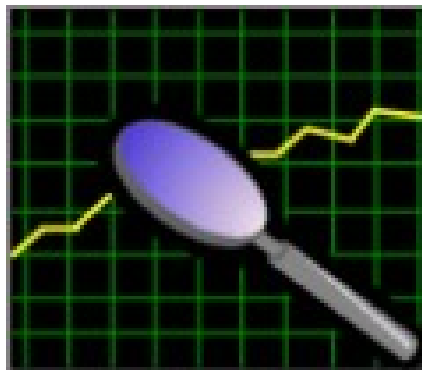


FindLeak – Handbuch

FindLeak Version 5.0
Stand Februar 2008



www.findleak.de

cjt Systemsoftware AG

Inhalt:

1. Lizenzvereinbarung.....	3
2. Informationsquellen rund um FindLeak.....	5
3. Systemvoraussetzungen für FindLeak 5.0.....	6
4. Installation von FindLeak.....	6
5. Willkommen bei FindLeak.....	6
6. Funktionsweise von FindLeak.....	7
Neuerungen in FindLeak 5.0.....	7
7. Aufbau von FindLeak.....	11
8. Beschreibung der einzelnen Menüpunkte.....	14
9. Definitionen	20
10. Workshop: FindLeak kennen lernen.....	21
11. Arbeiten mit FindLeak.....	29
12. Unterschiede zwischen Demo- und Vollversion.....	32
13. Registrierung / Freischaltung.....	32
14. Probleme und Lösungen.....	34
15. Fragen und Antworten.....	36

1. Lizenzvereinbarung

Mit der Installation der Software haben Sie die FindLeak- Lizenzvereinbarung anerkannt.

Im Folgenden finden Sie noch einmal den vollständigen Text dieser Vereinbarung:

WICHTIGER HINWEIS:

Lesen Sie die folgenden Bedingungen sorgfältig durch, bevor Sie die Software verwenden. Durch die Verwendung der Software erklären Sie sich mit den Bestimmungen dieser Lizenzvereinbarung einverstanden. Die im Folgenden beschriebene Software ist urheberrechtlich geschützt. Dies gilt auch für die im Folgenden definierten Lizenzschlüssel. Sowohl die Software als auch die Lizenzschlüssel dürfen nur im Rahmen der in diesem Dokument definierten Nutzungsbedingungen verwendet werden. Sollten Sie diesen Nutzungsbedingungen nicht zustimmen, dürfen sie weder die Software noch die Lizenzschlüssel verwenden.

LIZENZVEREINBARUNG

Die Software beinhaltet das Programm "FindLeak", die dazugehörenden Beispielprogramme, die dazu gehörenden Informationsdateien sowie jegliche Art der Dokumentation, die sich auf das Programm bezieht. Ausgenommen hiervon sind die Lizenzschlüssel. Der Lizenzschlüssel wird verwendet, um die Vollversion von FindLeak frei zu schalten. Mit der Demoversion von FindLeak lässt sich lediglich das mitgelieferte C++- Beispielprojekt bearbeiten.

NUTZUNGSBEDINGUNGEN

Dies ist eine rechtliche Vereinbarung zwischen Ihnen, dem Benutzer der Software und der cjt Systemsoftware AG. Mit der Installation und Verwendung der Software erkennen Sie diese Nutzungsbedingungen an. Sollten Sie diesen Bedingungen nicht zustimmen, ist es Ihnen nicht gestattet, die Software zu installieren oder zu verwenden.

1. Urheberrecht

Die Software ist urheberrechtlich geschütztes Eigentum der cjt Systemsoftware AG. Die Software darf nicht verändert, dekompiert, disassembliert, entschlüsselt, extrahiert oder irgendwie anderweitig verändert werden. Das Programm oder Teile davon dürfen nicht an Dritte vermietet, verkauft, weiter lizenziert oder sonst in irgendeiner Form ohne ausdrückliche Genehmigung der cjt Systemsoftware AG weiter vermarktet werden.

2. Lizenzschlüssel

Eine kommerzielle Lizenz von FindLeak kann nur auf EINEM Computer installiert und genutzt werden, da sie an die Hardware des Rechners gebunden ist.

Lizenzschlüssel dürfen nur von der Person verwendet, gespeichert oder kopiert werden, die diese Lizenz erworben hat und dies ausschließlich im Rahmen der in diesem Dokument definierten Nutzungsbedingungen. Es ist keiner Person oder Organisation gestattet, ohne schriftliche Genehmigung der cjt Systemsoftware AG eine Speicherung der Lizenzschlüssel zu einem anderen Zweck als der in dieser Lizenzvereinbarung definierten Nutzung erlaubt.

3. Copyright

Es ist Ihnen nicht erlaubt, Markennamen, Herstellerangaben oder Copyrighthinweise in der Software zu verändern oder zu entfernen.

Haftung

Die Software wurde mit Sorgfalt erstellt. Die cjt Systemsoftware AG übernimmt keine Gewährleistung dafür, dass die Software für einen bestimmten, vom Anwender angestrebten Zweck, geeignet ist. Bei Software muss nach Stand von Wissenschaft und Technik gegebenenfalls mit Fehlern gerechnet werden. Es können unter bestimmten Systemvoraussetzungen Unzulänglichkeiten nicht hundertprozentig ausgeschlossen werden. Für untypische, nicht vorhersehbare Schäden, insbesondere wegen des Auftretens von Computerviren, übernimmt die cjt Systemsoftware AG keine Haftung. Die cjt Systemsoftware AG übernimmt keine Gewährleistung über die Funktionsfähigkeit der Software auf dem Rechner des Anwenders.

Es wird jegliche Haftung ausgeschlossen, die aus der Nutzung der Software, deren Dateien und daraus resultierenden Schäden, Folgeschäden, entgangenem Gewinn, Betriebsunterbrechung, Verlust von Informationen oder irgendeinem anderen Vermögensschaden, für den Anwender oder für einen Dritten entstehen. Es wird jegliche Haftung ausgeschlossen für alle Schäden, gleich welcher Art, die durch Rückrufaktionen des Anwenders oder eines Dritten, der mit dem Anwender und der Software mittelbar oder unmittelbar im Zusammenhang steht, entstehen. Ansprüche Dritter, oder Ansprüche Dritter, die durch den Anwender getragen und vom Anwender von dem Autor eingefordert werden, sind in jedem Falle ausgeschlossen.

Die Lizenzbestimmungen können von Zeit zu Zeit angepasst werden.

Der Anwender der Software hat sich selbstständig über die aktuellen Lizenzvereinbarungen zu informieren. Sollte eine der Vereinbarungen unwirksam sein, so wird davon die Wirksamkeit der übrigen Vereinbarungen nicht berührt. Anstelle der unwirksamen Vereinbarungen treten wirksame Vereinbarungen, die dem Sinn der Absicht, der unwirksamen Vereinbarungen, zugunsten des Autors am nächsten kommen. Der Vertrag für die Überlassung von Lizenzen unterliegt dem anwendbaren Recht der Bundesrepublik Deutschland.

cjt Systemsoftware AG

Am See 14a

D 76297 Stutensee

Deutschland

Telefon: + 49 (7244) 73 41 11

Fax: + 49 (7244) 73 41 20

Die Lizenz ist Benutzer- und Hardware-gebunden und muss durch Registrierung aktiviert werden. Dies bedeutet, dass der Lizenzschlüssel, den Sie von uns erhalten, durch ein technisches Verfahren an die Festplatte des Rechners, auf dem Sie die Demoversion installiert haben und an den Benutzernamen gebunden ist. Sollten Sie zu einem späteren Zeitpunkt einen anderen Rechner bzw. eine andere Festplatte einsetzen, müssen Sie einen neuen Lizenzschlüssel beantragen.

Als Kunde von FindLeak informieren wir Sie über Programmaktualisierungen.

2. Informationsquellen rund um FindLeak

Informationen zu FindLeak erhalten Sie

im Internet unter www.findleak.de

per E-Mail unter support@findleak.de (bei technischen Fragen) oder

per E-Mail unter sales@findleak.de (bei Fragen zur Lizenzierung) oder

telefonisch unter +49 (7244) 73 41 11

FindLeak-Lizenzen können Sie entweder per Post bei

cjt Systemsoftware AG

Am See 14a

D 76297 Stutensee

Germany

oder per Fax unter folgender Nummer bestellen.

+ 49 (7244) 73 41 20

Verwenden Sie hierzu bitte jeweils den in der FindLeak- Demoversion enthaltenen Programmpunkt „Registrieren“, den Sie im Menü „Hilfe“ finden.

3. Systemvoraussetzungen für FindLeak 5.0

FindLeak kann unter Windows XP (mit SP2) und Windows Vista (32Bit) eingesetzt werden. Aufgrund des internen Aufbaus von FindLeak, kann dieses Tool nicht gleichzeitig mit anderen Tools, die zum Aufspüren von Memory Leaks dienen, eingesetzt werden.

Memory Leaks können bei allen Programmen angezeigt werden. Haltepunkte können allerdings nur für Programme angezeigt werden, die unter Microsoft Visual Studio (2002, 2003, 2005, 6.0 oder .Net) entwickelt wurden.

4. Installation von FindLeak

um die Installation zu beginnen, starten Sie die FindLeak.exe im Setup-Ordner. Nach dem Akzeptieren der Lizenzvereinbarung und Auswahl eines Speicherortes, wird FindLeak installiert. Die Demo-Version kann im Anschluss sofort benutzt werden. Um die Vollversion nutzen zu können, müssen Sie zuerst eine Lizenz beantragen. Weiteres Vorgehen dazu finden Sie unter Punkt 13 "[Registrierung/Freischaltung](#)".

5. Willkommen bei FindLeak

FindLeak wird eingesetzt bei der Suche nach Memory Leaks und zum Finden von Fehlern bei der Freigabe von BSTRs und Safearrays. Das Tool wird bereits seit einiger Zeit erfolgreich in verschiedenen großen Projekten im Bereich der Systemsoftwareentwicklung eingesetzt. Hinter FindLeak steckt das Know-how mehrjähriger Erfahrung in der Entwicklung, Fehlersuche und im Debugging unter den verschiedenen Win32-Plattformen. Während FindLeak aktiv ist, erhöht es den Speicherbedarf des zu untersuchenden Prozesses und lässt ihn auch dann steigen, wenn keine Memory Leaks vorhanden sind. Eine Überprüfung mit dem Systemmonitor oder dem Taskmanager, ob der Speicher eines Prozesses wächst, ist daher nur dann sinnvoll, wenn FindLeak nicht aktiv ist. Werden von FindLeak keine Memory Leaks angezeigt, ist dies kein Garant dafür, dass der analysierte Prozess keine Memory Leaks verursacht, da es noch einige Entstehungsmöglichkeiten für Memory Leaks gibt, die von FindLeak nicht überwacht werden.

6. Funktionsweise von FindLeak

In jedem startenden Prozess wird eine sog. Applnit-DLL geladen, die zunächst überprüft, ob Findleak.exe aktiv ist. Ist FindLeak aktiv, werden per Default-Einstellung alle nach FindLeak gestarteten Prozesse von FindLeak überwacht. Für jeden Prozess wird dabei eine Spy-DLL geladen, die das Anlegen und Freigeben der in der folgenden Tabelle beschriebenen Ressourcen überwacht. Dieses Verhalten kann aber durch das Setzen von Optionen beeinflusst werden (wird an späterer Stelle beschrieben).

FindLeak überwacht folgende Ressourcen:

BSTR	Funktionen, die für die Allokierung und Freigabe von BSTR in der Oleaut32.dll benutzt werden
SafeArray	Funktionen, die für die Allokierung und Freigabe von Safearrays in der Oleaut32.dll benutzt werden
Memory	die malloc, calloc, realloc und free -Funktionen der MSVCRT.dll und MSVCRTD.dll sowie die GlobalAlloc und GlobalFree Funktionen der Kernel32.dll
Event	CreateEvent und CloseHandle Funktionen der Kernel32.dll
GDI	die Funktionen zur Erzeugung und Freigabe von GDI-Objekten (außer Bitmaps) der GDI32.dll
Heap	HeapAlloc, HeapFree

Jeder Prozess, den die Spy-DLL überwacht, wird in FindLeak angezeigt. Bei jedem Aufruf einer Funktion, die eine Ressource anlegt, wird ein Eintrag in FindLeak erzeugt. Wird die Ressource wieder freigegeben, wird der Eintrag in FindLeak wieder entfernt. Dadurch werden in FindLeak alle momentan benutzten Ressourcen eines Prozesses angezeigt.

Neuerungen in FindLeak 5.0

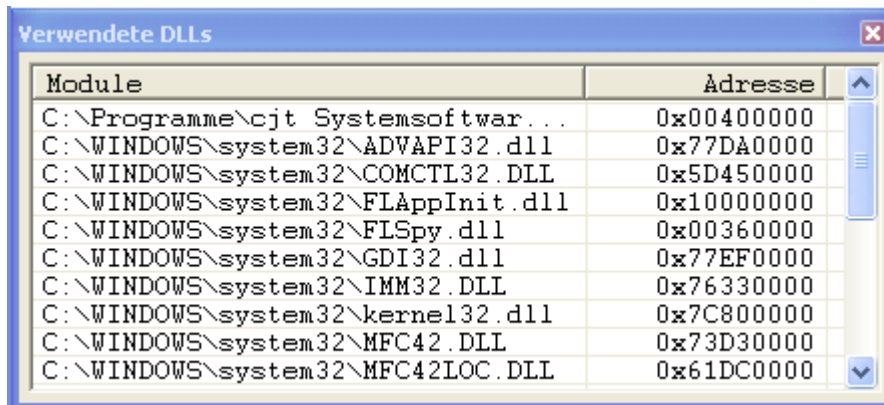
Ab Version 5.0 sind zahlreiche neue Features hinzugekommen:

Die Überwachung von Prozessen kann nun beendet und wieder gestartet werden

Die Überwachung von Prozessen kann in FindLeak über das Kontextmenü des jeweiligen Prozesses mit „Überwachung beenden“ beendet werden. Soll der Prozess dann wieder Überwacht werden, kann über das Kontextmenü „Überwachung fortsetzen“ die Überwachung wieder gestartet werden.

