



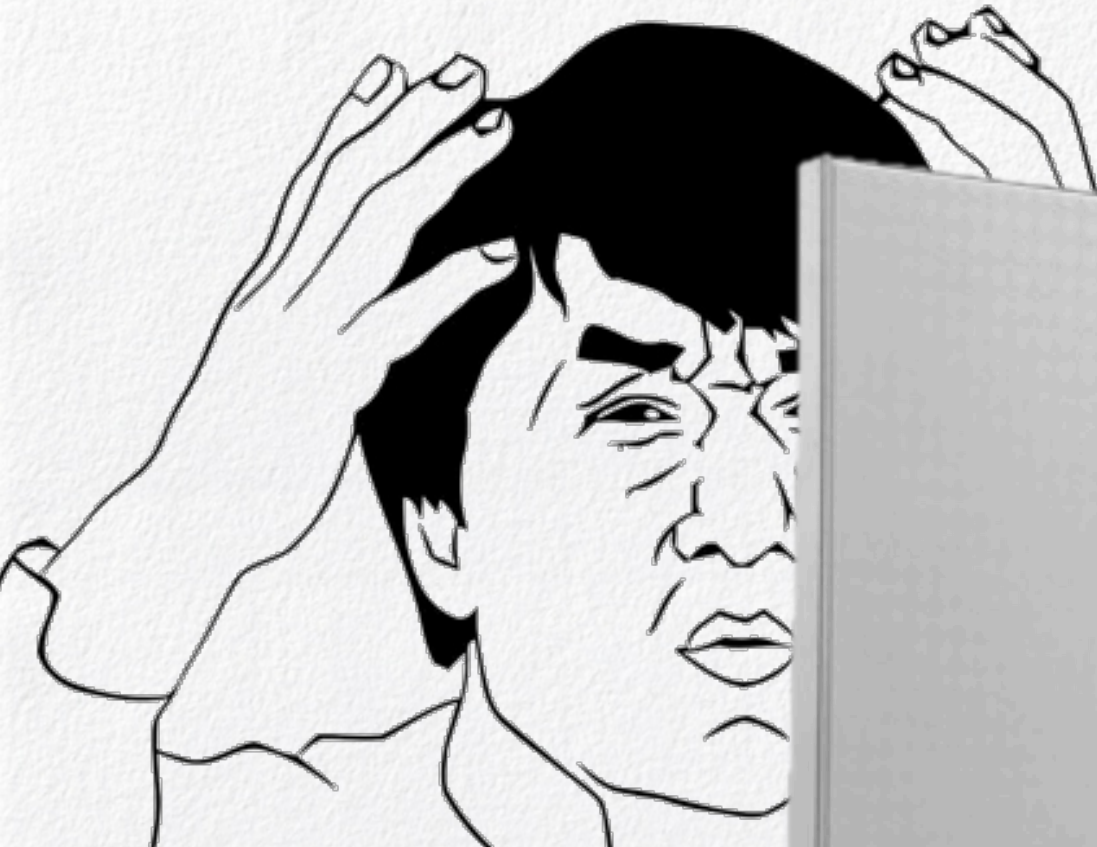
Memory&Performance

Tips&Tricks

Роман Белов

08.12.14

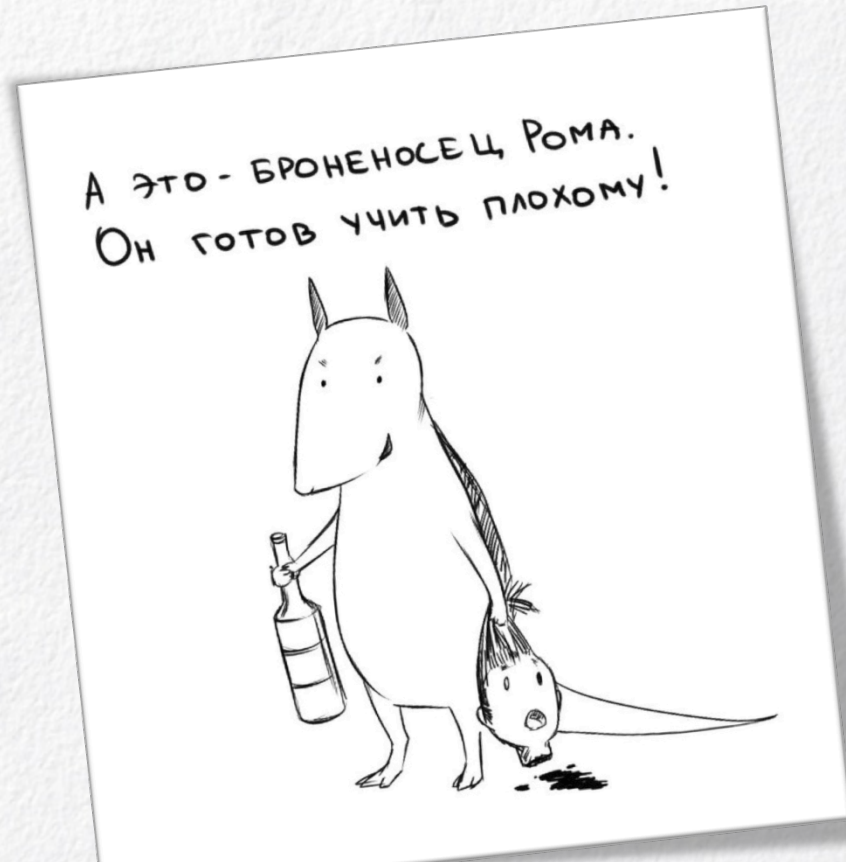
WAT?



Small performance issue

```
return new RGB(  
    Convert.ToInt32(Double.Parse(String.Format("{0:0.00}", r * 255.0))),  
    Convert.ToInt32(Double.Parse(String.Format("{0:0.00}", g * 255.0))),  
    Convert.ToInt32(Double.Parse(String.Format("{0:0.00}", b * 255.0)))  
