen möglichst sofort freigeben!

```
Sub OpenCloseWorkbook()  
    Dim wkbWorkbook As Workbook  
    Set wkbWorkbook = Workbooks.Open("C:\Daten\EineMappe.xls")  
    wkbWorkbook.Close False  
    ' Sonstiger Programmcode...  
    Set wkbWorkbook = Nothing ' (1)  
End Sub
```

Mit der *Close*-Methode wird zwar die Arbeitsmappe EineMappe.xls geschlossen, da aber die Objektvariable *wkbWorkbook* noch eine Referenz auf die Mappe enthält, befindet sich die Mappe zum Teil immer noch im Speicher (zumindest das VBA-Projekt der Mappe). Erst durch Setzen der Objektvariable auf *Nothing* wird die Mappe vollständig aus dem Speicher entfernt. Dadurch kann die Speicherbelegung ohne weiteres um 1 MB oder mehr reduziert werden.

Datenfeldvariablen (Arrays)

Dynamische statt statische Arrays verwenden.

Datenfeldgrösse mit *ReDim* während der Laufzeit anpassen.

Inhalt eines Datenfeldes mit *Erase* leeren.



Speicher von XXX

Verweise

Projekt-Einstellungen (Projektname, Beschreibung, Kennwort, Hilfedatei, Kontext-ID etc.)

Zeichenfolgenspeicher (String Memory), Variablenspeicher, Stackspeicher



Speicher von Benutzerformularen (UserForms)

UserForms mit *Unload* entladen statt mit *Hide* ausblenden.

Unload UserForm

- statt -

UserForm.Hide



TOP

Tipps und Tricks

Nachfolgend finden Sie ein paar nützliche Tipps und Tricks zur optimalen Nutzung des Arbeitsspeichers von Microsoft Excel.

» Excel beenden und neu starten

Durch das Beenden und erneute Starten von Microsoft Excel werden Speicherbereiche freigegeben, die wegen eines Excel-Bugs irrtümlich noch belegt sind bzw. nicht korrekt freigegeben wurden. Sie sollten Excel immer dann neu starten, wenn

- die Fehlermeldung "Nicht genügend Arbeitsspeicher" aufgetreten ist



» Excel im abgesicherten Modus starten (Safe Mode)

Wenn Sie Microsoft Excel mit dem Befehlszeilenparameter *'/s'*, *'/safe'* bzw. *'/safemode'* (je nach Excelversion verschieden) ausführen, so wird Excel im abgesicherten Modus gestartet. Bei diesem Modus werden unter anderem keine im Startordner *XIStart* und dem zusätzlichen Startordner liegenden Dateien geöffnet, keine Add-Ins geladen und keine benutzerdefinierten bzw. angepassten Symbolleisten verwendet (da die *xlb*-Datei nicht geladen wird). Dadurch belegt die Anwendung ein Minimum an Arbeitsspeicher. Beispielsweise benötigt Excel 97 mit ein paar aktivierten Add-Ins zwischen 10 und 14 MB Speicher. Beim Starten im Safe Mode werden lediglich ca.

2.6 MB Speicher durch Excel belegt. Man braucht somit nur 1/4 bis 1/5 des üblichen Speicherbedarfs.



» Add-Ins deaktivieren

Deaktivieren Sie alle Add-Ins, die Sie nicht zwingend brauchen. Aktivieren Sie stattdessen ein Add-In erst wenn Sie es benötigen. Beachten Sie bitte, dass nach der Deaktivierung von Add-Ins Excel neu gestartet werden sollte, damit der zuvor belegte Speicherplatz effektiv freigegeben wird.



» Undo-Speicher von Excel leeren

Leeren Sie den Undo-Speicher aller geöffneten Arbeitsmappen. Wie dies gemacht wird, lesen Sie unter [Undo-Speicher leeren](#).



» Makroarbeitsmappe Personl.xls optimieren

Die persönliche Makroarbeitsmappe Personl.xls wird bei jedem Start von Microsoft Excel geöffnet. Überarbeiten Sie diese Arbeitsmappe, damit sie möglichst wenig Speicher beansprucht.



