

# メモリリークと調査方法

タイコエレクトロニクス株式会社

萩原 智恵

# それはある日突然に

- 納品したユーザ様の先から電話がかかってきました。

ユーザ:

「アプリケーションが  
午後になると固まるんです！！！」

私:

「へ?! 午後……ですか? (^\_^;)」

# それはある日突然に

私:

「そんな、、、タイマーを使う(見る)ような、  
アプリケーションは今回作った  
覚えはありません！！！」

ユーザ:

「いや～。**Out Of Memory** っていう  
エラーがでるんです」

私:

「 Out Of Memory ? ? ? ? ? .....

いやいや～。またまた、悪い冗談を。。。」

# .NETとマネージアプリケーション

- .NETのアーキテクチャはマネージコードであるはず!?!?

- **Out Of Memory!!!**

マネージコードでのメモリーリークなんて

信じられな—い!?!? (バレンタイン監督風に!?)

# マネージコードとは？

- マイクロソフトが提唱する次世代インターネット戦略、Microsoft.NET プラットフォームにおいて、アプリケーションやコンポーネントを実行するためのエンジン
- CLR (Common Language Runtime) に準拠し、CLR が提供するすべての機能を利用可能なプログラム・コード。具体的には、CLR が提供するガベージ・コレクションの機能や、セキュリティー機能などが利用できる。プログラムは CLR によって完全に管理される (manage される)

# ガベージコレクション(GC)とは

- 長年、C言語開発で苦しんできた私にとっては神様！！のような存在！？
- プログラム作成者が明示的にメモリの確保・解放を行う必要が無い(はず？)
  - 使用されなくなったメモリーはきれいにお掃除や解放をしてくれる(はず？)

# どうして？

- マネージ(管理)されたコードなのに？
- そして何より、マネージコードには優秀なガベージコレクション(GC)があるはず！  
なのに！なんで、なんで、**なんで！**

「Out Of Memory!!!」

# 解決のために何をすべきか？

- その1

~~お昼になったら一度ログアウトしてください~~

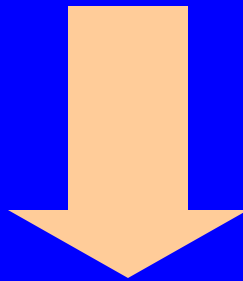
- その2

メモリーリークが発生している場所を特定する



# メモリアリークの場所を特定する

- メモリアリークの場所を特定するために有効な手段はなんでしょう?????