);
```

Вредные советы



Chapter I.

Утечки памяти

У меня
мемори
лик

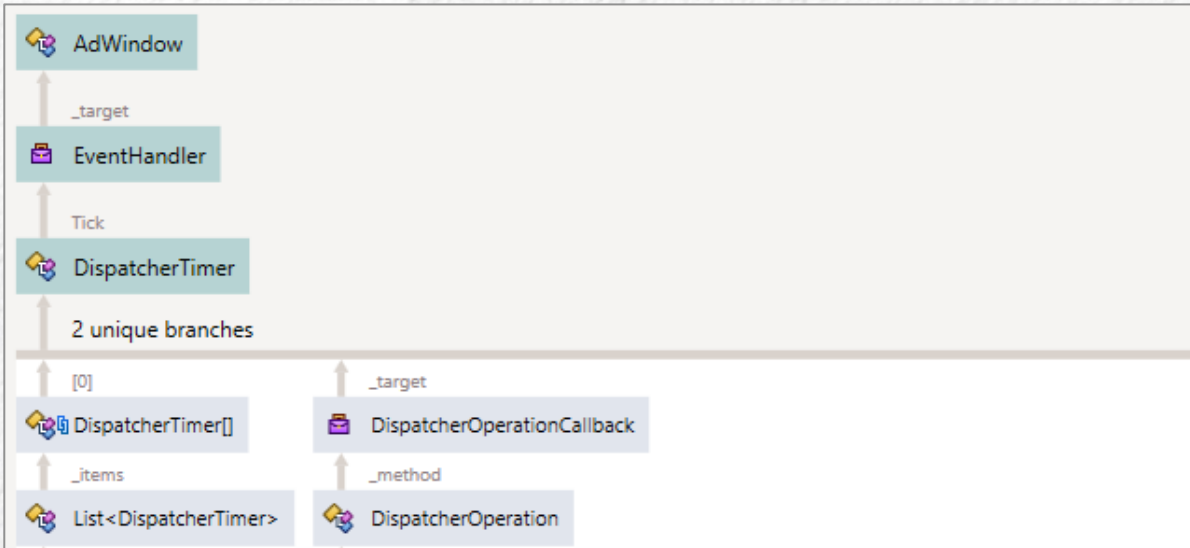


у
меня
тоже

Event Handlers Leak

```
public AdWindow()  
{  
    var adTimer = new DispatcherTimer  
                    {Interval = TimeSpan.FromSeconds(10)};  
    adTimer.Tick += ChangeAd;  
    adTimer.Start();  
}
```

Event Handlers. How to find?



Event Handlers. How to find?