» xlb-Datei defragmentieren

In der xlb-Datei speichert Excel alle Informationen über benutzerdefinierte Symbolleisten und Steuerelemente sowie angepasste integrierte Symbolleisten. Mit der Zeit kann die Datei grösser werden, da infolge eines Excel-Bugs immer mehr nicht mehr benötigte und korrupte Daten in der Datei liegen bleiben. Durch das Defragmentieren der xlb-Datei werden alle überflüssigen Informationen aus der Datei entfernt, was zu einer erheblich kleineren Datei führt. Da beim Excelstart weniger Daten geladen werden, wird weniger Speicherplatz beansprucht.



» Arbeitsmappen in separaten Excel-Instanzen öffnen

Microsoft Excel besitzt ein so genanntes Workbook Memory, das für die Verarbeitung/Bearbeitung von Daten benötigt wird. Alle in der gleichen Excel-Instanz geöffneten Arbeitsmappen benutzen das Workbook Memory gemeinsam. Vereinfacht ausgedrückt bedeutet dies: Je mehr Mappen geöffnet sind, desto weniger Workbook Memory steht pro Arbeitsmappe zur Verfügung. Da jede Excel-Instanz ein eigenes Workbook Memory besitzt, können Sie - insbesondere bei grösseren Arbeitsmappen - Excel mehrmals starten und pro Excel-Instanz nur eine einzige Arbeitsmappe öffnen.



» Arbeitsmappen in einer anderen Reihenfolge öffnen

Falls mehrere grössere Arbeitsmappen gleichzeitig offen sein müssen und bereits beim Öffnen einer Mappe die Fehlermeldung "Nicht genügend Arbeitsspeicher" erscheint, so können Sie versuchen, die Arbeitsmappen in einer anderen Reihenfolge zu öffnen. Angenommen Sie möchten die drei Arbeitsmappen Buchhaltung.xls, Bilanz.xls und Statistik.xls gleichzeitig bearbeiten. Sie öffnen zuerst Buchhaltung.xls und dann Bilanz.xls. Anschliessend öffnen Sie Statistik.xls, wobei jedoch der obige Fehler auftritt. Es ist möglich, dass die Fehlermeldung nicht erscheint, wenn Sie eine andere Reihenfolge anwenden. Öffnen Sie zum Beispiel zuerst die Mappe Statistik.xls, dann Buchhaltung.xls und zum Schluss Bilanz.xls.



» **Berechnen-Modus auf "Manuell" stellen**

Wenn Sie den Berechnen-Modus vor dem Öffnen einer Arbeitsmappe auf "Manuell" stellen, wird die zu öffnende Mappe nicht neu berechnet.



» **VBA-Editor geschlossen lassen**

Vermeiden Sie es, den VBA-Editor nach dem Excelstart zu öffnen. Lassen Sie ihn so lange geschlossen wie Sie ihn nicht zwingend benötigen. Sobald der VBA-Editor einmal geöffnet ist, verbleibt er im Arbeitsspeicher. Der vom VBA-Editor belegte Speicher wird erst freigegeben, wenn die Excel-Sitzung beendet wird. Weitere Informationen dazu finden Sie im Kapitel [Speicherbeanspruchung des VBA-Editors](#).



» **VBA-Projekt säubern**



» **Codefenster im VBA-Editor schliessen**

Eine überaus einfache Möglichkeit zum Sparen von Arbeitsspeicher stellt das Schliessen von Codefenstern im VBA-Editor dar.

```
Sub CloseCodeWindows ()
    Dim i As Integer
    With Application.VBE
        For i = .Windows.Count To 1 Step -1
            If .Windows(i).Type = 0 Then
                .Windows(i).Close
            End If
        Next i
    End With
End Sub
```