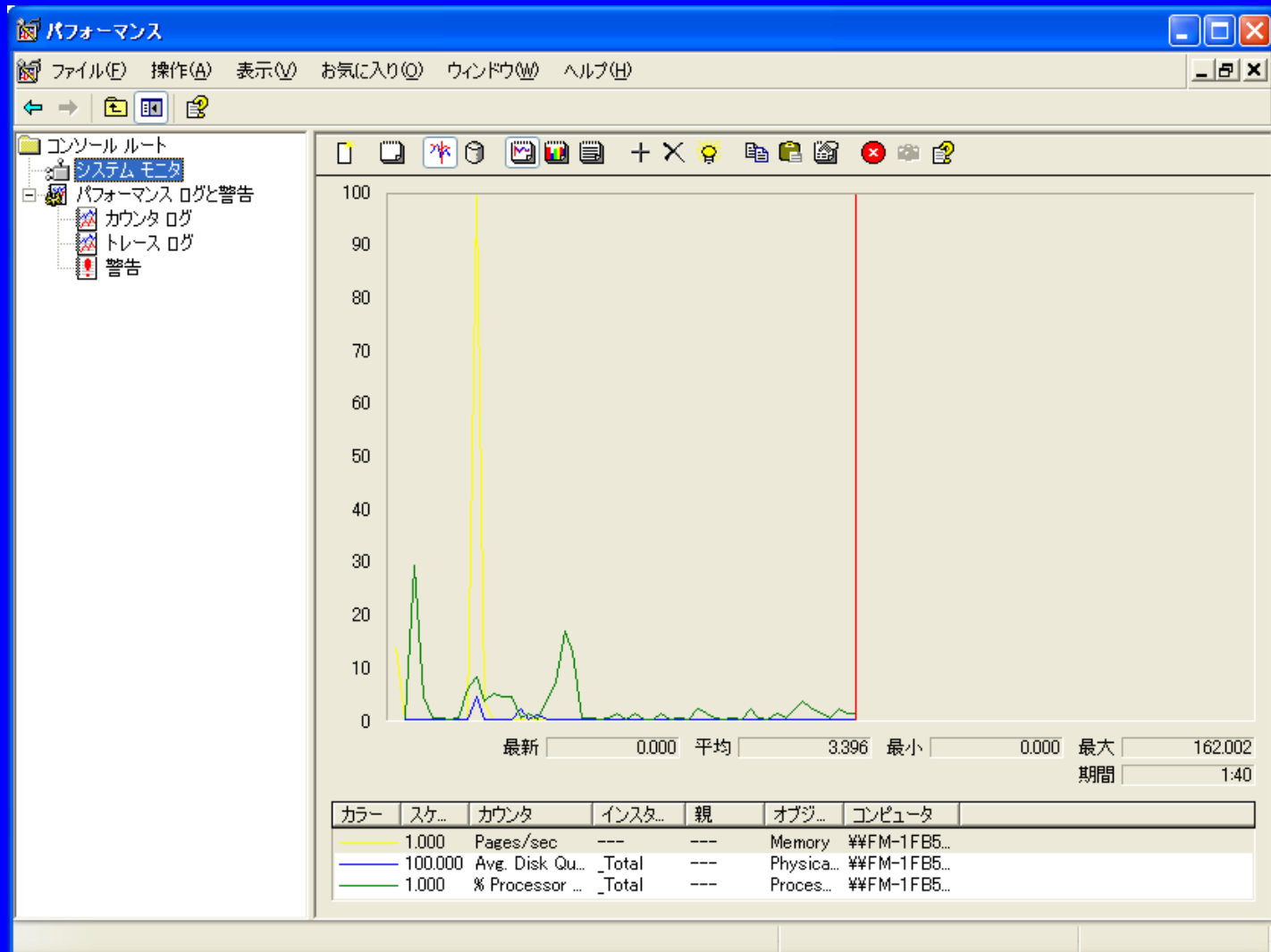


パフォーマンスモニター

# パフォーマンスモニターとは？

- Windows 信頼性とパフォーマンス モニタは Microsoft 管理コンソール (MMC) スナップインの 1 つで、システム パフォーマンスを分析するためのツール
- 1 つのコンソールから、アプリケーションとハードウェアのパフォーマンスをリアルタイムに監視したり、ログに収集するデータをカスタマイズしたり、警告と自動操作のしきい値を定義したり、レポートを生成したり、さまざまな方法で以前のパフォーマンス データを表示したりすることができます

# パフォーマンスモニターとは？



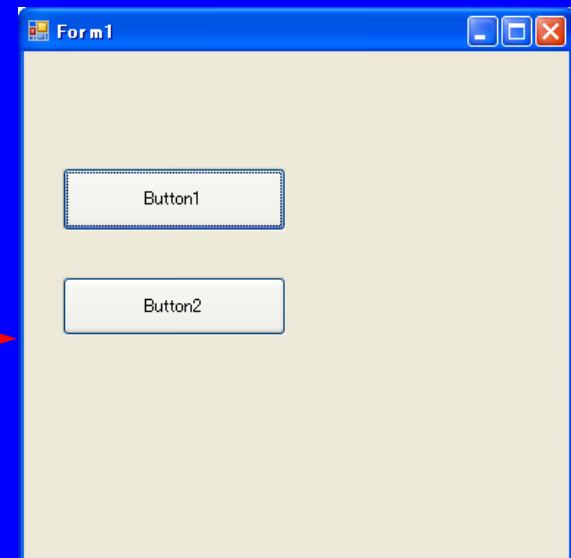
# パフォーマンスモニターを 使ってみよう！

- 実際にパフォーマンスモニターでログ収集を行ないメモリーリークが本当に起こっているかを調べ、場所を特定します
- デフォルトのカウンタではメモリーリーク調査に適切なカウンタが表示されていないので追加する必要があります