EventHandler
System

Outgoing References

Key Retention Paths

Incoming References

Creation Stack Trace

Shortest Paths to Roots:

List

Tree

Here you can view the stack trace responsible for creating the instance. Calls are shown starting from the last call that directly created the instance descending to the first call in the stack. [Learn more](#)

Function	Namespace
AdWindow.ctor(Window owner)	GameOfLife
MainWindow.StartAd()	GameOfLife
MainWindow.Button_OnClick(Object sender, EventArgs e)	GameOfLife
RoutedEventHandlerInfo.InvokeHandler(Object target, EventArgs e)	System.Windows
EventRoute.InvokeHandlersImpl(Object source, RoutedEventArgs e)	System.Windows
UIElement.RaiseEventImpl(DependencyObject sender, RoutedEventArgs e)	System.Windows
UIElement.RaiseEvent(RoutedEventArgs e)	System.Windows
ButtonBase.OnClick()	System.Windows.
Button.OnClick()	System.Windows.
ButtonBase.OnMouseLeftButtonUp(MouseButtonEventArgs e)	System.Windows.
UIElement.OnMouseLeftButtonUpThunk(Object sender, MouseButtonEventArgs e)	System.Windows

Необходимо
найти место
ПОДПИСКИ

Event Handlers. How to fix?

```
protected override void OnClosed(EventArgs e)
{
    myAdTimer.Tick -= ChangeAd;
    base.OnClosed(e);
}
```

```
public AdWindow(Lifetime lifetime)
{
    var adTimer = new DispatcherTimer;
    adTimer.Tick += ChangeAd;
    lifetime.AddAction(() => adTimer.Tick -= ChangeAd);
    adTimer.Start();
}
```

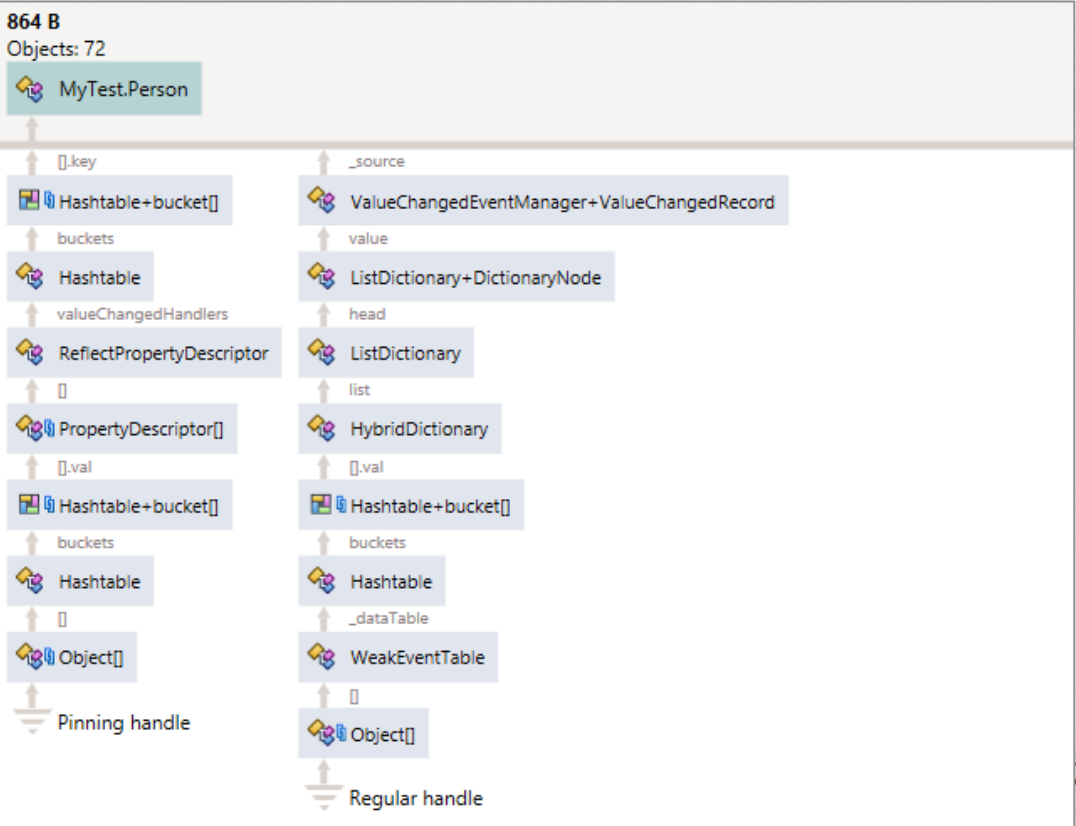
WPF Binding Leak

```
public class Person
{
    public Person(string name)
    {
        Name = name;
    }

    public string Name { get; set; }
}
```

```
<TextBox Text="{Binding Name}"/>
```

WPF Binding. How to find?