» **Verweise im VBA-Projekt der Makroarbeitsmappe Personl.xls entfernen**

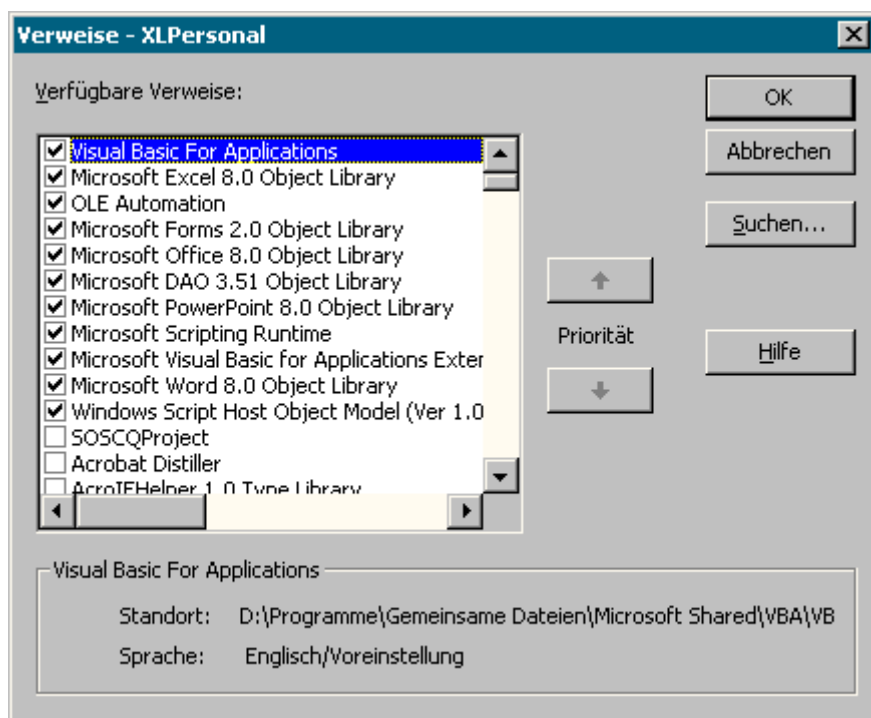


Abbildung: Verweise im VBA-Projekt der Mappe Personl.xls



» Benutzerformulare entladen statt ausblenden

Wenn Sie ein Benutzerformular mit *Hide* (z.B. UserForm1.Hide) schliessen, wird es lediglich ausgeblendet aber nicht aus dem Speicher gelöscht. Verwenden Sie besser *Unload* (z.B. Unload UserForm1), damit das Benutzerformular entladen und aus dem Speicher entfernt wird.



» Variablen und Konstanten mit geeigneten Datentypen deklarieren

Deklarieren Sie grundsätzlich sämtliche Variablen und Konstanten mit dem jeweils geeigneten Datentyp. Vermeiden Sie den Datentyp Variant. Damit keine Deklaration vergessen geht, sollten Sie die Option "Variablendeklaration erforderlich", zu finden im Optionen-Dialogfenster des VBA-Editors, aktivieren beziehungsweise immer aktiviert lassen.



» Variablen mit umfangreichen Inhalten nach Gebrauch leeren

Wenn Sie zum Beispiel im Programm eine Variable vom Typ String zum Speichern einer längeren Zeichenfolge einsetzen, dann sollten Sie die Variable leeren, so bald ihr Inhalt nicht mehr benötigt wird.

```
Sub GetFileContents()  
    Dim strFileContents As String  
    Open "C:\EineDatei.dat" For Binary Access Read As #1  
        strFileContents = Input(LOF(1), #1)  
    Close #1  
    ' Sonstiger Programmcode...  
    strFileContents = "" ' (1)  
End Sub
```



» Dynamische statt statische Arrays benutzen

Abgesehen davon, dass dynamische Datenfelder gewöhnlich weniger Speicher benötigen als statische, sollte man ganz grundsätzlich die Vorteile von dynamischen Datenfeldern nutzen.

Redimensionieren mit ReDim



» Objektvariablen unmittelbar nach Gebrauch zurücksetzen

Objektvariablen sollten zurückgesetzt werden, wenn sie nicht mehr benötigt werden. Das Zurücksetzen erfolgt durch das Zuweisen von *Nothing*.



» Codeausführung explizit beenden

Durch das explizite Beenden eines VBA-Programmes werden alle globalen und öffentlichen Variablen zurückgesetzt. Das Beenden kann erfolgen durch:

- Drücken der Beenden-Schaltfläche im VBA-Editor
- Ausführen der End-Anweisung im Direktfenster
- Ein- und Ausschalten des Entwurfsmodus (im VBA-Editor oder in Excel)
- Öffnen einer Arbeitsmappe
- Hinzufügen einer neuen Arbeitsmappe
- Schliessen einer Arbeitsmappe



TOP

Hilfsprogramme und Tools

Es gibt eine Vielzahl wirklich guter Tools, die bei der Analyse und Optimierung der Speichernutzung von Microsoft Excel äusserst nützlich sind. Zum Teil sind diese Tools bereits im Standard-Lieferumfang von Windows enthalten.