# パフォーマンスモニターを起動 する前に

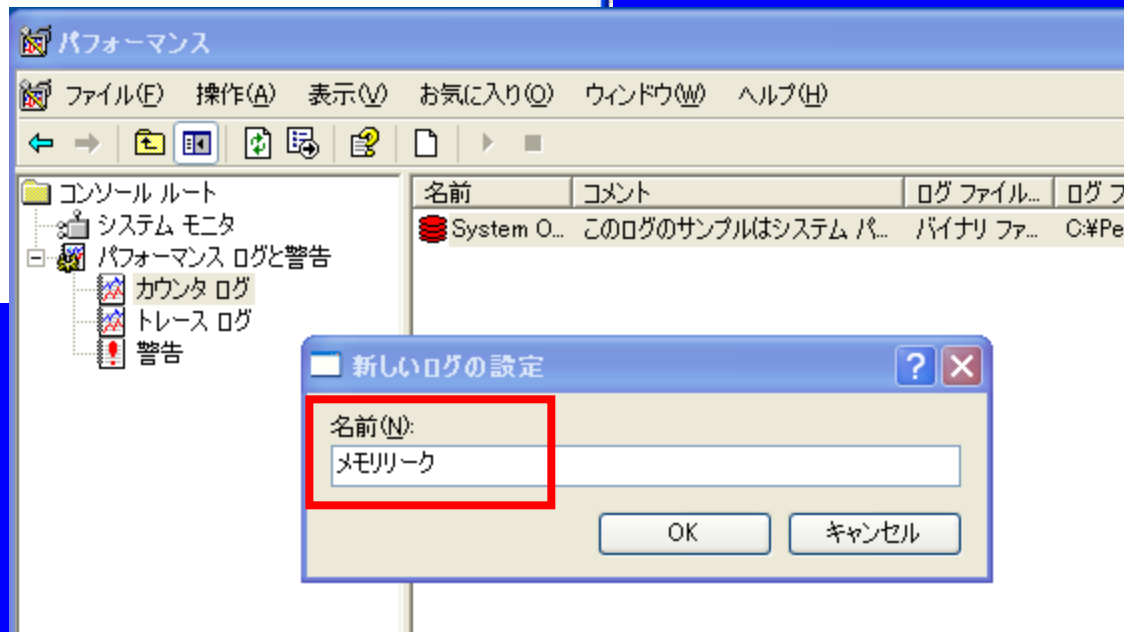
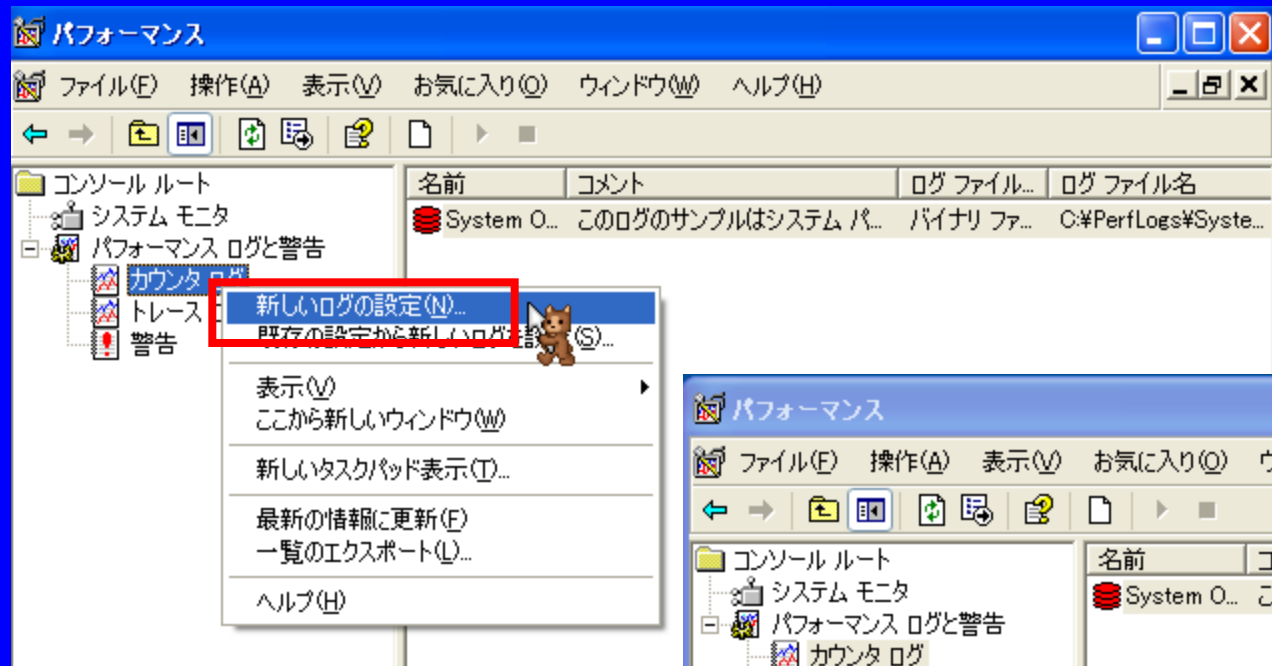
- パフォーマンスモニターを起動し、ログを取る前に、調査する対象のアプリケーションを起動しておく必要があります

名前	サイズ	種類	更新日時
MemoryLeaktest.exe	32 KB	アプリケーション	2008/11/16 23:16
MemoryLeaktest.pdb	44 KB	中間ファイル	2008/11/16 23:16
MemoryLeaktest.xml	1 KB	XML ドキュメント	2008/11/16 23:16



調査対象のアプリケーション  
を先に起動しておく

# パフォーマンスモニターを 使ってみよう！



# パフォーマンスモニターを使ってみよう！

The image shows a Windows Performance Monitor configuration window titled "メモリーーク" (Memory Leak). The "全般" (General) tab is active, showing the log file name "C:\PerfLogs\メモリーーク\_000001.blg". The "カウンタ(C):" (Counters) list is empty. A red box highlights the "カウンターの追加(D)..." button. A callout bubble points to this button with the text: "調査したいメモリ系のカウンタの追加を行います" (Add the counters you want to investigate in the memory system).