Я какой-то не
правильный
биндинг



WPF Binding. How to fix?

```
public class Person : INotifyPropertyChanged
{
    private string myName;
    public Person(string name)
    {
        Name = name;
    }
    public string Name
    {
        get { return myName; }
        set
        {
            myName = value;
            PropertyChanged.Raise(this, nameof.Property(() => Name));
        }
    }
    public event PropertyChangedEventHandler PropertyChanged;
}
```

WPF Binding. How to fix?

```
<TextBox Text="{Binding Name, Mode=OneTime}"/>
```

WPF Collection Binding Leak

```
<DataTemplate DataType="PostCards">  
  <ItemsControl ItemsSource="{Binding Collection}"/>  
</DataTemplate>
```

```
public class PostCards  
{  
    public PostCards(MyBigCollection postCards)  
    {  
        Collection = postCards;  
    }  
    public MyBigCollection Collection { get; private set; }  
}  
public class MyBigCollection : List<int>  
{}
```

WPF Collection Binding. How to find?

WPF collection binding leak: MyBigCollection

Objects of the MyBigCollection type that may cause the WPF collection binding Objects: 163 Total size: 4,46 KB

Type List:

- Plain List
- Group by Namespace
- Group by Assembly
- Group by Interface
- Group by Dominators**
- Group by Similar Retention
- Instances
- Group by Creation Stack Trace:
- Call Tree
- Call Tree as Icicle Chart
- Back Traces
- Group by Generations
- Group by Shortest Path

Here you can view who dominates the object set. The A is a dominator of B if it exclusively retains the latter in memory. If A is garbage collected, B is also garbage collected. [Learn more](#)

Type	Objects	Bytes
MS.Internal.Data.DataBindEngine	163	4 564
MS.Internal.Data.ViewManager	163	4 564
System.Collections.Specialized.HybridDict	163	4 564
System.Collections.Hashtable	163	4 564
System.Collections.Hashtable+bucket[]	163	4 564
163 MS.Internal.Data.ViewTable	163	4 564
163 System.Collections.Specialized.ListD	163	4 564
163 System.Collections.Specialized.List	163	4 564
163 MS.Internal.Data.ViewRecord	163	4 564
163 System.Windows.Data.List Collec	163	4 564

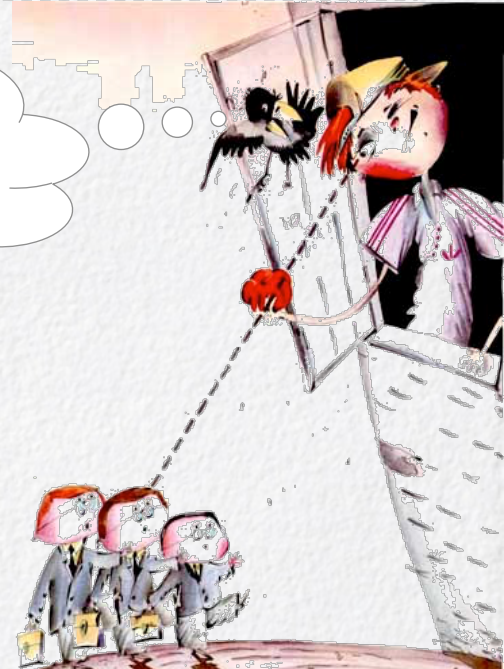
Collectio
n
Binding?



WPF Collection Binding. How to fix?

```
public class MyBigCollection : ObservableCollection<int>
{ }
```

О,
ObservableCollection
в хамі пошла

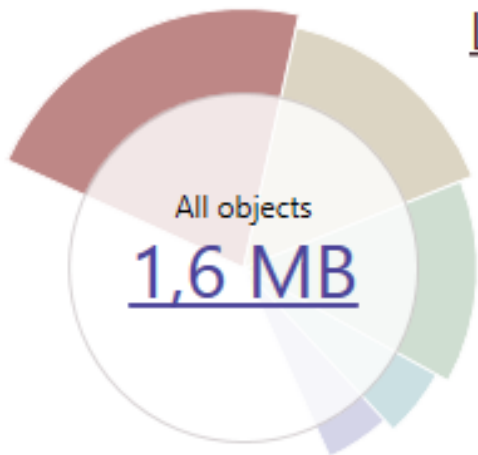


TextBox undo

```
<Grid>
```

```
  <TextBox HorizontalAlignment="Left" TextWrapping="Wrap"  
  AcceptsReturn="True"/>
```

```
</Grid>
```



Largest Size

348,32 KB [Char\[\]](#)

249,57 KB [TextBoxView+LineRecord](#)

227,04 KB [String](#)

81,81 KB [Object\[\]](#)

81,66 KB [Byte\[\]](#)

TextBox Undo. How to find?