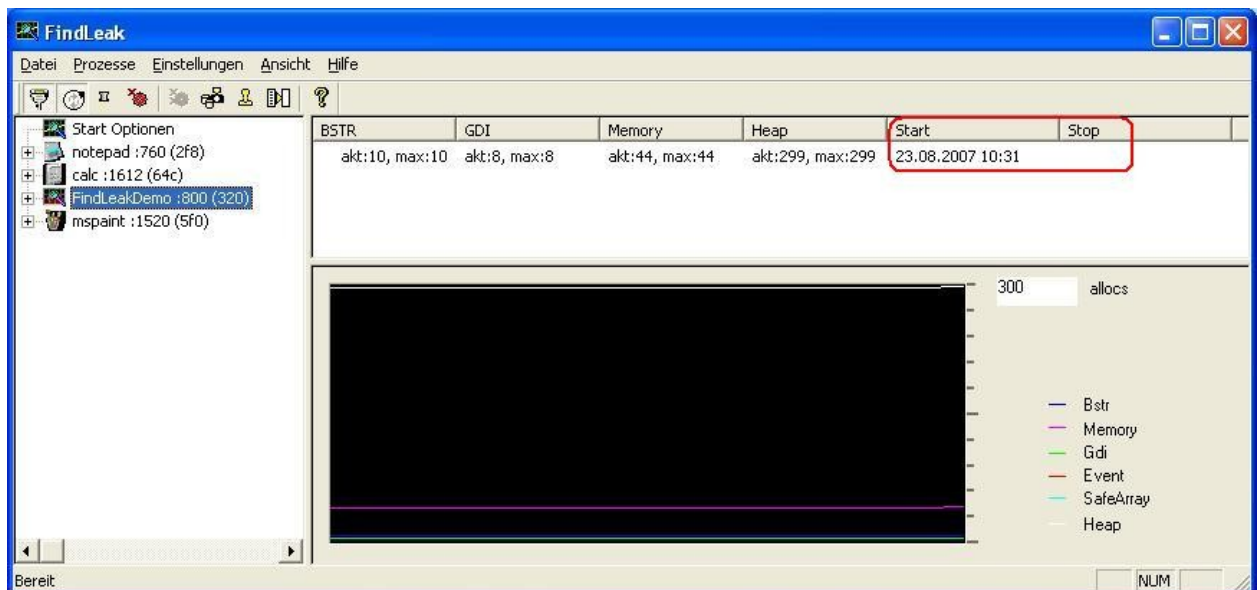
Benutzte DLLs des Prozesses können angezeigt werden

Über das Kontextmenü eines Prozesses wird der Unterpunkt „Verwendete DLLs“ aufgerufen. In dem Dialog werden dann alle verwendeten DLLs angezeigt.



Die Start- und die Stoppzeit des Prozesses wird mitprotokolliert

Für jeden Prozess wird die Startzeit mitprotokolliert. Sobald der Prozess beendet ist, wird zusätzlich noch die Stoppzeit eingetragen.



Zeitliche Markierungen werden unterstützt

Über den Menüpunkt „Einstellungen“ – „Zyklisch markieren“ kann über einen Dialog eingestellt werden, in welchem Abstand die Objekte markiert werden sollen (In der Abbildung alle 30 Sekunden).

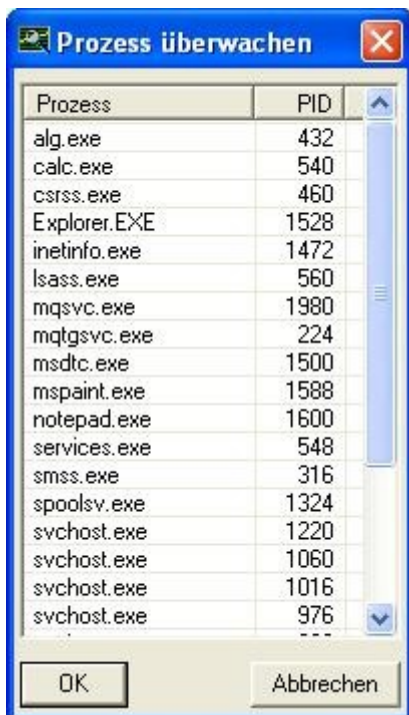


FindLeak kann nun auch beendet werden, wenn es noch Prozesse überwacht

Auch wenn Prozesse aktiv sind und diese überwacht werden, kann FindLeak über den Menüpunkt *Datei – Beenden* oder durch einfaches Schließen beendet werden.

Es ist möglich, auch bereits laufende Prozesse zu überwachen

Prozesse, die bereits aktiv sind, bevor FindLeak geöffnet ist, können über den Menüpunkt „Prozesse“ – „Laufenden Prozess überwachen“ angezeigt werden. Mit Doppelklick auf einen Prozess kann dieser dann in FindLeak (nur in der Registrierten Version) überwacht werden.



Hierbei gilt jedoch zu beachten, dass die Speicheranforderungen erst mit Beginn der Überwachung erfolgt. D.h. Allokierungen und Freigabebefehle die vor der Überwachung aufgetreten sind, können nicht angezeigt und beanstandet werden.

Verbesserte Anzeige der Callstacks

Es werden nicht mehr nur die Adressen des Stacks, sondern auch die Module angezeigt. Wenn eine PDB für den überwachten Prozess gefunden wird, werden auch zusätzlich die Funktion, die Datei und die Zeile angezeigt, in deren Stelle die Funktion im Quellcode steht.

Mit Doppelklick oder Rechtsklick auf eine solche Zeile wird der Quellcode in Visual Studio, bzw. wenn dieser nicht installiert ist – in Notepad, angezeigt.

Zusätzlich ist es möglich, per Drag&Drop die Adresse direkt in Visual Studio bzw. einen Editor zu ziehen.

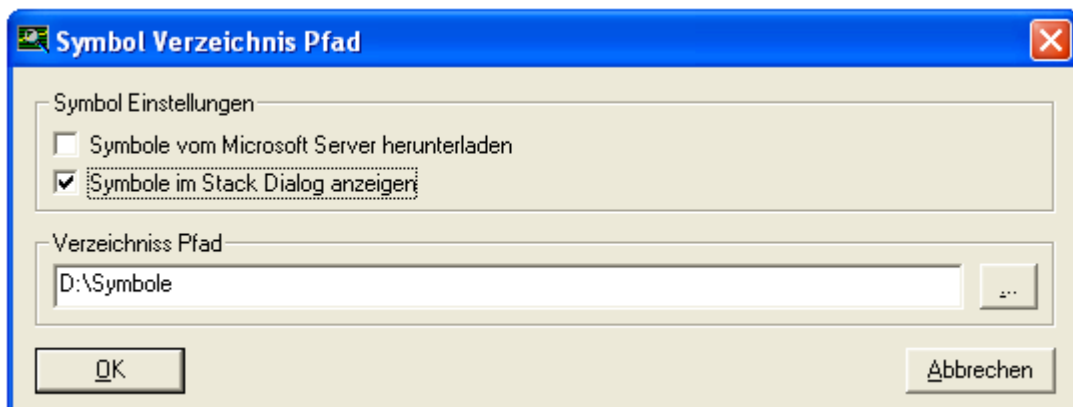


Adresse	Module	Funktion	Datei	Zeile
774CDE7B	ole32.dll	StringFromGUID2		
774D2A46	ole32.dll	CoInitialize		
004012B6	FindLeakDe...	CFindLeakDemoApp::InitATL	FindLeakDemo....	145
00401102	FindLeakDe...	CFindLeakDemoApp::InitInstance	FindLeakDemo....	46
73D3CF74	MFC42.DLL	Ordinal1576		
7C816D4F	kernel32.dll	RegisterWaitForInputIdle		

Der Pfad für Symbole kann angegeben werden.

Über den Menüpunkt Einstellungen – Symbolpfad, erscheint der Dialog „Symbol Verzeichnis Pfad“. Durch setzen des zweiten Häkchens (siehe Abbildung), werden in der CallStack-Anzeige zusätzlich die Funktion und (falls vorhanden) die Datei und die Zeile angegeben (siehe oben).

Durch das setzen des ersten Häkchens werden die Symbole direkt vom Microsoft Symbol-Server herunter geladen. Hier muss nur noch ein lokales Verzeichnis ausgewählt werden indem die Symbole gespeichert werden sollen.



7. Aufbau von FindLeak

Nach dem Starten von FindLeak sehen Sie links eine Baumdarstellung mit dem Eintrag „Startoptionen“. Rechts sehen Sie eine Liste, in der sich zunächst nur der Eintrag „<Other>“ befindet.



Wenn Sie den Eintrag „<Other>“ doppelklicken, öffnet sich das Fenster „Prozess Optionen“. Hier können Sie die Optionen einstellen, die beim Start eines neuen Prozesses gültig sein sollen. Durch zusätzliche Einträge können die Optionen für bestimmte Prozesse geändert werden. Alle Prozesse, die keinen eigenen Eintrag haben, werden über den Eintrag „<Other>“ initialisiert.

Alle neu gestarteten Prozesse werden in der obersten Ebene der Baumdarstellung mit ihrem Default-Icon angezeigt. Ist kein Default-Icon vorhanden, erhält ein aktiver Prozess das grüne  -Icon. Jeder Prozess im Baum besitzt Untereinträge, die die Ressourcentypen Heap, Event, Memory, GDI darstellen. Selektieren Sie im Baum einen Prozess, werden in der Listenansicht alle benutzten Ressourcentypen, die aktuelle Anzahl der Allokierungen sowie die maximale Anzahl der Allokierungen angezeigt (Übersichtsansicht).

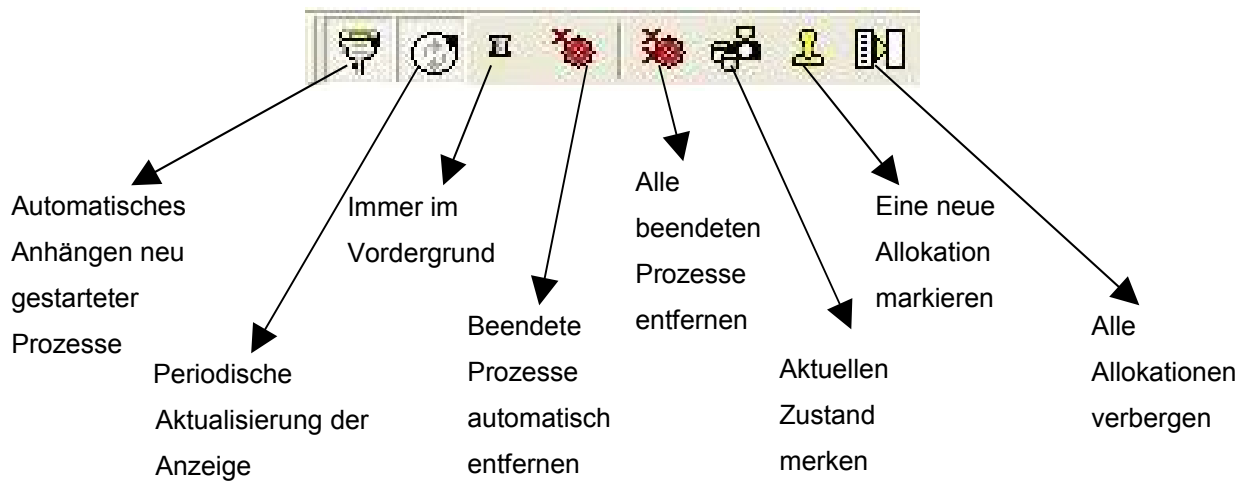
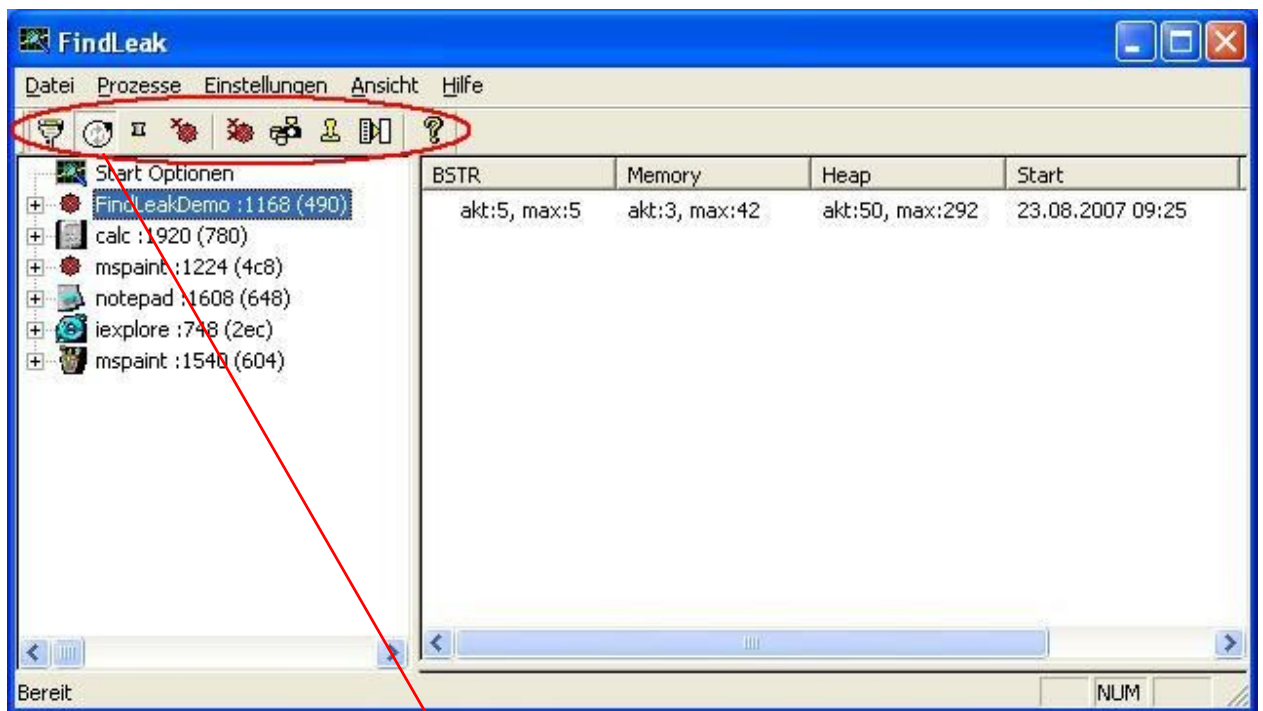
Zusätzlich wird unter der Listenansicht ein Kurvenfenster angezeigt, in dem der zeitliche Verlauf der Anzahl aller Allokierungen angezeigt wird. Selektieren Sie im Baum einen Ressourcentyp, werden in der Listenansicht alle aktuell angelegten Ressourcen dieses Typs angezeigt (Detailansicht).