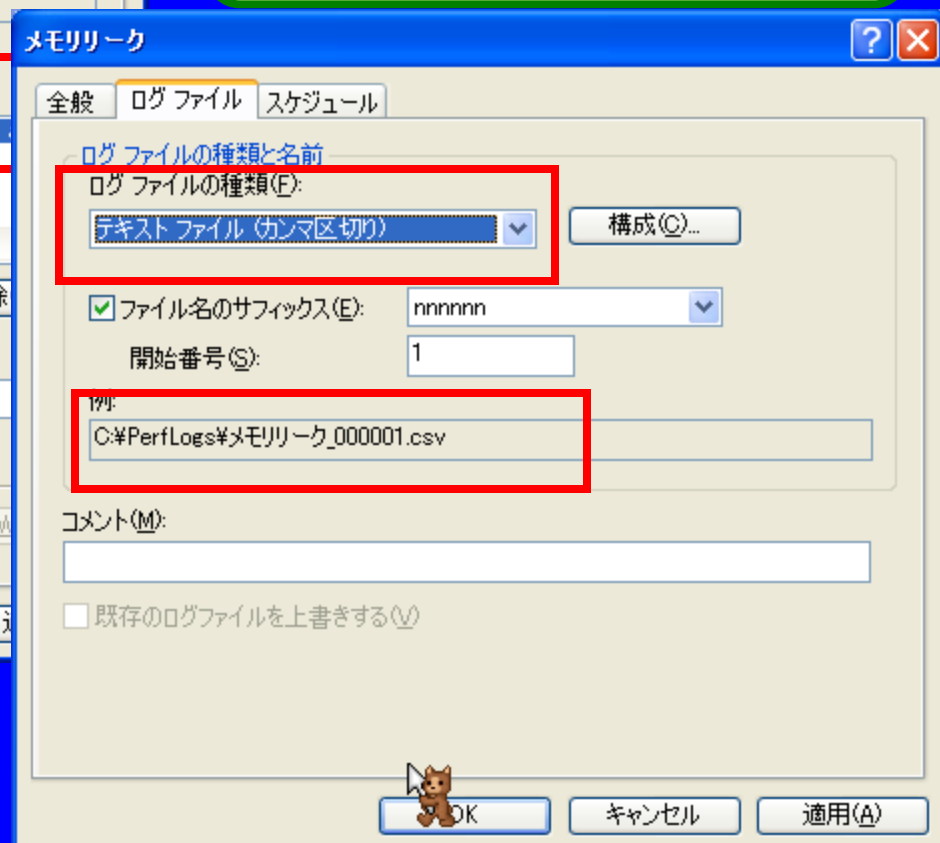
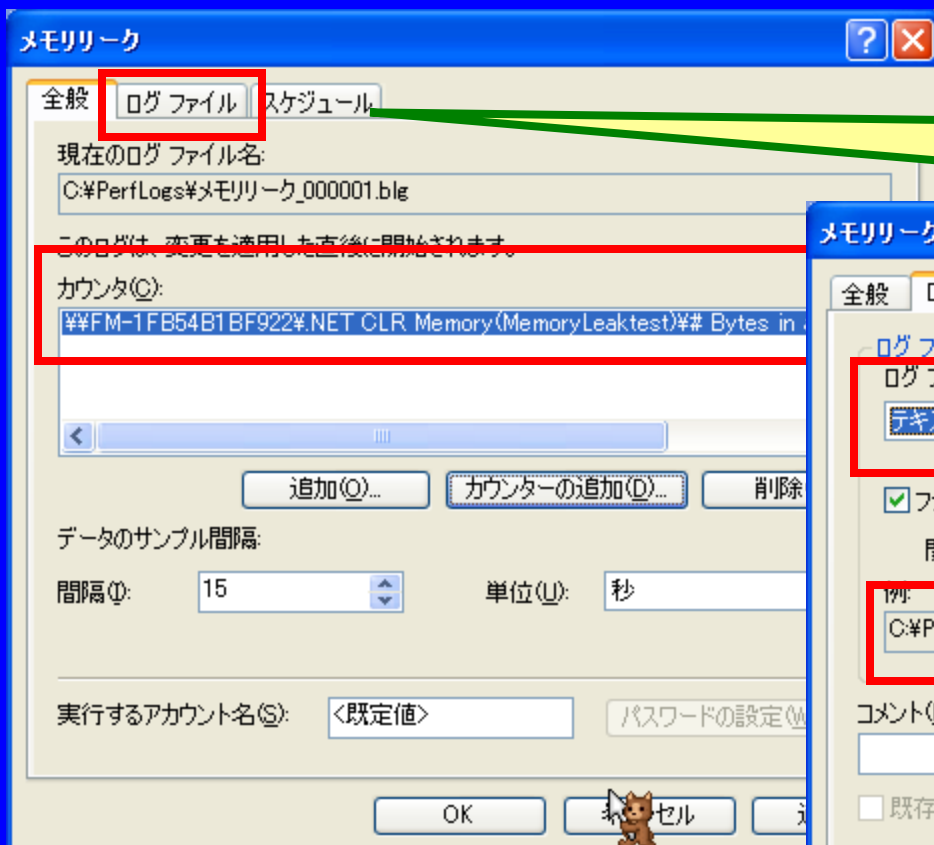
The "カウンタの追加" (Add Counter) dialog box is open. It shows the computer name "¥¥FM-1FB54B1BF922". The "パフォーマンス オブジェクト(O):" (Performance Object) is set to ".NET CLR Memory". The "一覧からカウンタを選ぶ(T)" (Select counter from list) radio button is selected. The list of counters includes "# Bytes in all Heaps", "# GC Handles", "# Gen 0 Collections", "# Gen 1 Collections", "# Gen 2 Collections", and "# Induced GC". A red box highlights "# Bytes in all Heaps". A callout bubble points to this counter with the text: "起動しておいた調査したいアプリケーションを選択します" (Select the application you want to investigate that is already running). The list also shows "Global MemoryLeaktest" selected. A red box highlights the "追加" (Add) button at the bottom of the dialog.