Char[]

Objects of the type Char[]

Objects: 81

Total size: 348,32 KB

Type List:

Plain List

Group by Namespace

Group by Assembly

Group by Interface

Group by Dominators

Group by Similar Retention

Instances

Group by Creation Stack Trace:

Call Tree

Call Tree as Icicle Chart

Back Traces

Here you can view who dominates the object set. The A is a dominator of B if it exclusively retains the latter in memory. If A is garbage collected, B is also garbage collected. [Learn more](#)

Type	Objects
System.Windows.Documents.TextTreeNode	39
MS.Internal.Documents.UndoManager	8
System.Collections.Generic.List<IUndoUnit>	8
System.Windows.Controls.TextBoxView	8
System.Object[]	24
MS.Internal.Resources.ResourceManagerWrapper	1
System.Windows.ResourceDictionary	1

TextBox Undo. How to fix?

```
<Grid>
```

```
  <TextBox UndoLimit="10" HorizontalAlignment="Left"  
    TextWrapping="Wrap" AcceptsReturn="True"/>
```

```
</Grid>
```

В этом
TextBox'е Undo
не будет



x:Name leak

```
<controls:PersonEditorControl Grid.Row="0" x:Name="personEditor"/>
```

```
private void DeleteButton_OnClick(object sender, RoutedEventArgs args)
{
    if (personEditor != null)
    {
        myGrid.Children.Remove(personEditor);
        personEditor.Remove(personEditor);
        personEditor = null;
    }
}
```




Автоматические проверки

Event handlers leak

4,96 KB retained by [3 objects](#)

Objects subscribed to an event of another object and never unsubscribed from this event.

[Learn more](#)


 AdWindow
 TextServicesContext+TextServicesContextShutDownListe
 ManagedWndProcTracker+ManagedWndProcTrackerShu

WPF binding leak

864 B wasted by [72 objects](#)

	Size	Count
--	------	-------

Binding target objects that do not implement the `INotifyPropertyChanged` interface or do not use the `OneTime` binding mode. [Learn more](#)


 Person.Name	864 B	72
---	-------	----

WPF collection binding leak

310,90 MB retained by [163 objects](#)

	Size
--	------

Binding target objects that do not implement the `INotifyCollectionChanged` interface. [Learn more](#)


 MyBigCollection	310,90 MB
---	-----------


x:Name WPF leak

1,92 MB wasted by [1 objects](#)

	x:Name
--	--------

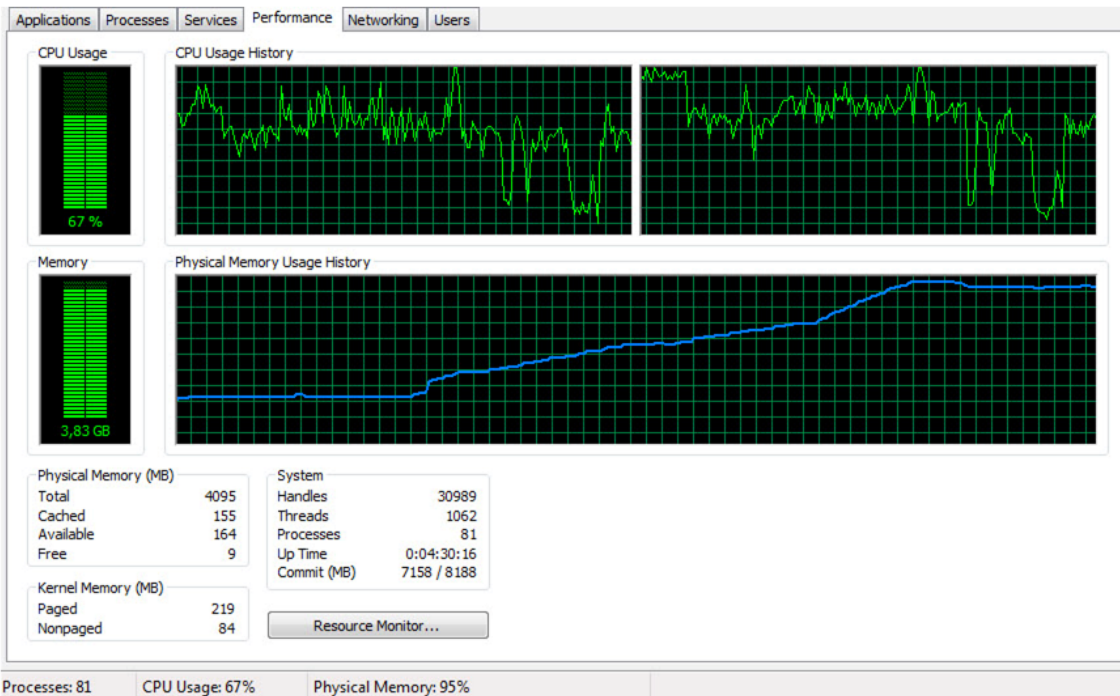
Dynamically-removed UI elements declared in XAML with the `x:Name` directive stay in memory until the `UnregisterName` method is called. [Learn more](#)