1. Spalte:	akt/alle	Die Ressourcen werden in der Reihenfolge, in der sie angelegt werden, durchnummeriert. In der Titelzeile werden die aktuelle Anzahl von Allokierungen sowie die gesamte Anzahl aller Allokierungen dieses Typs angezeigt.
2. Spalte	Haltepunkt	Hier wird eine von FindLeak vergebene Nummer angezeigt, die der Stelle im Sourcecode entspricht, an der der Speicher angelegt wurde.
3. Spalte	Adresse	Der Pointer bzw. der Handle der Ressource
4. Spalte	Größe	Der Inhalt dieser Spalte ist typabhängig:

		Bei „Memory“, „Heap“, „BSTR“ und „SafeArray“ wird die Größe des Speichers angezeigt. Bei „Event“ wird der Name des Events angezeigt, falls es sich um ein „named event“ handelt. Bei „GDI“ sehen Sie den GDI-Typ.
5. Spalte	Stack	Hier wird der Callstack zum Zeitpunkt der Allokierung angezeigt, wenn Sie diese Option aktiviert haben.

Zusätzlich wird in einem Kurvenfenster der zeitliche Verlauf der Allokierungen angezeigt. Wird ein Prozess beendet und sind alle Ressourcen freigegeben worden, wird er aus der Baumansicht entfernt. Wurden nicht alle Ressourcen freigegeben, wird der Prozess als beendet markiert und bleibt in der Baumdarstellung erhalten. Die Farbe des Prozess-Icons ändert sich entsprechend in rot: . Dieses Verhalten kann durch den Anklicken der Schaltfläche *"Beendete Prozesse automatisch entfernen"* dahingehend geändert werden, dass auch beendete Prozesse, bei denen nicht alle Ressourcen freigegeben worden sind, automatisch beim Beenden aus dem Baum entfernt werden. Wenn Sie dagegen auf die Schaltfläche *"Alle beendeten Prozesse entfernen"* klicken, werden alle bis dahin beendeten Prozesse aus der Baumdarstellung entfernt. Der Unterschied zur zuvor beschriebenen Schaltfläche besteht darin, dass hier kein Automatismus für das Entfernen beendeter Prozesse sorgt.

Icon-Leiste



8. Beschreibung der einzelnen Menüpunkte

Menü „Prozesse“

Wenn zu einem Menüpunkt zusätzlich eine Schaltfläche existiert, die dieselbe Funktionalität anbietet, wird das entsprechende Icon bei der Beschreibung des Menüpunktes abgebildet.

Aktuellen Zustand merken

Von allen aktiven Prozessen wird die Anzahl der aktuell allokierten Objekte gespeichert.

Alle beendeten Prozesse entfernen

Alle beendeten Prozesse werden aus der Baumansicht entfernt.

Haltepunkte löschen

Alle aktuell gesetzten Haltepunkte werden wieder entfernt.

Alle Allokierungen verstecken

Alle bisher allokierten Objekte werden in der Detailansicht nicht mehr angezeigt.

Alle Allokierungen anzeigen

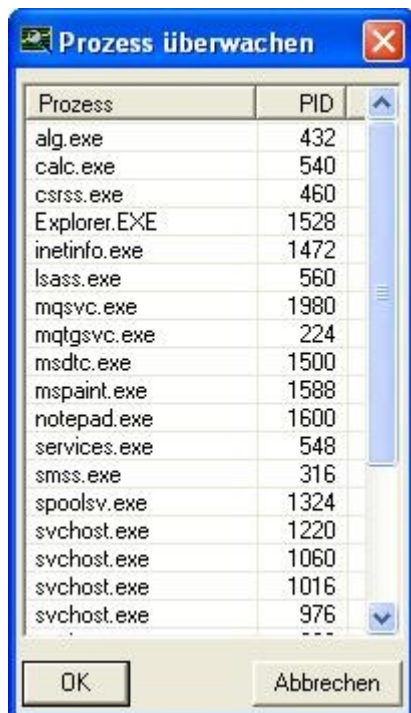
Es werden wieder alle aktuell allokierten Objekte angezeigt.

Neue Allokierungen markieren

Die Allokierungsnummer erhält eine zusätzliche Nummer, die bei jedem Aufruf dieses Menüpunktes um 1 erhöht wird.

Laufenden Prozess überwachen...

Hier werden alle Prozesse angezeigt, die ausgeführt werden. Mit Doppelklick auf einen Prozess kann dieser zu FindLeak hinzugefügt und überwacht werden.



Menü „Einstellungen“

Prozesse anhängen

Solange dieser Menüpunkt aktiviert ist, werden alle neu gestarteten Prozesse von FindLeak in die Überwachung aufgenommen.

Laufend aktualisieren

Wenn dieser Menüpunkt aktiviert ist, wird die Detailansicht (wenn sie angezeigt wird), zyklisch aktualisiert. Dies kann manchmal zu langen Laufzeiten führen. Vor allem dann, wenn in einem Prozess viel allokiert und wieder freigegeben wird. Ist dieser Menüpunkt nicht aktiviert, wird die Detailansicht nur beim Markieren eines Objektes im Baum aktualisiert, was die Verarbeitungsgeschwindigkeit erheblich erhöht.

Immer im Vordergrund

Wenn dieser Menüpunkt aktiviert ist, erscheint FindLeak immer im Vordergrund.

Beendete Prozesse automatisch entfernen

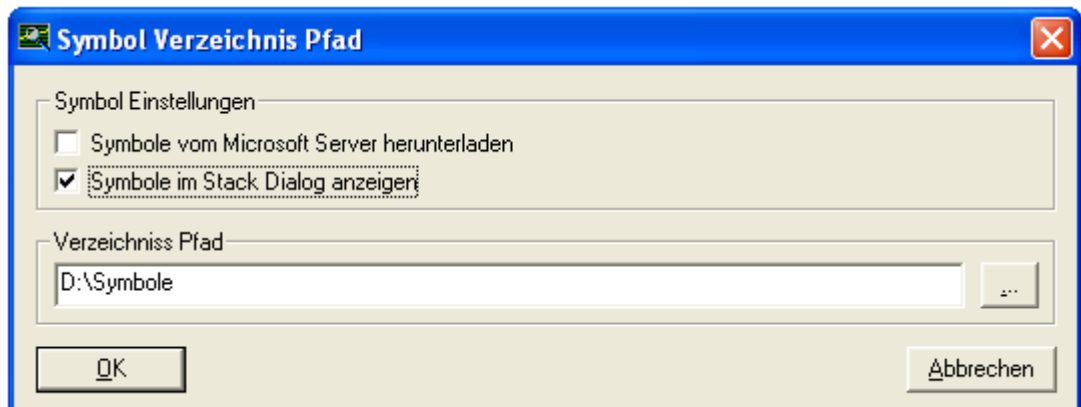
Ist dieser Menüpunkt aktiviert, werden alle Prozesse automatisch aus der Baumdarstellung entfernt, sobald sie beendet wurden. Ist der Menüpunkt nicht aktiviert, werden nur solche Prozesse entfernt, die alle Objekte freigeben.

Zyklisch markieren

Hier kann über einen Dialog eingestellt werden, in welchem Abstand die Objekte markiert werden sollen.

Symbolpfad eingeben

Bei der Auswahl dieses Menüpunktes erscheint ein Dialog, indem durch das setzen des ersten Häkchens, die Symbole direkt vom Microsoft Symbol-Server herunter geladen und beim setzen des zweiten Häkchens in der CallStack-Anzeige zusätzlich die Funktionen, die Datei und die Zeile angegeben werden.



Sprache auswählen

Hier kann zwischen der Sprache Deutsch und Englisch ausgewählt werden

Kontextmenü der Baumansicht (linke Seite des FindLeak- Fensters):

Dieses Kontextmenü erreichen Sie, wenn Sie einen der überwachten Prozesse in der Baumansicht per Mausklick markieren und dann die rechte Maustaste drücken.

Achten Sie in der Baumansicht auch auf die Farbe, in der die einzelnen Prozesse dargestellt werden: GRÜN  bedeutet, dass es sich um einen AKTIVEN Prozess handelt, der von FindLeak überwacht wird. ROT  dagegen weist darauf hin, dass es sich hier um einen Prozess handelt, der von FindLeak überwacht wurde, der jetzt aber nicht mehr aktiv ist.

Prozess entfernen

Der Prozess wird aus der Baumdarstellung entfernt, wenn er nicht mehr aktiv ist.

Optionen...

Das Fenster „Prozessüberwachung“ wird angezeigt. Hier können sie die Überwachungsoptionen für den markierten Prozess auswählen.

Debugger starten

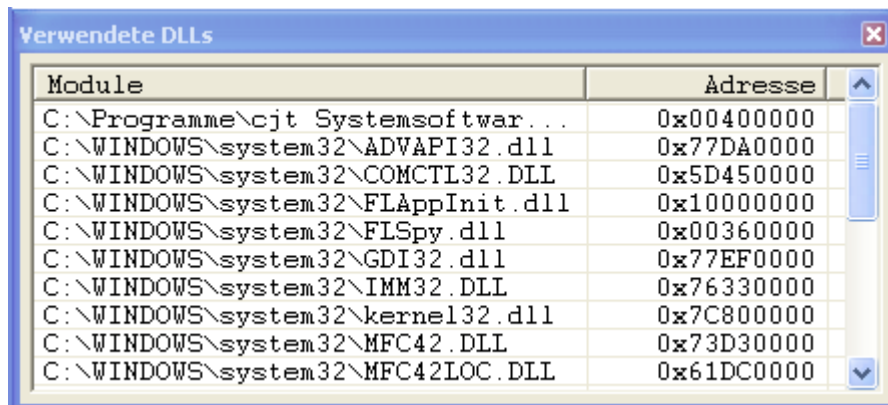
Der Default-Debugger Ihres Arbeitsplatzes wird für den selektierten Prozess gestartet, ohne dass der Debugger selbst in die FindLeak -Überwachung aufgenommen wird.

Aktuellen Zustand merken

Es wird die Anzahl der aktuell allokierten Objekte für diesen Prozess gespeichert.

Verwendete DLLs

In einem Dialog werden alle verwendeten DLLs mit der dazugehörigen Adresse angezeigt.



Module	Adresse
C:\Programme\cjt Systemsoftwar...	0x00400000
C:\WINDOWS\system32\ADVAPI32.dll	0x77DA0000
C:\WINDOWS\system32\COMCTL32.DLL	0x5D450000
C:\WINDOWS\system32\FLAppInit.dll	0x10000000
C:\WINDOWS\system32\FLSpy.dll	0x00360000
C:\WINDOWS\system32\GDI32.dll	0x77EF0000
C:\WINDOWS\system32\IMM32.DLL	0x76330000
C:\WINDOWS\system32\kernel32.dll	0x7C800000
C:\WINDOWS\system32\MFC42.DLL	0x73D30000
C:\WINDOWS\system32\MFC42LOC.DLL	0x61DC0000

Haltepunkte löschen

Alle aktuell gesetzten Haltepunkte werden wieder entfernt.

Alle Allokierungen verstecken

Alle bisher allokierten Objekte werden in der Listenansicht verborgen.

Allokierungen anzeigen

Es werden alle momentan allokierten Objekte angezeigt.

Neue Allokierungen markieren

Die Allokierungsnummer erhält eine zusätzliche Nummer, die bei jedem Aufruf dieses Menüpunktes um 1 erhöht wird.

Objekt Typ Item:

Haltepunkte löschen

Der aktuell gesetzte Haltepunkt wird wieder entfernt.

Alle Allokierungen verstecken

Alle bisher allokierten Objekte werden in der Detailansicht nicht mehr angezeigt.

Alle Allokierungen anzeigen

Es werden alle aktuell allokierten Objekte angezeigt.

Neue Allokierungen markieren

Die Allokierungsnummer erhält eine zusätzliche Nummer, die bei jedem Aufruf dieses Menüpunktes um 1 erhöht wird.

Kontextmenü des Listenansicht (rechte Seite des FindLeak-Fensters)

Startoptionen

Optionen hinzufügen ...

Startoptionen für einen speziellen Prozess hinzufügen. Die Einstellungen für einen neuen Eintrag werden erst nach einem erneuten Start von FindLeak aktiv.

Optionen entfernen ...

Startoptionen für einen Prozess wieder entfernen.

Optionen ändern ...

Ändern der Startoptionen.

Übersichtsansicht

Aktuellen Zustand merken

Es wird die Anzahl der aktuell für diesen Prozess allokierten Objekte gespeichert.

Detailansicht

Detailinformationen ...

Es wird der Inhalt des selektierten Objektes angezeigt. Wenn der Detail-Dialog geöffnet ist, kann der angezeigte Inhalt durch Doppelklick auf eines der anderen Objekte in der Liste gewechselt werden.

Stack anzeigen ...