Buttons in the main window include "追加(O)..." (Add...), "カウンターの追加(D)..." (Add Counter...), "削除(R)" (Remove), "データのサンプル間隔:" (Data sampling interval), "間隔(O):" (Interval), "単位(U):" (Unit), "実行するアカウント名(S):" (Run as user), and "適用" (Apply).

Buttons in the dialog include "追加" (Add), "説明(E)" (Help), and "閉じる" (Close).

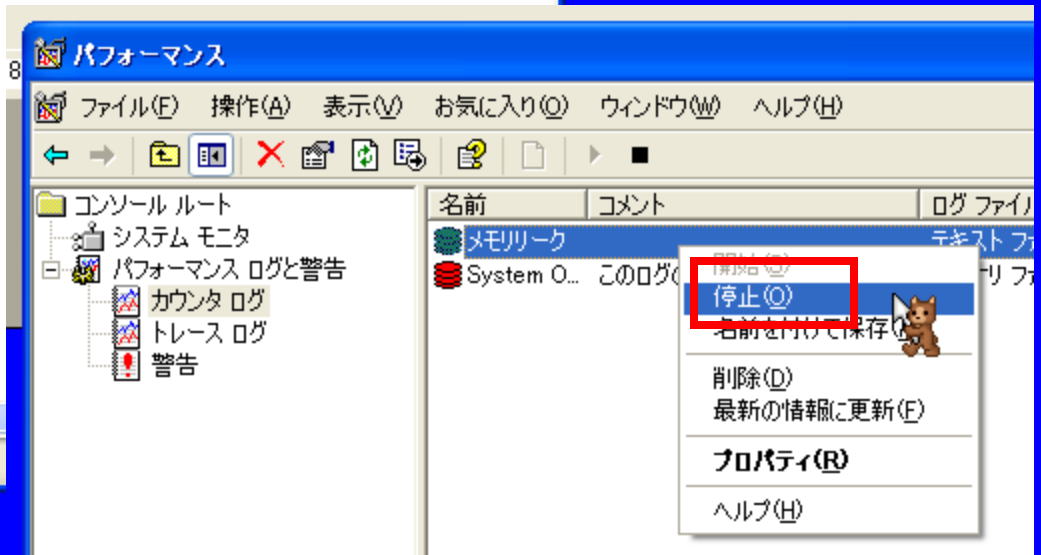
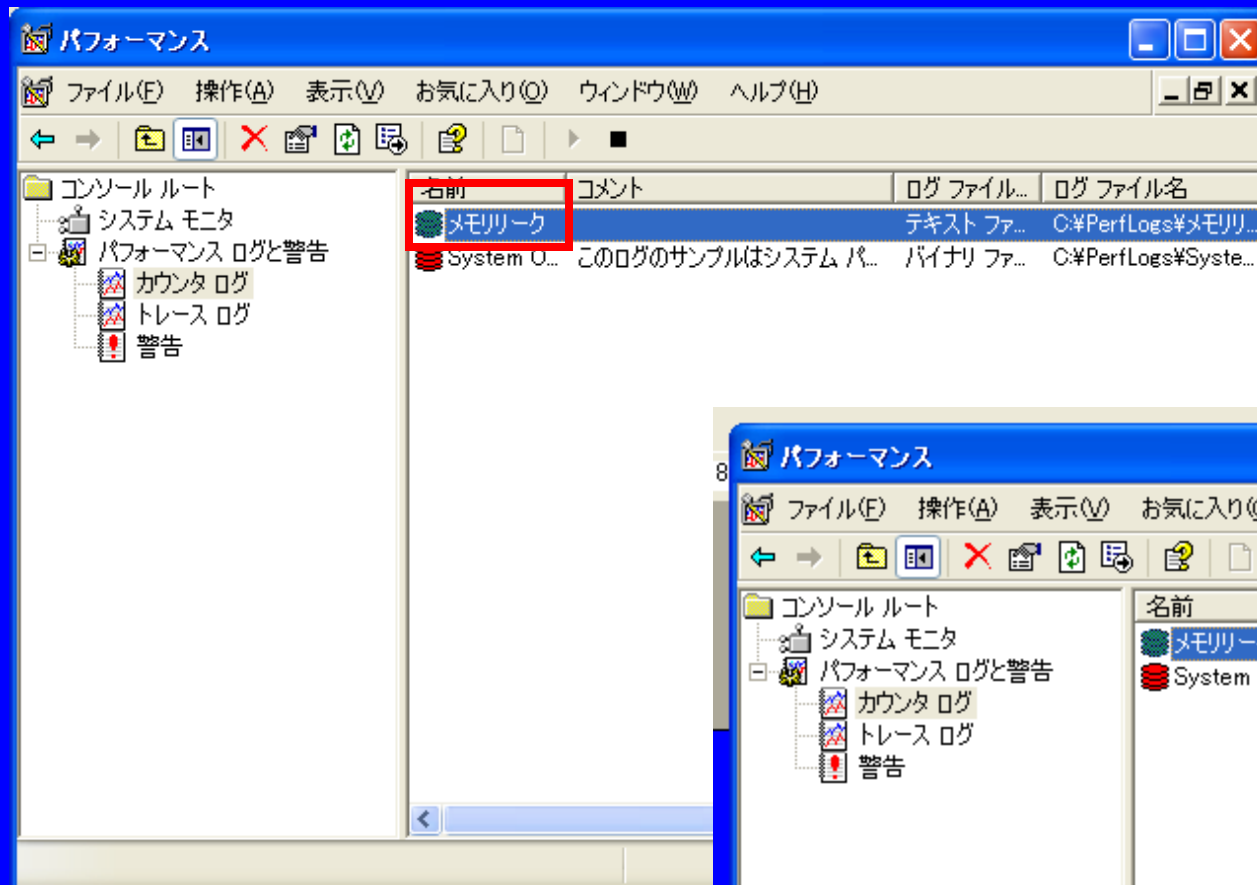
# パフォーマンスモニターを 使ってみよう！