 PersonEditorControl	personEditor
---	--------------



Че у вас там?

Когда фиксить меморилик?

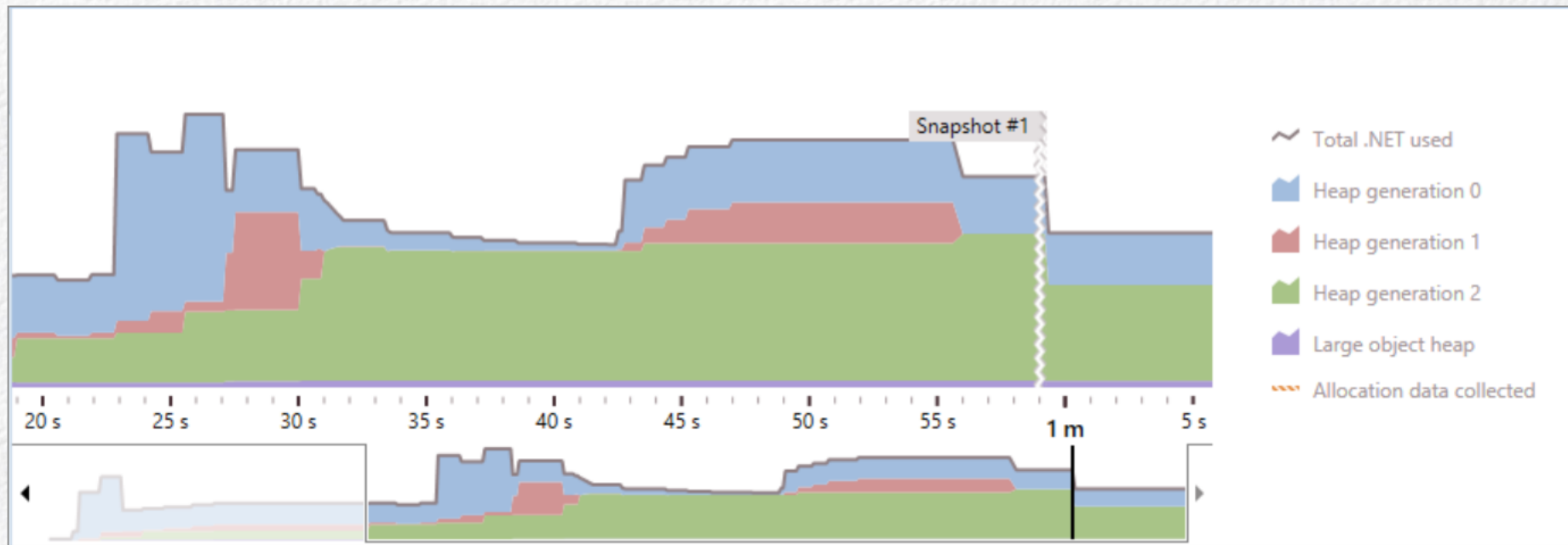