Wenn im Fenster „Prozessüberwachung“ die Option „Stack anzeigen ...“ aktiviert ist, kann der Stack in einem Fenster angezeigt werden.

Anhalten bei Allokierung

Setzt einen Haltepunkt im zu untersuchenden Prozess, an dem das selektierte Objekt angelegt wurde. Wird der Code zum Anlegen dieses Objekts erneut durchlaufen, wird der Debugger mit einem Debug-break angehalten. Läuft der Prozess nicht in einem Debugger, wird die MessageBox "User Break" angezeigt.

Haltepunkte löschen

Der aktuell gesetzte Haltepunkt wird wieder entfernt. Die selektierten Objekte werden in der Detailansicht nicht mehr angezeigt.

Allokierung verstecken

Die markierte Allokierung wird ausgeblendet

Alle Allokierungen verstecken

Alle bisher allokierten Objekte werden in der Detailansicht nicht mehr angezeigt.

Alle Allokierungen anzeigen

Es werden wieder alle aktuell allokierten Objekte angezeigt.

Neue Allokierungen markieren

Die Allokierungsnummer erhält eine zusätzliche Nr., die bei jedem Aufruf dieses Menüpunktes um 1 erhöht wird.

Dialog Prozessüberwachung

	halten bei	unbek. Zeiger	2. Freigabe	Stack zeigen	ausschalten	permanent auf
BSTR:	0	☒	☒	☐	☐	☐
Safearray:	0	☒	☒	☐	☐	☐
Memory	0	☒	☐	☐	☐	☐
Event:	0	☒	☐	☐	☐	☐
GDI:	0	☒	☐	☐	☐	☐
Heap:	0	☒	☐	☐	☐	☐

Über die Optionen in der Prozessüberwachung stellen Sie das Verhalten von FindLeak ein. Die Optionen werden beim Starten eines Prozesses gelesen und können, nachdem der Prozess aktiv ist, über den Kontextmenüpunkt "Optionen ..." geändert werden. Ist kein spezieller Eintrag für einen Prozess vorhanden, werden die Einstellungen von "Startoptionen" <Other> verwendet. Für eine prozessspezifische Initialisierung muss ein neuer Eintrag angelegt werden. Der Prozess wird durch seinen Namen (ohne Pfad und Extension) gekennzeichnet.

Unbekannter Zeiger

Wenn Inkonsistenzen bei FindLeak auftreten, wird die MessageBox „FindLeak failed ...“ angezeigt. Wird diese MessageBox angezeigt, konnte das Deallokieren von einem oder mehreren Speicherbereichen nicht mitprotokolliert werden. Dadurch könne bereits freigegebene Speicherbereiche noch als Memory Leaks in FindLeak angezeigt werden.

Zweite Freigabe

Wenn versucht wird, bereits freigegebenen Speicher erneut freizugeben, wird eine entsprechende MessageBox angezeigt. Läuft der Prozess in einem Debugger, wird der Debugger durch ein Debugbreak angehalten. Diese Fehler müssen unbedingt sofort beseitigt

werden. Ist dies nicht möglich, kann diese Option in dem Prozess, in dem dieser Fehler auftritt, ausgeschaltet werden, wenn gerade ein anderer Prozess untersucht wird.

Stack zeigen

Zu jeder Allokierung wird der aktuelle Stack mitgespeichert, so dass diese Option verschlechtert die Performance verschlechtert und den Speicherbedarf von FindLeak erhöht. Ohne die Anzeige der Stackinformation können Memory Leaks auch mit Hilfe der Haltepunkte gefunden werden.

ausschalten

Das Mitprotokollieren der Allokierungen wird abgeschaltet.

Permanent aus

Die Allokierungen werden nie mitprotokolliert. Diese Einstellung kann allerdings nur durchgeführt werden, wenn der Prozess gerade nicht aktiv ist.

Mit dem Button „Start Option setzen“ werden die eingegebenen Daten gespeichert und beim nächsten Start von FindLeak geladen.

9. Definitionen

Memory Leaks

Memory Leaks oder Speicherlecks entstehen durch Fehler bei der Entwicklung einer Software. Hierbei werden Systemressourcen angefordert, die nicht wieder freigegeben werden, wenn sie nicht mehr benötigt werden. Das fehlende Freigeben von Ressourcen führt allerdings nur dann zu Problemen, wenn der Code, der zum Anlegen dieser Ressourcen dient, während der Laufzeit des Prozesses mehrfach durchlaufen wird.

Freigabefehler

Werden von einem Prozess BSTRs und Safearrays verwendet, die über die Oleaut32.dll angelegt werden, kann dies zu Problemen bei der Freigabe dieser Ressourcen führen. Die Freigabefunktionen liefern keine Fehlermeldung, wenn der Speicher bereits freigegeben ist. Dies kann z.B. zu folgenden Szenarien führen:

Fall A – Richtige Memory-Verwaltung

1. Anlegen des BSTR 0x00140000 im Thread A
2. Freigeben des BSTR 0x00140000 im Thread A
3. Erneutes Freigeben des BSTR 0x00140000 im Thread A (liefert keinen Fehler)

Fall B – Auslösen eines Freigabefehler

1. Anlegen des BSTR 0x00140000 im Thread A
2. Freigeben des BSTR 0x00140000 im Thread A
3. Anlegen des BSTR 0x00140000 im Thread B
4. Erneutes Freigeben des BSTR 0x00140000 im Thread A (Ab jetzt wird im Thread B Speicher benutzt, der bereits freigegeben wurde. Eine anderweitige Nutzung dieses Speicherbereichs kann zu Programmabstürzen führen)
5. Freigeben des BSTR 0x00140000 im Thread B (liefert keinen Fehler). Da der Fall B 3 wahrscheinlich relativ selten auftritt, führt dieses Verhalten zu sporadischen Programmabstürzen, die nur durch Code Reviews zu finden sind.

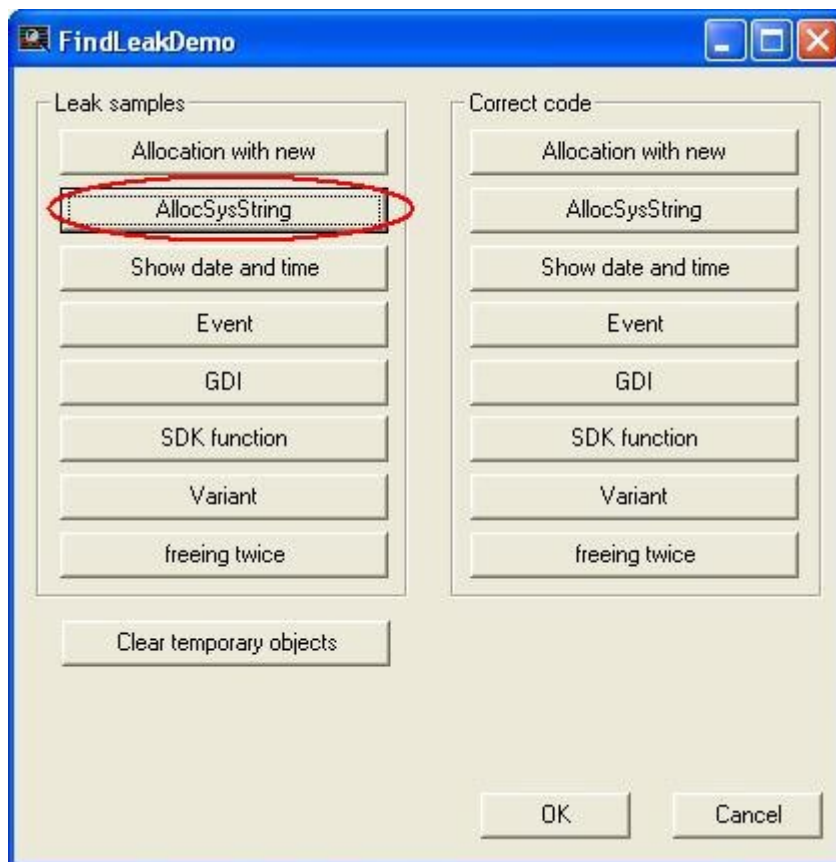
10. Workshop: FindLeak kennen lernen

Den folgenden Workshop können Sie sowohl mit der Demo- als auch mit der Vollversion von FindLeak durcharbeiten. Wir möchten Sie als Besitzer der Demoversion ausdrücklich dazu ermuntern, das beschriebene Beispiel gemeinsam mit uns nachzuvollziehen, damit Sie sich selbst einen Eindruck von den Vorteilen und Arbeitserleichterungen verschaffen können, die Ihnen der Einsatz von FindLeak bietet. Das Arbeiten mit FindLeak soll im Folgenden am Beispiel einer Speicherallokierung mit "AllocSysString" vorgestellt werden. Dieses Beispiel steht exemplarisch für all jene Fälle, in denen Speicher allokiert und nicht wieder freigegeben wird. Es stellt nur einen der vielen Anwendungsfälle dar, bei denen der Einsatz von FindLeak sinnvoll ist.

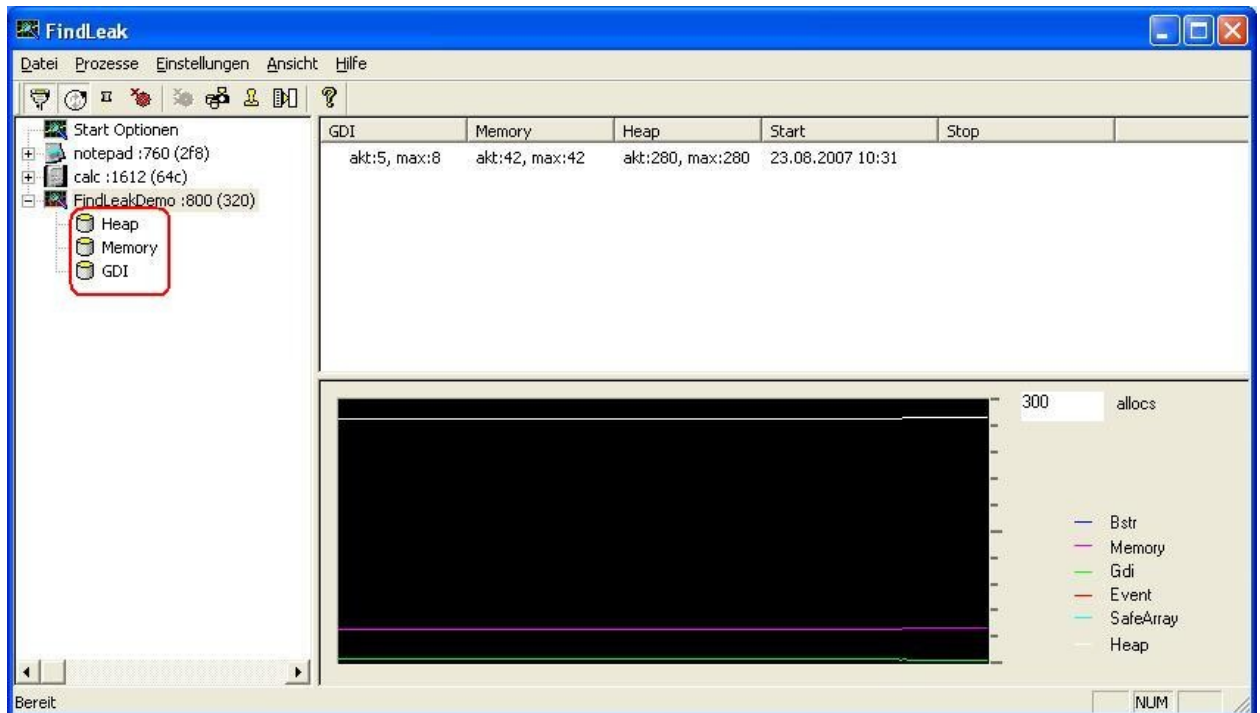
1. Starten Sie FindLeak und anschließend die zu überwachende Anwendung.

Als Beispiel verwenden wir hierzu die mitgelieferte Microsoft Visual C++-Beispielanwendung "FindLeakDemo".