情報を収集し、結果を  
ログファイルに出力します

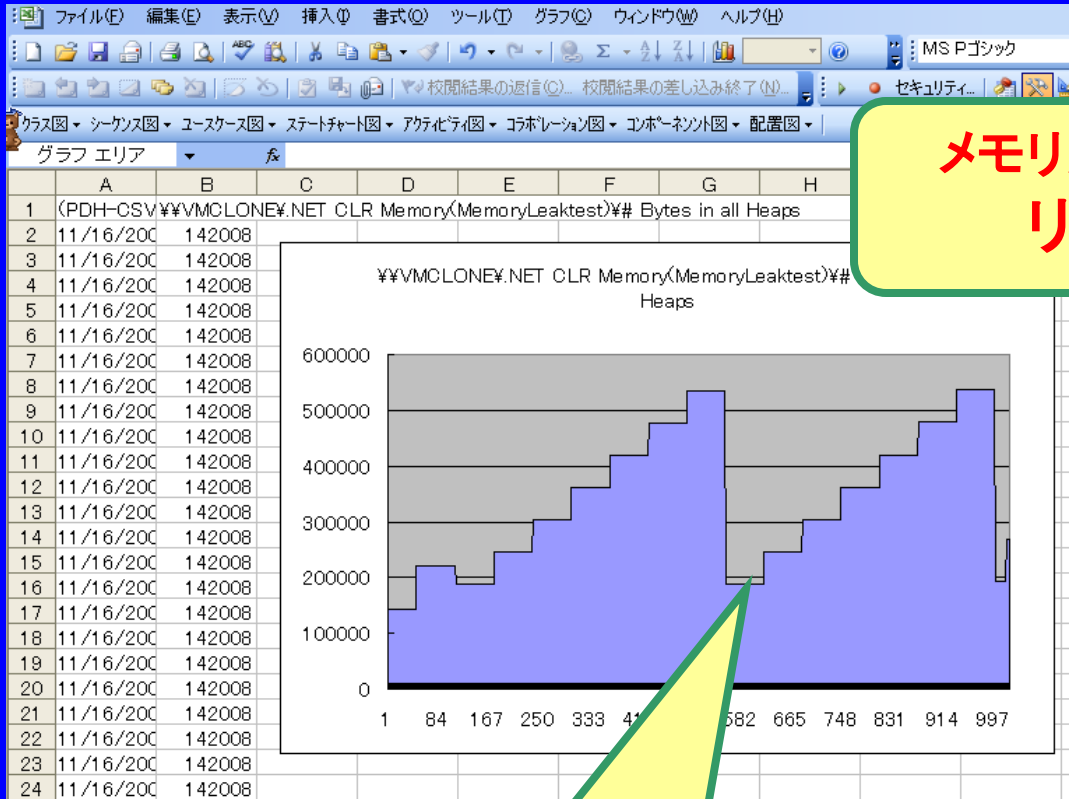




# パフォーマンスモニターを 使ってみよう！



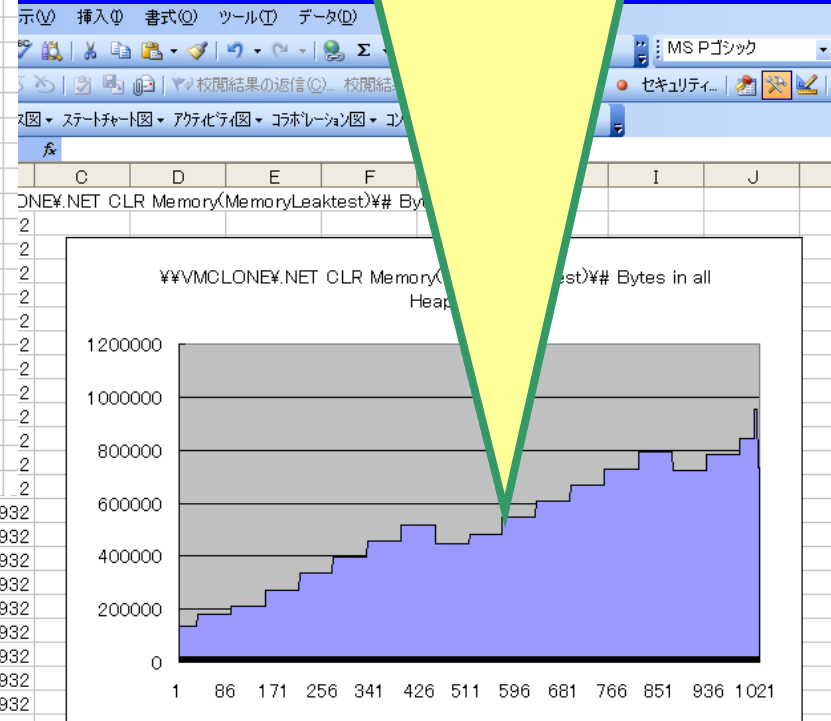
# ログを解析してみよう！



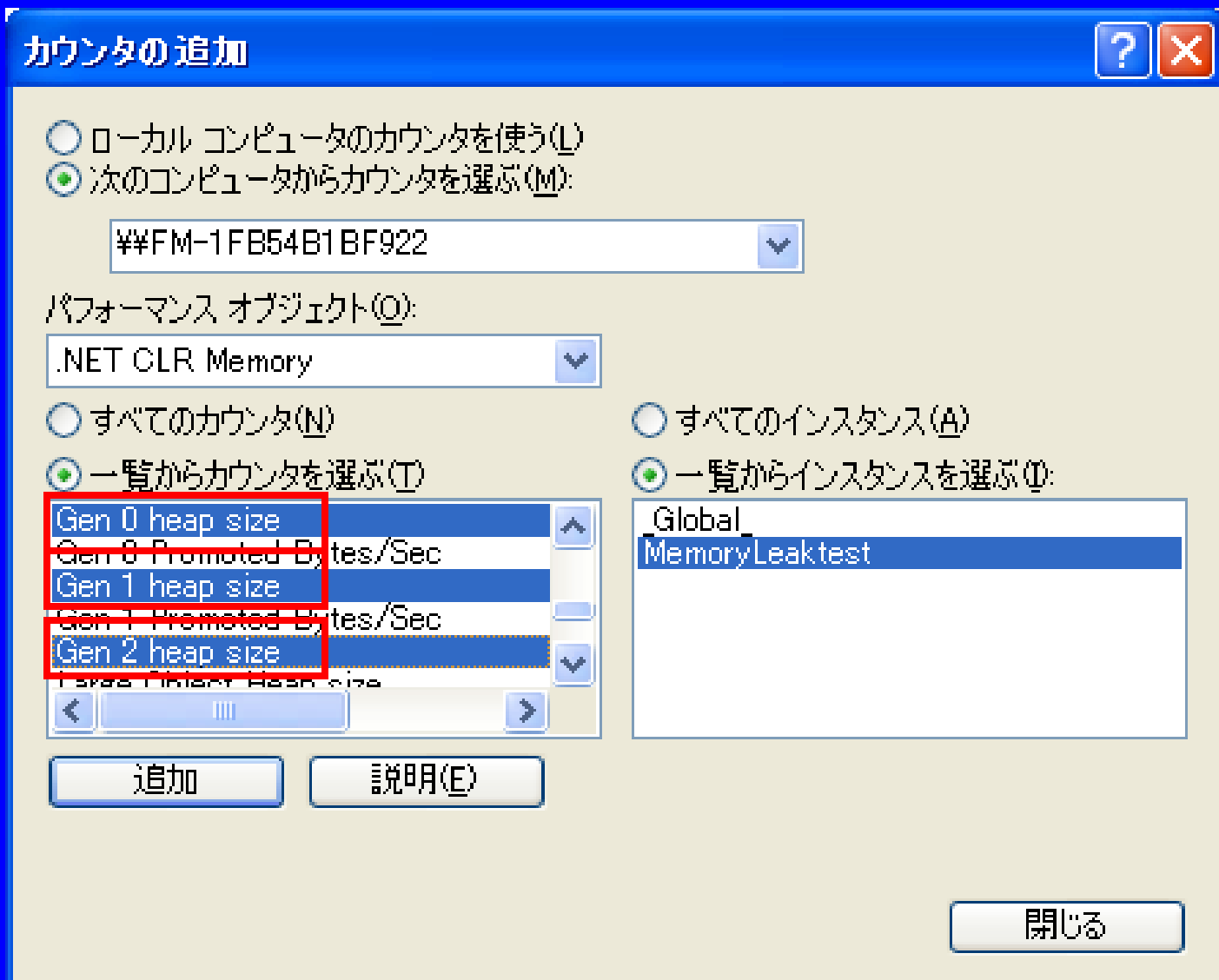
メモリが解放されず、増え続けている  
リークは起きている！！

メモリが解放されている  
リークは起きていない

14	11/16/200	136932
15	11/16/200	136932
16	11/16/200	136932
17	11/16/200	136932
18	11/16/200	136932
19	11/16/200	136932
20	11/16/200	136932
21	11/16/200	136932
22	11/16/200	136932
23	11/16/200	136932
24	11/16/200	136932
25	11/16/200	136932