Шеф у нас память горит



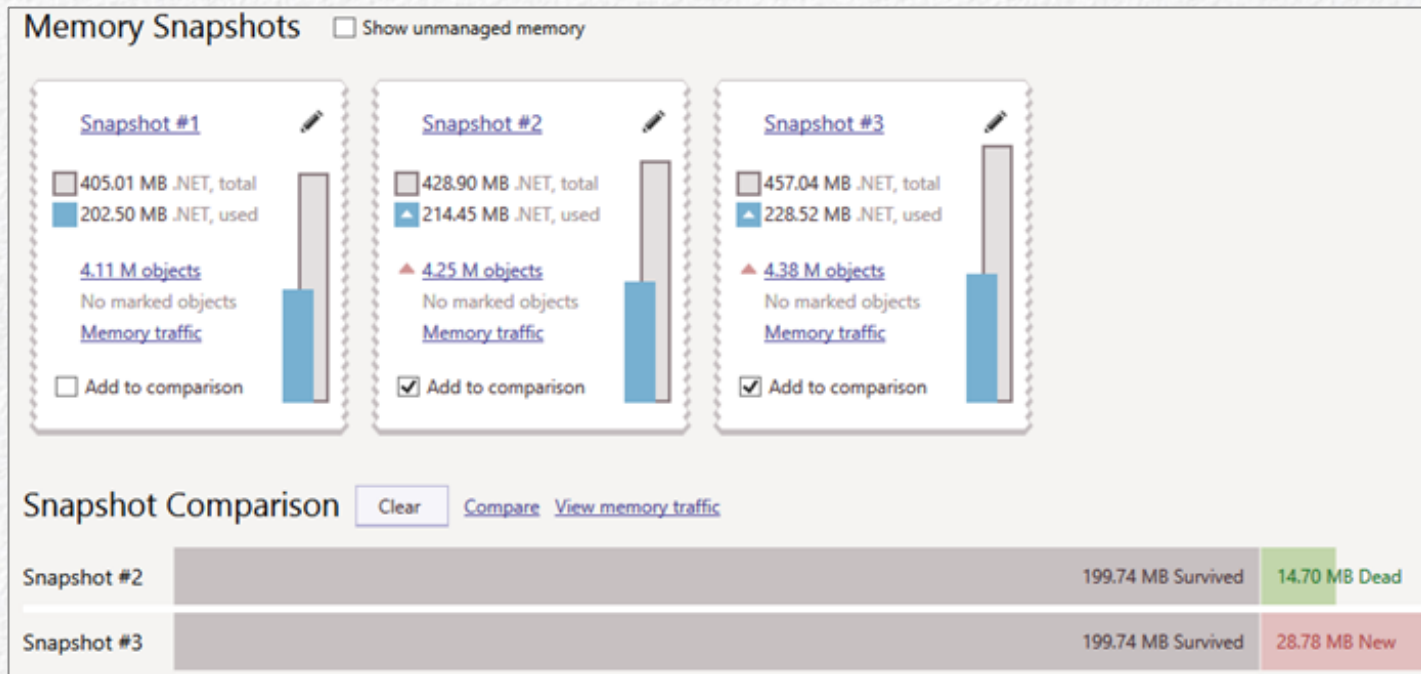
Как найти и пофиксить меморилик?

- Если есть подозрения



Как найти и пофиксить меморилик?