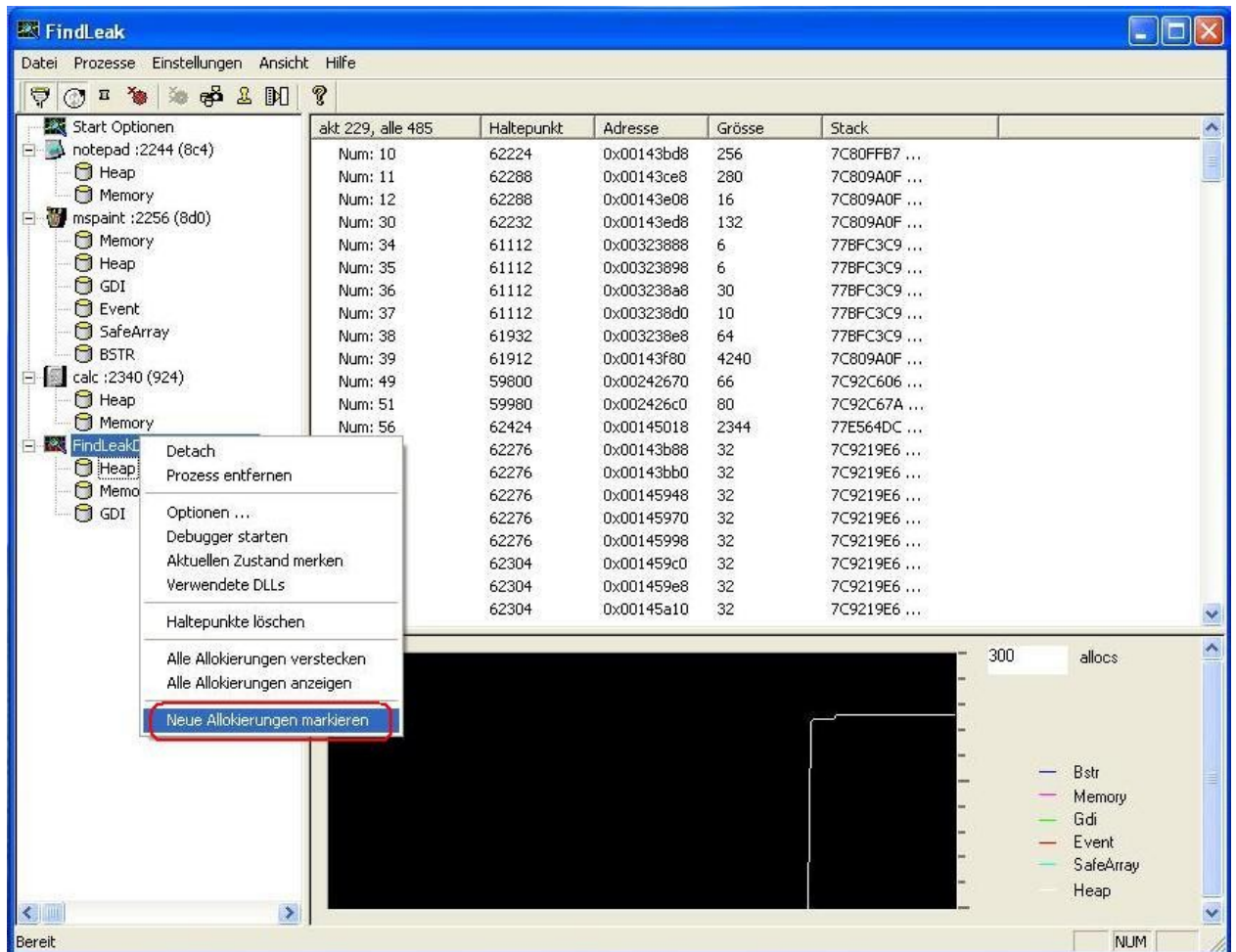
Bitte wählen Sie innerhalb der Kategorie "Leak samples", den Button "AllocSysString". In unserem Beispiel wird daraufhin Speicher belegt, der anschließend nicht wieder freigegeben wird. Unser Ziel ist es nun, dieses Ressourcenleck nachzuweisen und auf eine einfache Art und Weise zu lokalisieren.



Die gestartete Anwendung ist nun in FindLeak zu sehen. Nach dem Expandieren der Baumdarstellung sind auf der rechten Seite die momentan von der Anwendung belegten Speicherressourcen für Heap, Memory und GDI zu sehen.

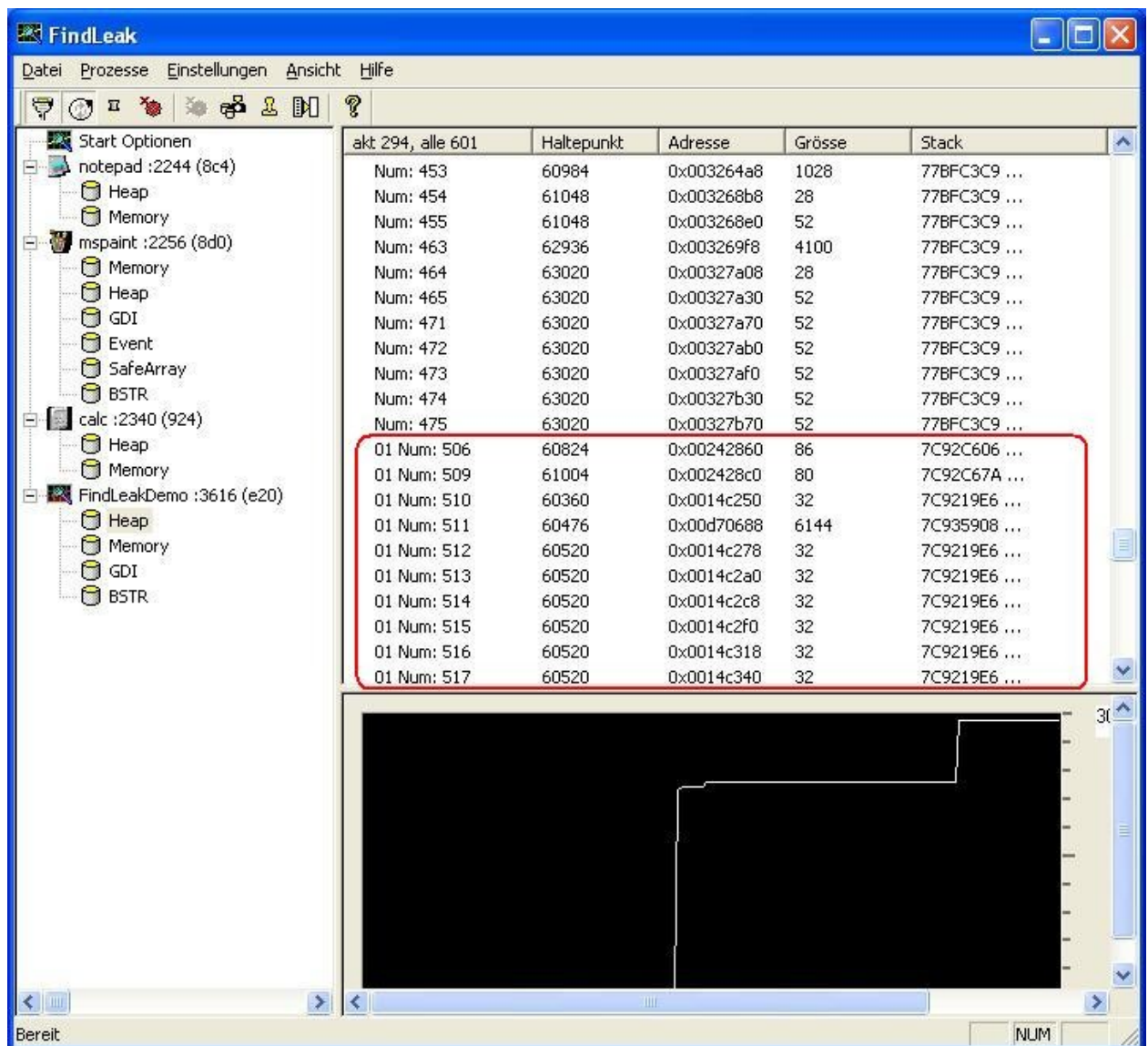


Damit neue Speicherallokierungen angezeigt werden können, markieren Sie in FindLeak Ihre Anwendung. Öffnen sie den Menüpunkt Prozesse und wählen Sie den Unterpunkt „Neue Allokierungen markieren“ aus.



Wir führen den fehlerhaften Code erneut aus, indem wir in unserem Beispielprojekt FindLeak-Demo wieder auf das Feld "AllocSysString" klicken.

Anschließend selektieren Sie bitte in der Baumdarstellung von FindLeak ein weiteres Mal den Eintrag "FindLeakDemo" und wählen hier "Heap" aus. In der Tabellenansicht auf der rechten Seite sind nun neue Einträge erschienen, die mit dem Präfix "01" vor "Num..." beginnen. Diese Einträge stehen für die in diesem Durchgang neu belegten Speicherplätze.



Um weitere Speicherallokierungen durch erneutes Ausführen des fehlerhaften Codes zu demonstrieren, markieren Sie Ihre Anwendung in FindLeak erneut mit "Neue Allokierung markieren" und klicken danach in der Demoanwendung nochmals auf "AllocSysString".

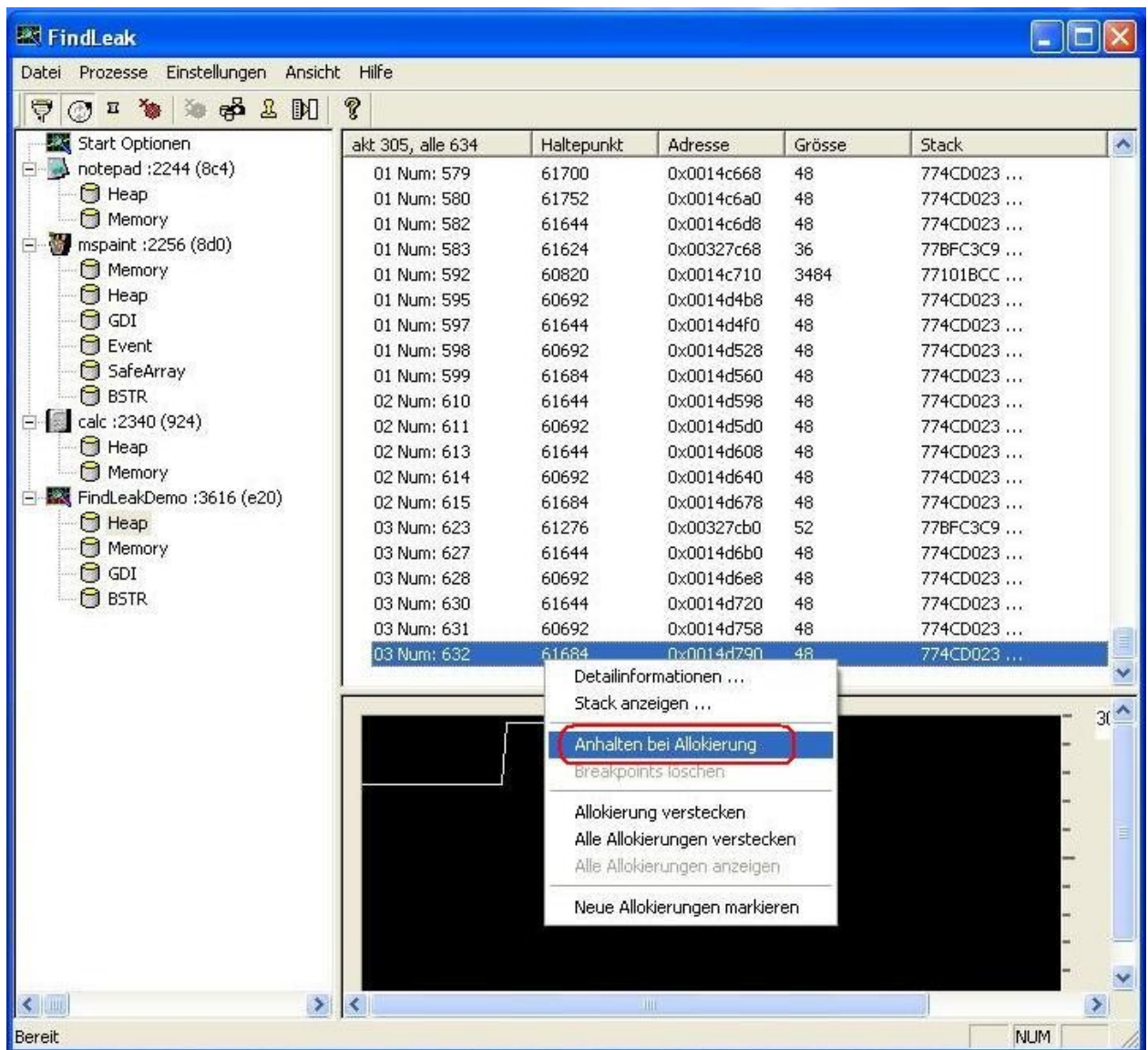
Es erscheinen weitere neue Einträge für Speicherbelegungen, die jetzt mit "02" vor "Num..." beginnen. Diese stehen für die beim zweiten Durchgang neu hinzugekommenen Allokierungen.

akt	Haltepunkt	Adresse	Grösse	Stack
01 Num: 561	60640	0x00d72f38	18	003F3CD4 ...
01 Num: 562	60640	0x00d72f58	128	003F3CD4 ...
01 Num: 571	61724	0x0014c538	228	774CD023 ...
01 Num: 572	61684	0x0014c628	16	774CD023 ...
01 Num: 573	61516	0x0014c640	32	7C9219E6 ...
01 Num: 578	61752	0x0014c500	48	774CD023 ...
01 Num: 579	61700	0x0014c668	48	774CD023 ...
01 Num: 580	61752	0x0014c6a0	48	774CD023 ...
01 Num: 582	61644	0x0014c6d8	48	774CD023 ...
01 Num: 583	61624	0x00327c68	36	77BFC3C9 ...
01 Num: 592	60820	0x0014c710	3484	77101BCC ...
01 Num: 595	60692	0x0014d4b8	48	774CD023 ...
01 Num: 597	61644	0x0014d4f0	48	774CD023 ...
01 Num: 598	60692	0x0014d528	48	774CD023 ...
01 Num: 599	61684	0x0014d560	48	774CD023 ...
02 Num: 610	61644	0x0014d598	48	774CD023 ...
02 Num: 611	60692	0x0014d5d0	48	774CD023 ...
02 Num: 613	61644	0x0014d608	48	774CD023 ...
02 Num: 614	60692	0x0014d640	48	774CD023 ...
02 Num: 615	61684	0x0014d678	48	774CD023 ...