# もっと詳しく内容を見て見たい！



# ちなみに、原因は。。

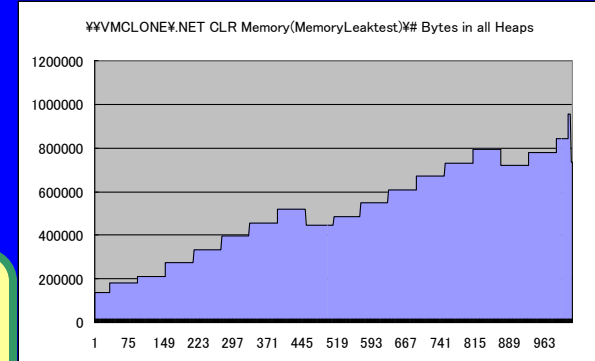
--- 明示的に子フォームをクローズしない ---

```
Dim [frmChild] As New frmChild
```

```
[frmChild].Show(Me)
```

```
Me.Close()
```

メモリーリークを  
起こす



--- 明示的に子フォームをクローズする ---

```
Dim [frmChild] As New frmChild
```

```
[frmChild].Show(Me)
```

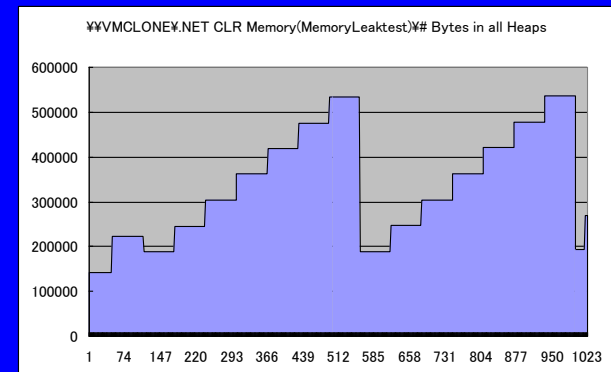
```
[frmChild].Close()
```

```
[frmChild].Dispose()
```

```
[frmChild] = Nothing
```

```
Me.Close()
```

【解決】  
明示的な子ウインドウ  
のクローズが必要



# まとめ

- NET ではメモリに関する心配事が減りますが、アプリケーションのメモリ使用に注意し、適切で効率的に動作することを保証する必要があります。
- アプリケーションが管理されているからといって、優れたソフトウェア エンジニアリング手法を投げ出し、GC が行う魔法に頼ってはいけません。開発やテストのプロセス中は、アプリケーションのメモリのパフォーマンスカウンタを監視し続ける必要があります。アプリケーションの適切な動作が顧客満足につながることを忘れないでください。  
(James Kovacs氏より)

# 参考資料

- ・ マネージコードでのメモリリークの識別と回避  
<http://msdn.microsoft.com/ja-jp/magazine/cc163491.aspx>
- ・ ガベージコレクション入門: Microsoft .NET Framework の自動メモリ管理 Part I  
<http://www.microsoft.com/japan/msdn/net/mag00/GCI.aspx>
- ・ ガベージコレクション入門: Microsoft .NET Framework の自動メモリ管理 Part II  
<http://www.microsoft.com/japan/msdn/net/mag00/GCI2.aspx>