Diese Programme werden nachfolgend kurz vorgestellt:

- » [Code Cleaner](#)
- » [vbClear](#)
- » [XLB-Defragmentor](#)
- » [FreeMem](#)
- » [FastExcel](#)
- » [Windows Task-Manager](#)
- » [Windows Systemmonitor](#)
- » [Windows Systeminfo](#)
- » [Windows NT-Diagnose](#)

Nützliche Tools zum Vermeiden und Beheben von Problemen mit Limitationen finden Sie hier:



Code Cleaner

Der Code Cleaner ist ein sehr gutes Tool zum Aufräumen von VBA-Projekten. Dieses Excel Add-In löscht nicht den aktuellen VBA-Programmcode aus einer Arbeitsmappe sondern entfernt alte, früher einmal gelöschte Prozeduren und Module aus der Arbeitsmappen-Datei. Bei grösseren VBA-Projekten, die laufend erweitert und geändert werden, kann das Entfernen von bereits gelöschtem Programmcode zu einer erheblich kleineren Exceldatei führen. Eine Reduktion von 200 KB und mehr ist durchaus realistisch.

Das Tool Code Cleaner wurde durch Rob Bovey von den Application Professionals entwickelt. Die Webseite finden Sie hier: www.appspro.com

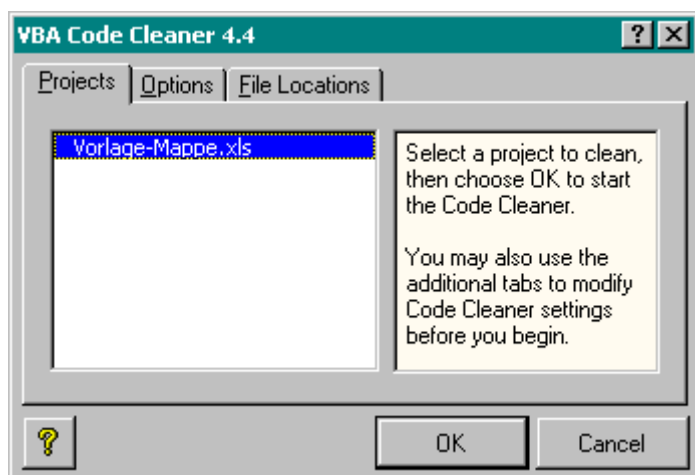


Abbildung: Das Excel Add-In 'Code Cleaner'

Der Code Cleaner ist Freeware und kann auf der Downloadseite heruntergeladen werden:

 [Hier gehts zur Downloadseite](#)



vbClear

Meiner Meinung nach ist vbClear (auch vbCleaner genannt) das beste zur Zeit erhältliche Freeware Tool zum Aufräumen von VBA-Projekten. Entfernt wie das Utility "Code Cleaner" sämtlichen alten, überflüssigen Programmcode aus der Excel-Arbeitsmappe, was zu einer kleineren Exceldatei führt. vbClear reduziert die Dateigrösse minim besser als der "Code Cleaner". Dafür bietet der "Code Cleaner" viele Optionen, die vbClear nicht besitzt.

vbClear wurde von Hans W. Herber entwickelt. Seine Internetseite finden Sie hier: www.herber.de

Das Add-In vbClear ist Freeware und kann auf der Downloadseite heruntergeladen werden:

 [Hier gehts zur Downloadseite](#)



XLB-Defragmentor

Der XLB-Defragmentor ist ein einzigartiges Tool zum Defragmentieren der xlb-Datei von Microsoft Excel. In der xlb-Datei speichert Excel alle Informationen über benutzerdefinierte Symbolleisten und Steuerelemente. Mit der Zeit kann die Datei grösser werden, da immer mehr nicht mehr benötigte und korrupte Daten in der Datei enthalten sind. Der XLB-Defragmentor analysiert die xlb-Datei, erstellt einen ausführlichen Report und entfernt alle fehlerhaften und überflüssigen Daten. Dadurch wird die xlb-Datei kleiner und ist bedeutend weniger fehleranfällig.