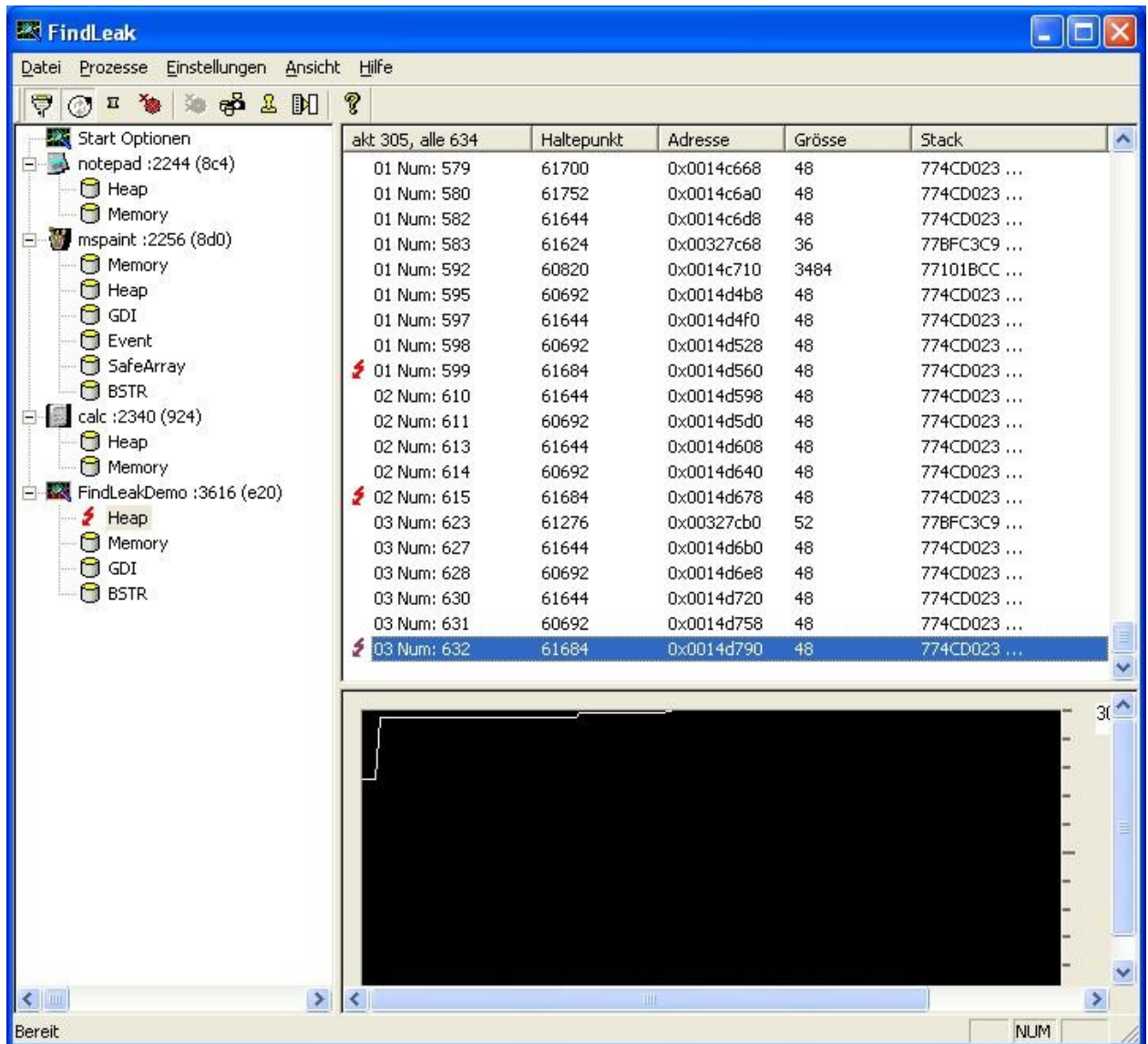
Wiederholen Sie diese Schritte noch einige Male. Sie sehen, dass bei jedem Durchgang eine bestimmte Menge Speicher allokiert und nicht wieder freigegeben wird. In all diesen Fällen wird in der Tabellenspalte "Haltepunkt" auf denselben Wert verwiesen.

Sie haben jetzt den Nachweis erbracht, dass diese Anwendung allokierten Speicher nicht vollständig zurückgibt und haben auch die Stelle im Programmcode gefunden, an der dieser Speicher nicht wieder zurückgegeben wird.

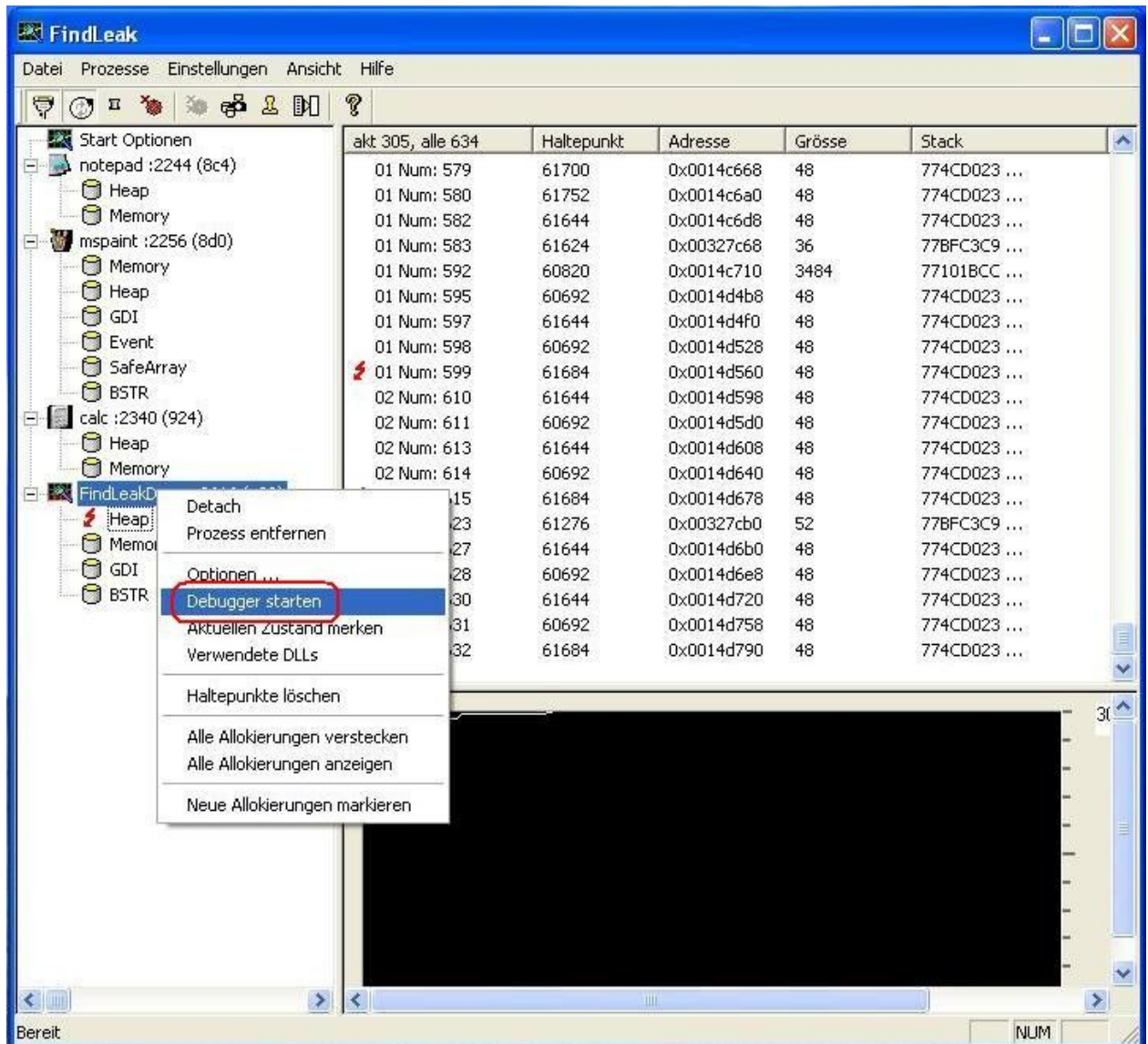
Um an dieser Stelle später zu debuggen, setzen Sie in FindLeak einen Haltepunkt. Hierzu markieren Sie einen der Einträge, die in der Spalte "Haltepunkt" auf denselben Wert verweisen. Im Kontextmenü wählen Sie nun "Anhalten bei Allokierung" aus.



Das Symbol vor "Heap" in der Baumansicht verwandelt sich daraufhin in einen roten Blitz. Vor allen Listeneinträgen (rechts), die auf dieselbe Speicherstelle verweisen, erscheint nun ebenfalls ein roter Blitz. Das Blitz-Symbol wird innerhalb von FindLeak verwendet, um einen Haltepunkt darzustellen.



Abschließend selektieren Sie in der Baumansicht Ihre Anwendung und klicken im Kontextmenü auf den Eintrag "Debugger starten".



Wenn Sie den kritischen Programmcode jetzt ein weiteres Mal ausführen, wird das Programm an der markierten Stelle angehalten und der Debugger Ihrer Entwicklungsumgebung, wird gestartet. Dort haben Sie dann die Möglichkeit, die gefundene Fehlerstelle näher zu analysieren.

Wir sind nun am Ende unseres kleinen Workshops angelangt und Sie haben gesehen, wie man mit Hilfe von FindLeak in kurzer Zeit sehr einfach ein Speicherleck lokalisieren kann.

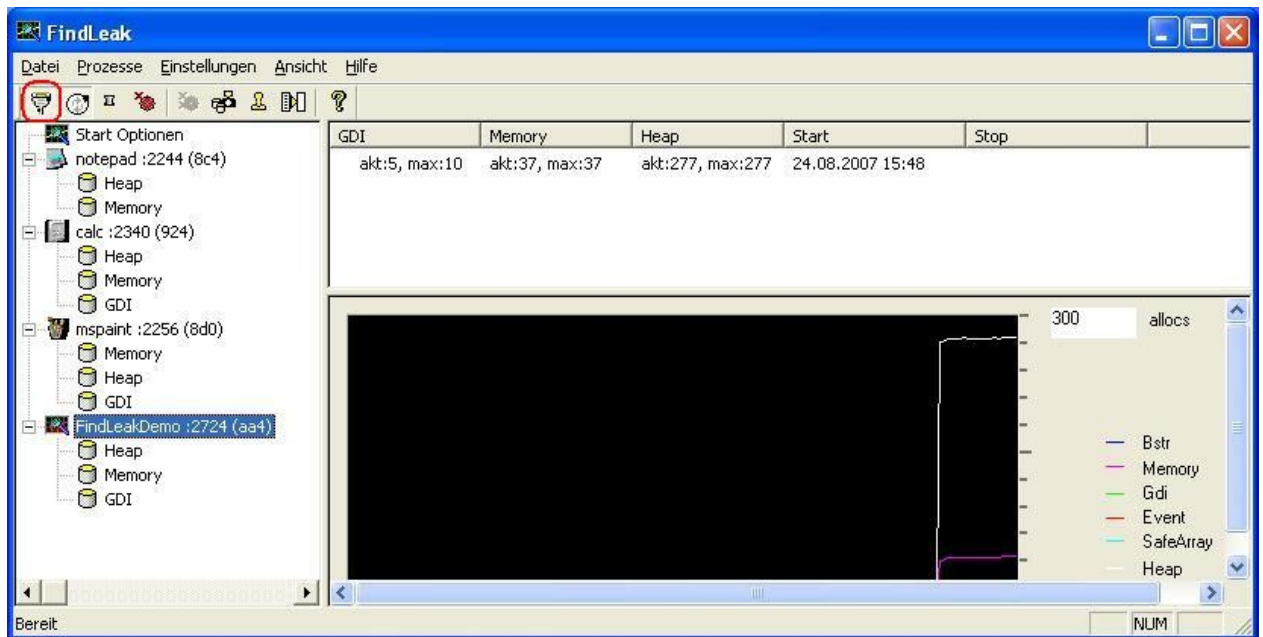
Sollten Sie Fragen zu diesem Workshop oder zu FindLeak haben, nehmen Sie bitte Kontakt mit uns auf. Unsere Mitarbeiter beraten Sie gerne.

Im Menü „Hilfe“ unter dem Menüpunkt „Registrieren“ erfahren Sie, wie Sie eine „FindLeak“-Vollversion erwerben können.

11. Arbeiten mit FindLeak

11.1 Finden von doppelten Freigaben

Tritt in einem von FindLeak überwachten Prozess eine doppelte Freigabe auf, wird der Prozess an dieser Stelle angehalten. Es wird entweder eine MessageBox ausgegeben oder (wenn der Prozess in einem Debugger läuft) ein Debugbreak aufgerufen. Wird eine MessageBox angezeigt, muss der Prozess in einen Debugger genommen werden und durch einen „Break-Aufruf“ angehalten werden. Bevor Sie den Debugger starten, sollten Sie in FindLeak "Prozesse anhängen" ausschalten, da sonst auch der Debugger auf Memory Leaks untersucht wird, was sehr zeitraubend sein kann.



Den Debugger können Sie auch über das Popup-Menü starten. Hierbei wird, bis der Debugger gestartet ist, automatisch "Prozesse anhängen" deaktiviert. Der Prozess hält also an, sobald der Speicher ein zweites Mal freigegeben werden soll. Der Programmierer kann jetzt im Quelltext feststellen, ob das Freigeben an dieser Stelle unnötig ist. Wenn diese Freigabe aus Programmiersicht in Ordnung ist, kann der Fehler auch bei der ersten Freigabe liegen. Um die Stelle der ersten Freigabe zu finden, muss in FindLeak die Option "show stack" (im Fenster „Prozessüberwachung“) eingeschaltet werden. Ist diese Option aktiv, wird beim Feststellen der zweiten Freigabe der Stack der ersten Freigabe (wenn er ermittelt werden konnte) im Output des Debuggers ausgegeben.

11.2 Untersuchen, ob Memory Leaks vorhanden sind

Erster Schritt:

Nachdem FindLeak gestartet wurde, werden verschiedene Testszenarien gefahren. Danach wird der Prozess beendet. Wurden Ressourcen nicht freigegeben, werden diese nach Beenden des Prozesses in FindLeak angezeigt.

Durch die Lade- und Entladereihenfolge der DLLs kann es dazu kommen, dass FindLeak Memory Leaks

anzeigt, die noch freigegeben werden, nachdem die Spy-DLL entladen wurde. Es können auch Ressourcen sein, die bei der Initialisierung einmalig angelegt wurden und bei denen der Entwickler es nicht für nötig gehalten hat, sie während der Programmlaufzeit freizugeben, da sie beim Beenden des Programms sowieso freigegeben werden.

Bei den meisten Prozessen zeigt FindLeak nach dem Beenden noch Ressourcen an, die nicht freigegeben wurden. Diese Anzeige ist prozessspezifisch und sollte auch bei unterschiedlicher Laufzeit immer gleich sein. Wird das Testszenario mehrfach durchlaufen, bevor der Prozess beendet wird, sollte sich die Anzahl der nicht freigegebenen Ressourcen nicht erhöhen.

Zweiter Schritt:

Nachdem ein Prozess in FindLeak angezeigt wird, kann in der Übersichtsansicht geprüft werden, ob die Anzahl der Allokierungen zunimmt.

Hierzu sollten sie ein Testszenario benutzen, bei dem immer wieder ein definierter Zustand erreicht werden kann. Dieser Zustand kann durch Caching-Mechanismen überlagert sein, so dass sich erst nach einer gewissen Laufzeit, ein Gleichgewicht einstellt. Um Vergleichswerte zu erhalten, kann der aktuelle Zustand über das Kontextmenü der Listenansicht als neuer Eintrag gespeichert werden. Der zuletzt gespeicherte Eintrag, steht ganz unten.

Dritter Schritt:

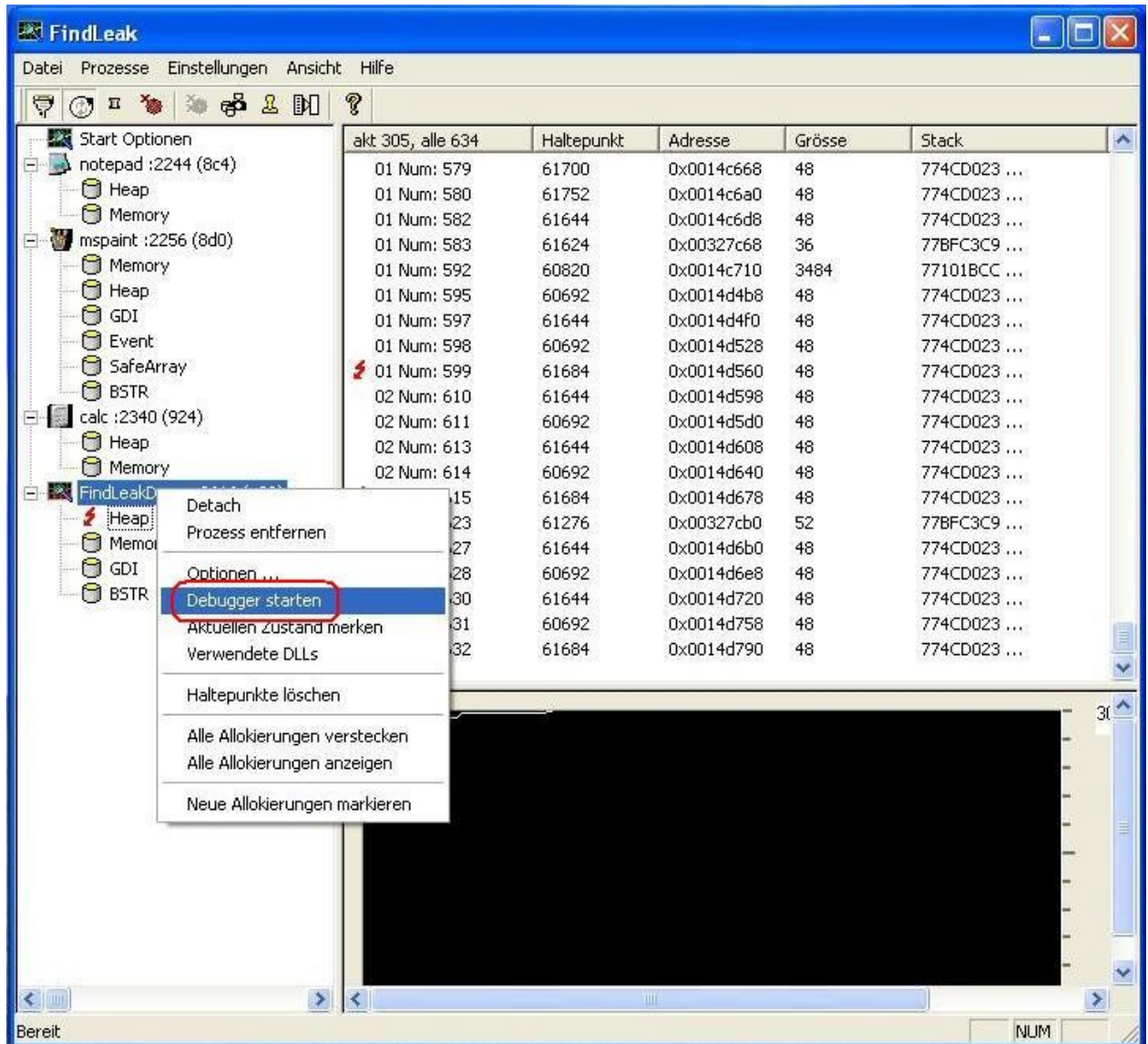
Nachdem ein Prozess in FindLeak angezeigt wird, kann in der Detailübersicht geprüft werden, ob die Anzahl der Allokierungen zunimmt.

Hierzu können zuerst einmal über das Kontextmenü alle bisherigen Allokierungen ausgeblendet werden. Dann sollte, wie unter Schritt zwei beschrieben, ein Testszenario mit einem definierten Endzustand durchlaufen werden. Über das Kontextmenü können jetzt alle neuen Allokierungen markiert werden. Die Markierung erfolgt durch eine Nummer, die der Allokierungsnummer vorangestellt wird. Wird das Testszenario jetzt erneut durchlaufen, sollten alle nicht markierten Einträge verschwinden. Das Markieren und Durchlaufen des Testszenarios kann jetzt mehrfach nacheinander durchgeführt werden. Ist ein Speicherfresser vorhanden, wird zu jeder neuen Markierung ein Eintrag vorhanden sein.

11.3 Beseitigen von Memory Leaks

Um die Stelle im Sourcecode zu finden, an der die nicht freigegebene Ressource angelegt wurde, sollte zuerst der zu untersuchende Prozess in einen Debugger genommen werden.

Dies geschieht am zweckmäßigsten über das Kontextmenü des markierten Prozesses (s. Abb.), da hierbei verhindert wird, dass der Debugger ebenfalls von FindLeak untersucht wird.



Dann muss in der Detailansicht der entsprechende Eintrag selektiert und der Menüpunkt "Anhalten bei Allokierung" aufgerufen werden. Anschließend wird das Testszenario erneut durchlaufen. Sobald der Speicher angelegt wird, läuft der Debugger auf ein Debugbreak. Im Callstack sieht man nun normalerweise, wo der Speicher angelegt wird. Der Entwickler kann jetzt entscheiden, ob diese Ressource noch länger benötigt wird, oder ob sie tatsächlich, obwohl sie nicht mehr benötigt wird, nicht freigegeben wird.

12. Unterschiede zwischen Demo- und Vollversion

In der zeitlich uneingeschränkten FindLeak-Demo können Sie in dem mitgelieferten Microsoft Visual C++ - Beispielprojekt alle Möglichkeiten erforschen, die FindLeak Ihnen bietet. Das Analysieren Ihrer eigenen Projekte ist mit der Demoversion nicht möglich. Nach dem Erwerb eines FindLeak-Lizenzschlüssels, können Sie diese Demo frei schalten und erhalten so die uneingeschränkte Vollversion

13. Registrierung / Freischaltung

Die Demo-Version von FindLeak erlaubt ausschließlich das Arbeiten mit der mitgelieferten Demoanwendung. Um Ihre Version von FindLeak registrieren und frei zu schalten, gehen Sie folgendermaßen vor:

Klicken Sie im Menü auf „Hilfe“ und danach auf „Registrierung“. Der folgende Dialog erscheint:

sales@findleak.de'. Below the text are three input fields: 'Identification:' with the value '205 30B 345 684 0A3 914 C24 155 7F4 F84 864 444 E14', 'Name:', and 'Company:'. Below these is a button labeled 'Order form ...'. At the bottom, there is a text prompt 'If you received a license key, enter it here.' followed by an empty input field. At the very bottom are 'OK' and 'Cancel' buttons."/>

Register FindLeak

You can order a full-version of FindLeak here. Compared to the demo version, the full version is not limited to analyzing the FindLeak demo project.
The license is user- and hardware-dependent; this means that it is valid for the computer your demo version is currently installed on.
To order a FindLeak license, click on the button "Order form". An order form is generated containing your user data; please fax a signed copy of this form to us.
Detailed information on ordering FindLeak can be found on the form.

If you still have questions about ordering FindLeak, please send us an E-Mail to: sales@findleak.de

Identification: 205 30B 345 684 0A3 914 C24 155 7F4 F84 864 444 E14

Name: [REDACTED]

Company: [REDACTED]

Order form ...

If you received a license key, enter it here.

[REDACTED]

OK Cancel

1. Registrieren / Kaufen der Vollversion

Klicken Sie auf den Button „Zum Bestellformular“. Ein Bestellformular wird erzeugt und angezeigt. Drucken Sie dieses aus und ergänzen Sie die dafür vorgesehenen Felder. Nachdem Ihr ausgefülltes Bestellformular bei uns eingegangen ist (per Fax oder per Post), generieren wir aus

Ihren Adressdaten und der Kennung, die in obigem Dialog angezeigt wird, einen Lizenzschlüssel. Dieser berechtigt Sie zur Nutzung von FindLeak ausschließlich auf dem Computer, auf dem die Kennung erzeugt wurde.

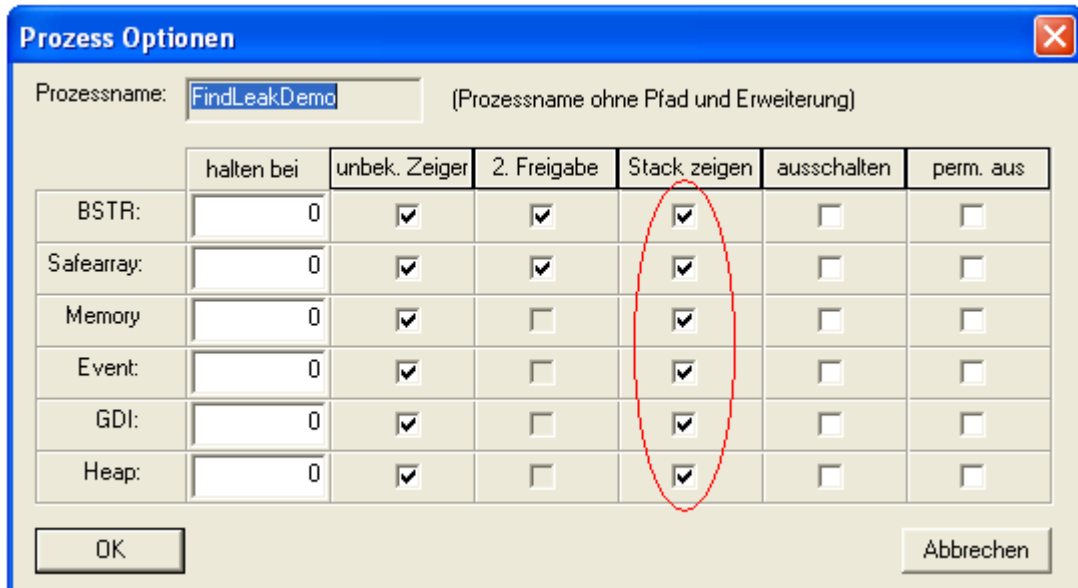
2. Freischaltung

Sobald Sie den Lizenzschlüssel von uns erhalten haben, starten Sie FindLeak und bringen den obigen Dialog zur Anzeige. In dem noch freien Feld geben Sie den Lizenzschlüssel ein. Achten Sie dabei genau auf Groß- und Kleinschreibung. Wenn Sie den Lizenzschlüssel eingegeben haben, klicken Sie auf OK. Ihre Eingabe wird nun überprüft. Wenn der von Ihnen eingegebene Lizenzschlüssel von FindLeak als gültig erkannt wird, wird Ihnen die Freischaltung bestätigt. Nach einem Neustart von FindLeak ist die Freischaltung abgeschlossen.

14. Probleme und Lösungen

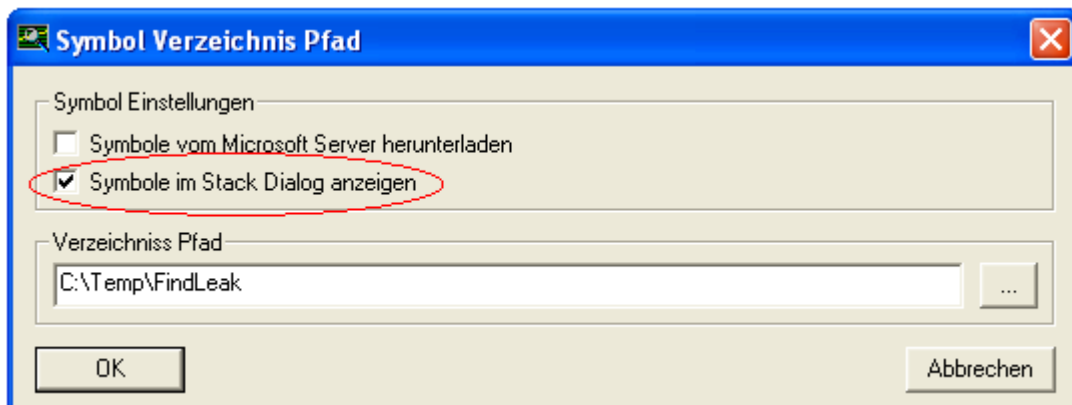
Es wird kein Callstack angezeigt

Bitte überprüfen Sie die Einstellungen in dem Dialog „Prozess Optionen“, ob die entsprechende Häkchen unter „Stack zeigen“ markiert sind. Durch Anklicken auf den „Stack zeigen“ Button werden alle Häkchen in der Spalte markiert.