Der XLB-Defragmentor wurde von Philipp von Wartburg entwickelt. Detaillierte Produkt-Informationen finden Sie auf der offiziellen [Produkt-Webseite des XLB-Defragmentors](#).



Abbildung: Der XLB-Defragmentor zum Defragmentieren der xlb-Datei

Der XLB-Defragmentor ist Freeware und kann auf der Downloadseite heruntergeladen werden:

 [Hier gehts zur Downloadseite](#)



FreeMem

Mit dem Freeware Tool "FreeMem" kann man den Verlauf des verfügbaren Arbeitsspeichers ausgeben. Sie können natürlich auch ein anderes Tool einsetzen. Im Internet wird eine ganze Reihe vergleichbarer Programme angeboten (z.B. auf www.freeware.de oder www.jumbo.com).

Im Gegensatz zum Windows Task-Manager zeigt FreeMem nicht den Verlauf der benutzten Speichers sondern des verfügbaren Speichers, also genau die umgekehrte Darstellung. Zum Messen bzw. Kontrollieren der Speicherwerte eignet sich diese Darstellungsart besser (persönliche Meinung des Autors). FreeMem verfügt zudem über eine äusserst nützliche Funktion, die insbesondere beim Erforschen des Excelspeichers eingesetzt wird. Sie können den Arbeitsspeicher defragmentieren, indem der Speicherinhalt auf die Festplatte geschrieben und wieder eingelesen wird. Dadurch werden einerseits im Speicher infolge eines Programmabsturzes liegengebliebene Daten gelöscht und andererseits fragmentierte Speicherblöcke eliminiert. Als Resultat wird der verfügbare Arbeitsspeicher grösser (zum Teil um mehrere MB).

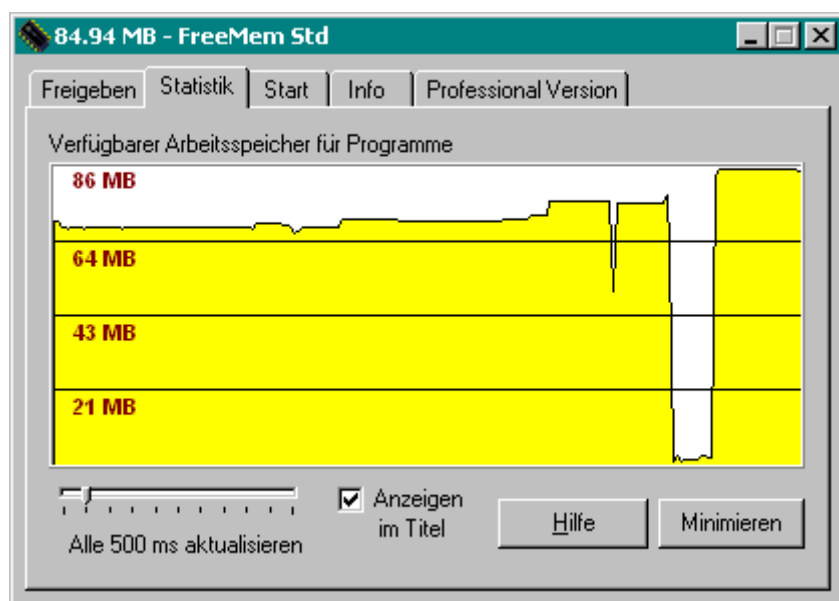


Abbildung: Das Utility 'FreeMem'

Die Webseite mit Produkt-Informationen und einer Download-Möglichkeit finden Sie hier: www.meikel.com/de/products/freemem



FastExcel

FastExcel ist ein professionelles und meiner Meinung nach das beste erhältliche Tool zum Optimieren von Arbeitsmappen. Informationen über dieses Produkt finden Sie auf der Webseite des Herstellers Decision Models: www.decisionmodels.com/fastexcel.htm

Eine Demo finden Sie auf der Downloadseite des Herstellers: www.decisionmodels.com/downloads.htm



Windows Task-Manager

Der Task-Manager ist ein in Windows integriertes Programm und somit standardmässig auf jedem Windows-Computer vorhanden.