Es wird kein Funktionsname im Callstack angezeigt

Bitte überprüfen Sie die Einstellungen in dem Dialog „Symbol Verzeichnis Pfad“. Das entsprechende Häkchen „Symbole im Stack Dialog anzeigen“ muss markiert werden.



Ist die zu untersuchende Applikation mit **Microsoft VisualC++ 6.0 mit ServicePack5** (oder kleiner), kann der Funktionsname und die Zeilennummer nicht angezeigt werden.

Es wird im Debugger immer nur Code in System-DLLs angezeigt

Dies tritt vor allem bei Aufrufen in der Oleaut32 auf. Der Speicher wird hierbei normalerweise von der Oleaut32.dll selbst angelegt, um damit ein Interface zu versorgen. Der Fehler liegt dann meist

in der Automationsfunktion, die aber durch Step-Over Befehle nicht erreicht werden kann. Hier kann man es mit „Step Into“-Befehlen probieren, was aber ziemlich langwierig ist und oft auch nicht zum gewünschten Erfolg führt. Es hilft meist weiter, wenn man sich die allokierte Ressource (BSTR, SafeArray) in FindLeak anschaut und dadurch Erkenntnisse gewinnt, welche Funktion aufgerufen werden sollte. Der Speicher an der gefundenen Stelle wird mit Sicherheit freigegeben. Der Mechanismus, der eine Codestelle findet, an der eine Ressource angelegt wird, ist nicht 100%ig eindeutig. Es können ähnliche Codestellen von FindLeak als gleich betrachtet werden. Hier hilft es nur, den Debugger weiterlaufen zu lassen. Er müsste dann noch an einer anderen Stelle im Code anhalten, an der der Speicher tatsächlich angelegt wird.

15. Fragen und Antworten

Worin unterscheidet sich die kostenlose Demo-Version von der FindLeak-Vollversion?

Mit der Demo-Version können Sie lediglich das mitgelieferte Demo-Projekt bearbeiten, um so alle Möglichkeiten von FindLeak kennen zu lernen.

Ich kann das Beispiel aus dem Workshop nicht nachvollziehen. Mein Debugger scheint auf die Haltepunkte nicht zu reagieren. Woran kann das liegen?

Die mitgelieferte Beispiel-Applikation wurde mit **Microsoft VisualC++ 6.0 mit ServicePack6** erstellt. Um das Beispiel wie beschrieben nachvollziehen zu können, müssen Sie die gleiche Entwicklungsumgebung mit installiertem **ServicePack6 oder einer höheren Version von Microsoft VisualC++** benutzen.

Kann FindLeak gleichzeitig mit anderen Tools zum Finden von Memory Leaks eingesetzt werden?

Nein. Aufgrund des inneren Aufbaus derartiger Tools ist dies nicht möglich. Sie können zu einem bestimmten Zeitpunkt immer nur eines dieser Tools verwenden.

Ein Blick in den Systemmonitor zeigt mir, dass der Speicherbedarf des zu untersuchenden Prozesses erhöht ist, obwohl sich in diesem Prozess keine Memory Leaks befinden.

Woran liegt das?

Während FindLeak aktiv ist, erhöht es den Speicherbedarf des zu untersuchenden Prozesses und lässt ihn auch steigen, wenn keine Memory Leaks vorhanden sind.

Eine Überprüfung des Speicherbedarfs eines Prozesses mit dem Systemmonitor oder dem Task Manager ist nur dann sinnvoll, wenn FindLeak nicht aktiv ist.

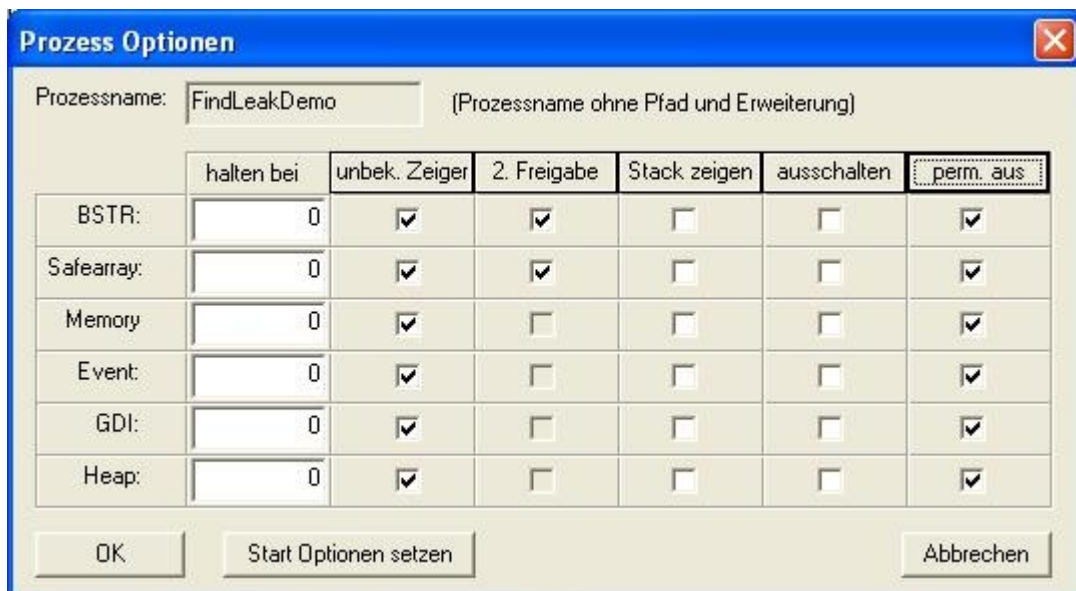
Es werden immer alle neu gestarteten Prozesse in FindLeak angezeigt. Ich möchte aber nur bestimmte Prozesse untersuchen.

Wenn Sie nur bestimmte Prozesse untersuchen wollen, klicken sie in der Baumansicht auf "Start Optionen".

In der Listenansicht rechts erscheint nun der Eintrag <Other>, den sie bitte doppelklicken (s. Abb.).



In dem nun erscheinenden Dialog "Prozess Optionen" aktivieren sie durch Ankreuzen die Option "Permanent aus" für alle Speichertypen (siehe nächste Abb.).



Die Wahl dieser Option bewirkt, dass ab sofort kein Prozess mehr automatisch aufgenommen wird. Damit Ihre Prozesse wieder angezeigt werden, fügen Sie für jeden Prozess einen neuen Eintrag in die Prozessliste ein. Starten Sie danach FindLeak neu.

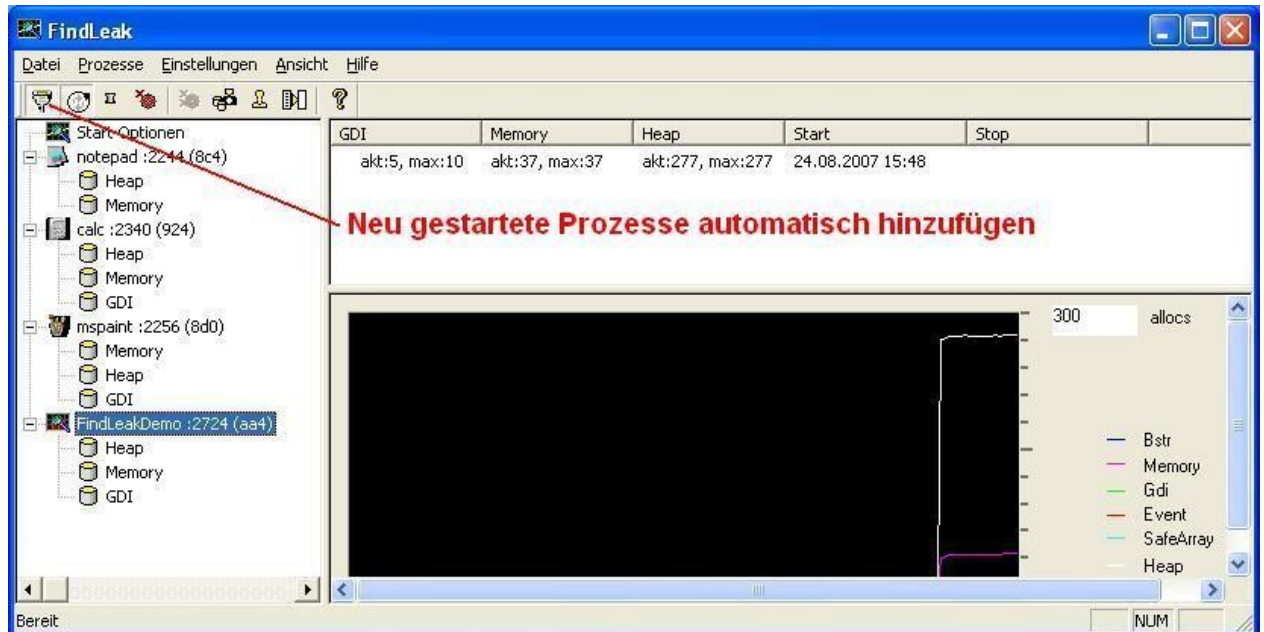
Wenn FindLeak aktiv ist, ist das System deutlich langsamer als zuvor. Kann ich dagegen etwas tun?

Ja. Um die Performance zu verbessern, können Sie die Untersuchung des Speichers in zwei Schritte aufteilen. Deaktivieren Sie zuerst den Speichertyp "heap" und untersuchen Sie dann, ob Memory Leaks auftreten. Treten bei den übrigen Speichertypen keine Memory Leaks auf, deaktivieren Sie alle Speichertypen mit Ausnahme von "heap". Untersuchen Sie dann erneut.

Die Performance des Gesamtsystems kann auch sehr schlecht werden, wenn in FindLeak der Speicher in der Detailansicht betrachtet wird und gleichzeitig die zyklische Aktualisierung aktiviert ist.

Wie kann ich Prozesse im Arbeitsspeicher von der Überwachung durch FindLeak ausschließen?

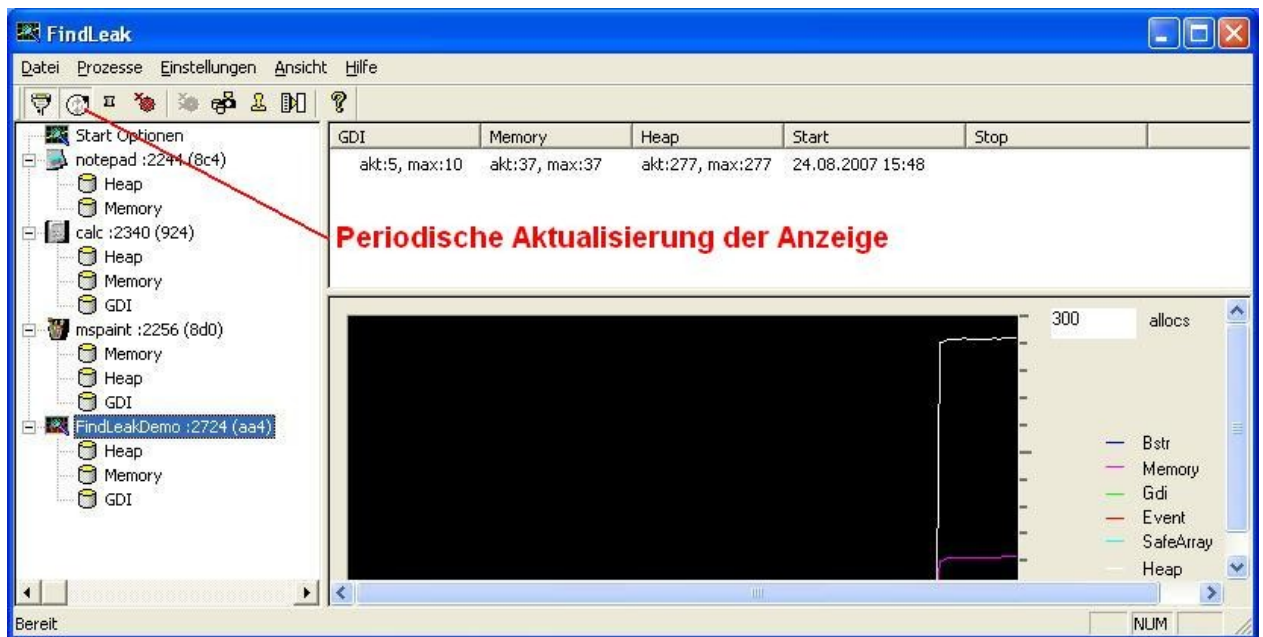
Entweder Sie deaktivieren die Funktion "Neu gestartete Prozesse anzeigen", indem sie einfach den ersten Icon anklicken (s. Abb.),



oder Sie klicken im Menü "Einstellungen" auf den Menüpunkt "Prozesse anhängen", so dass das Häkchen vor diesem Menüpunkt verschwindet. Weitere Prozesse, die nach dem Ändern dieser Einstellung gestartet werden, bleiben nun von der Überwachung durch FindLeak ausgeschlossen. Sie können diese Einstellung jederzeit durch ein erneutes Klicken wieder auf den ursprünglichen Zustand (Überwachung aller Prozesse) zurücksetzen. Prozesse, die danach gestartet werden, werden nun wieder von FindLeak überwacht.

Wie kann ich die Arbeit mit FindLeak beschleunigen?

Hierzu deaktivieren Sie den Icon "Ständige Aktualisierung der Anzeige" (s. Abb.), indem Sie auf diesen klicken.



Oder klicken Sie im Menü "Einstellungen" auf den Menüpunkt "Laufend aktualisieren", so dass das Häkchen vor diesem Eintrag verschwindet. Ab sofort wird die Listenansicht mit der Anzeige der Speicherallokierungen nicht mehr laufend aktualisiert. Dieses Ausschalten erhöht die Arbeitsgeschwindigkeit spürbar, da Ihr Rechner nun nicht mehr durch das ständige Neuzeichnen des Ausgabefensters belastet wird.

Die zyklische Aktualisierung wird erst wieder aktualisiert, wenn Sie ein weiteres Mal auf diesen Menüpunkt bzw. das entsprechende Icon klicken.