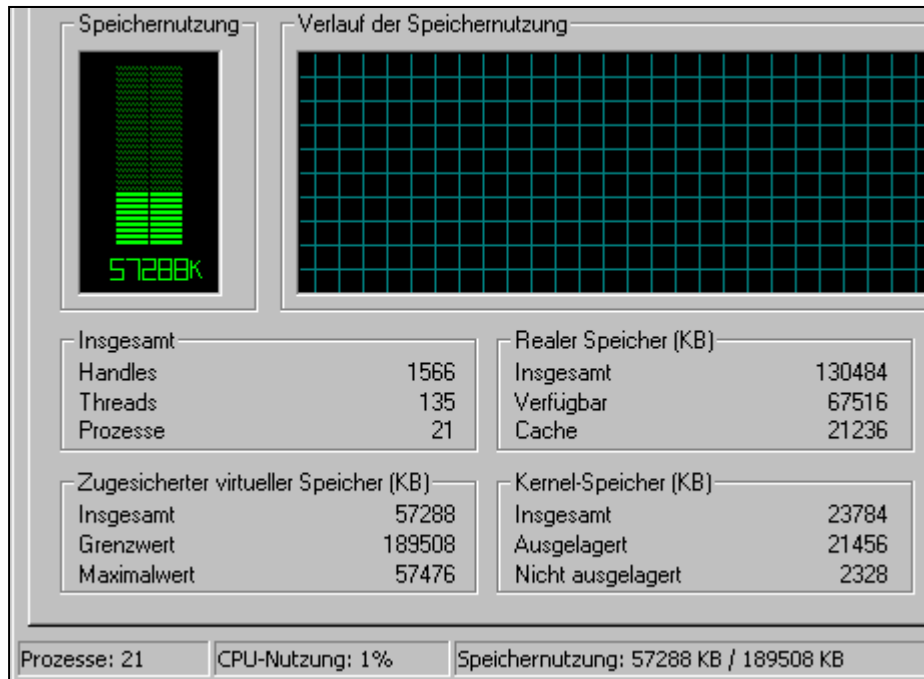


Abbildung: Der Task-Manager von Windows (Ausschnitt der Registerseite "Systemleistung")

Windows Systemmonitor

Der Systemmonitor ist ein Verwaltungsprogramm, das standardmässig auf einem Windows-Rechner zur Verfügung steht. Die Programmdatei des Systemmonitors heisst PerfMon.exe.

Arbeitsseiten: Speichernutzung (im Task-Manager)
 Auslagerungsdateiseiten: Virtueller Speicher (im Task-Manager)

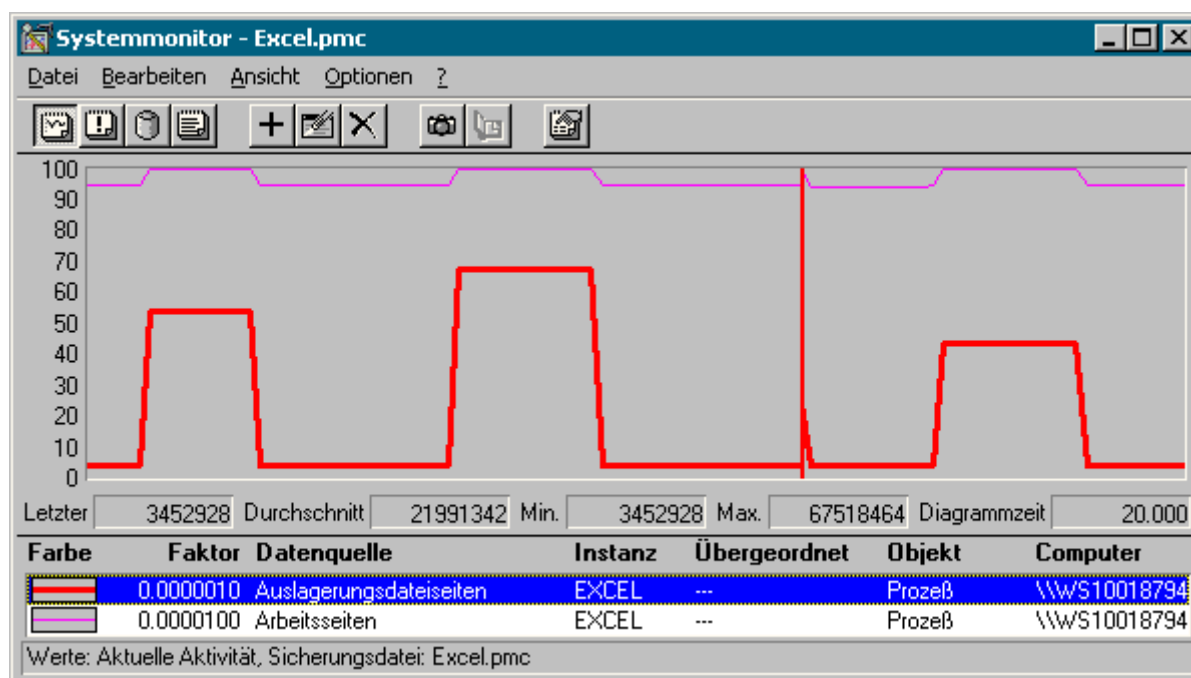


Abbildung: Der Systemmonitor von Windows

Windows Systeminfo

Das Utility "Windows Systeminfo" gehört zum Lieferumfang von Windows. Je nach Windows-Version sieht die Benutzeroberfläche ein bisschen anders aus.

Element	Wert
Betriebssystem:	Windows NT
Windows-Version:	4.0
Prozessor:	Pentium
Gesamter physischer Speicher:	130484 KB
Verfügbare physischer Speicher:	68624 KB*
Benutzer-Speicher:	90% frei
GDI-Speicher:	90% frei
Auslagerungsdatei:	189508 KB
Auslagerungsdatei-Nutzung:	34%
Auslagerungsdatei-Einstellungen:	Dynamisch
Verfügbare Speicher auf Laufwerk C:	17696 KB
Verfügbare Speicher auf Laufwerk D:	54033 KB
Windows-Ordner:	D:\WINNT
TEMP-Ordner:	D:\TEMP

* Siehe "Speicher" im Hilfeindex

Abbildung: Das Utility 'Microsoft Systeminfo'

Windows NT-Diagnose

Die Windows NT-Diagnose ist ein Standard-Utility von Windows NT (und evtl. auch neueren Windows-Versionen wie Windows 2000 und Windows XP). In Windows NT 4 können Sie es über *Start/Programme/Verwaltung* öffnen.

The screenshot shows the 'Windows NT-Diagnose' window for user 'WS10018794'. It features a menu bar with 'Datei' and a help icon. Below the menu are four tabs: 'Dienste', 'Ressourcen', 'Umgebung', and 'Netzwerk'. Under 'Ressourcen', there are sub-tabs for 'Version', 'System', 'Anzeige', 'Laufwerke', and 'Speicher'. The 'Speicher' sub-tab is active, displaying various memory statistics:

- Handles:** 1'755
- Threads:** 159
- Prozesse:** 23
- Realer Speicher (KB):**
 - Insgesamt: 130'484
 - Verfügbar: 73'940
 - Datei-Cache: 18'720
- Zugesicherter virt. Speicher (KB):**
 - Insgesamt: 77'912
 - Grenzwert: 271'412
 - Maximalwert: 271'248
- Kernel-Speicher (KB):**
 - Insgesamt: 25'700
 - Ausgelagert: 23'188
 - Nicht ausgelagert: 2'512
- Auslagerungsdateispeicher (KB):**
 - Insgesamt: 153'600
 - Insgesamt: 24'732
 - Maximalverwendung: 153'596

A table at the bottom shows details for the swap file:

Auslagerungs...	Insgesamt [...]	Verwendet (K...	Maximalverwen...
D:\pagefile.sys	153'600	24'732	153'596

Buttons at the bottom include 'Eigenschaften', 'Aktualisieren', 'Drucken', and 'OK'.

Abbildung: Die Windows NT-Diagnose



TOP

Weitere Informationsquellen

Links zum Thema "Speicher"

EXCEL: "Slow Response and Memory Problems"

<http://www.mvps.org/dmccritchie/excel/slowresp.htm>

» Ein hervorragender Artikel über Speicher- und Ressourcen-Probleme. Mit Anleitungen zur Performance-Optimierung und eine grösseren Liste mit Links zu weiteren Artikeln.

Decision Models

<http://www.decisionmodels.com/index.htm>

» Diese Webseite ist ein absolutes Muss für alle, die mehr zum Thema Speicher, Berechnen und Optimieren wissen möchten. Decision Models ist der Hersteller des Utilities 'Fast Excel', ein professionelles Werkzeug zum Optimieren von Excel-Arbeitsmappen.

Excel CrashesGPFs

<http://www.talkaboutsoftware.com/group/microsoft.public.excel.crashesgpf/>

» Eine Newsgroup von Microsoft ausschliesslich für Probleme durch Excel-Abstürze.

Umdhtools.exe: How to Use Umdh.exe to Find Memory Leaks

<http://support.microsoft.com/?scid=kb;en-us;268343>

WIN95/WIN98: How Windows 95/98 Manages Virtual Memory

<http://support.microsoft.com/?scid=kb;en-us;128327>

Your Right to Know: Finding Leaks and Bottlenecks with a Windows NT PerfMon COM Object

<http://msdn.microsoft.com/library/en-us/dnperfmo/html/perfmon.asp>

» Ein äusserst interessanter englischer Fachartikel von Microsoft Corporation zum Thema "Aufspüren von Speicherlecks in einer Anwendung anhand des Performance Monitors".

Umdhtools.exe: How to Use Umdh.exe to Find Memory Leaks

<http://support.microsoft.com/?scid=kb;en-us;268343>



Links zum Thema "Limitationen"

Eine umfassende Liste mit Links zu Artikeln über Limitationen finden Sie auf der Seite [Links](#).



Links zum Thema "Bugs und Probleme"

XL97: "Out of Virtual Memory" Opening Many Workbooks with ActiveX or OCX Controls

<http://support.microsoft.com/?scid=kb;en-us;238570>

XL97: "Not Enough Memory" Error Using Linking Formulas

<http://support.microsoft.com/?scid=kb;en-us;167079>

XL97: Deleting a Cell Changes the UsedRange Property

<http://support.microsoft.com/?scid=kb;en-us;232094>

XL97/XL2000/XL2002: Memory Leak Occurs When You Query an Open Excel Worksheet Using ADO

<http://support.microsoft.com/?scid=kb;en-us;319998>

XL97/XL2000: Memory Leak When Excel Calculates User-Defined Functions

<http://support.microsoft.com/?scid=kb;en-us;265023>

OFFICE: PageSetup Object Causes Memory Leak and Hangs System

<http://support.microsoft.com/?scid=kb;en-us;192869>

OFFICE97: Speicherprobleme bei Verwenden von Hyperlinks zwischen Programmen

<http://support.microsoft.com/?scid=kb;de;157763>

OFFICE97: Memory Problems When Hyperlinking Between Programs

<http://support.microsoft.com/?scid=kb;en-us;157763>

XL2000: Excel Considers Cell Outside Data Area As Last Cell in the Worksheet

<http://support.microsoft.com/?scid=kb;en-us;230998> (Dead Link!)

XL: How to reset the last cell in Excel (ex "XL2000: Resetting the Last Cell Fixes Memory and Printing Problems")

<http://support.microsoft.com/?scid=kb;en-us;244435>

OFFICE2000/OFFICEXP: A GDI Resource Leak Occurs When a Visual Basic for Applications UserForm Is Displayed

<http://support.microsoft.com/?scid=kb;en-us;321242>

XL2002: Anwendungsfehler in Modul "Excel.exe" bei Offset 0001d05f

<http://support.microsoft.com/default.aspx?scid=kb;de;321312>

XL2002: Application Error in Module Excel.exe at Offset 0001d05f

<http://support.microsoft.com/default.aspx?scid=kb;en-us;321312>

▲
TOP

Haben Sie Fragen, Anregungen oder einen Fehler entdeckt?
[Kontakt zum Autor](#)

Zuletzt aktualisiert am 11.12.2005 / 23:00 Uhr
© 2002-2005 by Philipp von Wartburg, CH-8916 Jonen
Alle Rechte vorbehalten