- Подозрений нет, а лик есть



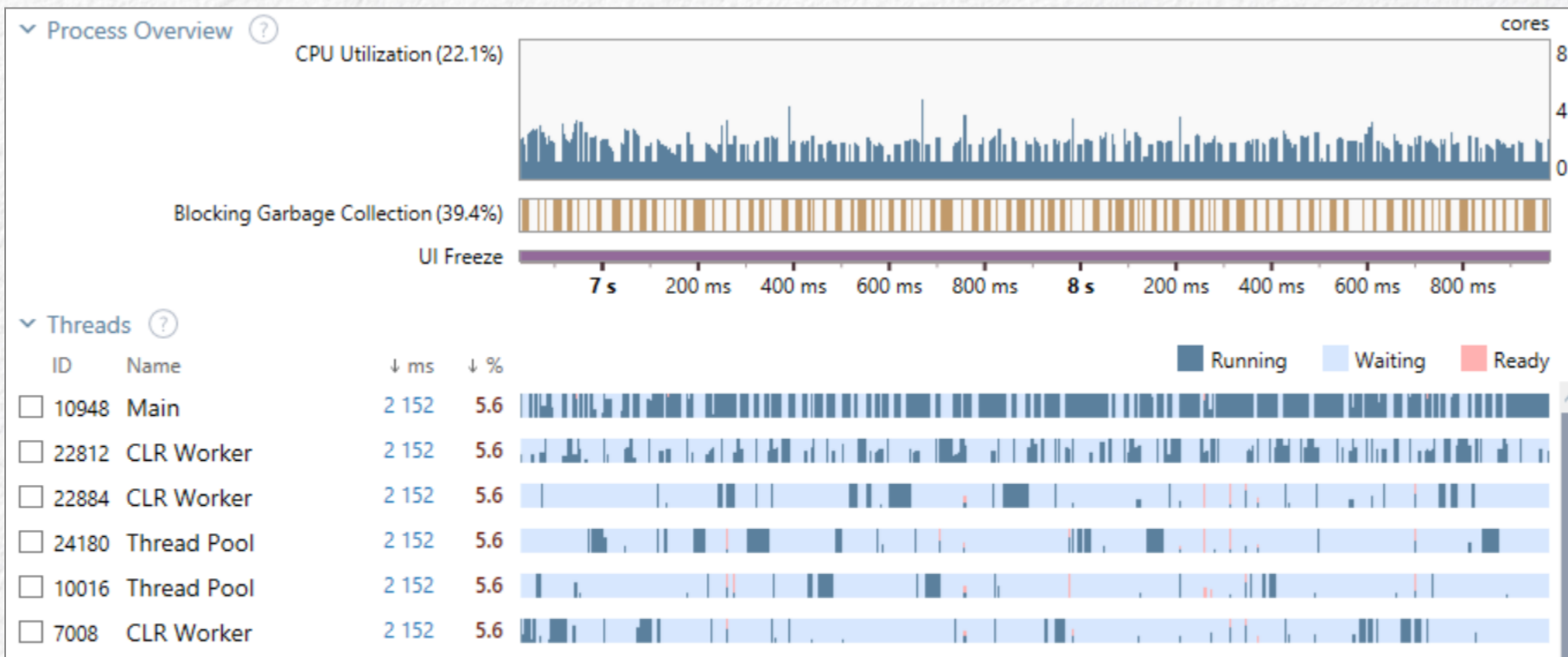
Chapter II.

Memory Traffic

Это не я,
это
траффик



А в чем собственно проблема?



Два потока

Мы будем
вызывать
сборку
мусора



Нет мы будем
вызывать
сборку
мусора

Strings are immutable?

```
static void ReverseString(string str)
{
    var i = 0;
    var j = str.Length - 1;
    while (i < j)
    {
        var temp = str[j];
        str[j--] = str[i];
        str[i++] = temp;
    }
}
```

Что значит, «даже
не
скомпилируется»?



Strings are immutable?

```
var str = "BELOW!";  
str = str.Replace("w", "v");  
Console.WriteLine(str);
```

Output:

Below!

Strings are immutable?

```
private static unsafe void ReverseString(string str)
{
    var i = 0;
    var j = str.Length - 1;
    fixed (char* fstr = str)
    {
        while (i < j)
        {
            char temp = fstr[j];
            fstr[j--] = fstr[i];
            fstr[i++] = temp;
        }
    }
}
```

Strings are immutable?

```
private static void ReverseString(string str)
{
    var i = 0;
    var j = str.Length - 1;
    var setter = typeof(string).GetMethod("SetChar",
        BindingFlags.Instance | BindingFlags.Public);
    while (i < j)
    {
        char temp = str[j];
        setter.Invoke(str, new object[] {j--, str[i]});
        setter.Invoke(str, new object[] {i++, temp});
    }
}
```

У нас же есть
рефлекшн!
Хе-хе



Strings are immutable?

```
Console.Write("What's your name? ");  
var name = Console.ReadLine();  
var reverse = name;  
ReverseString(reverse);  
Console.WriteLine("Your name is {0}, which spelled " +  
                  "backwards is {1}.", name, reverse);
```

Your name is AMOR, which spelled backwards is AMOR.

Strings are immutable?

```
var x = "Hello";  
var y = "Hello";  
ReverseString(x);  
Console.WriteLine(y);
```

olleH

Strings are immutable?

```
ReverseString("Traffic!");  
Console.WriteLine("Traffic!");
```

!ciffarT



String traffic.

```
static string Reverse(string line)
{
    var result = String.Empty;
    for (var i = line.Length - 1; i >= 0; i--)
        result += line[i];

    return result;
}
```

String traffic. How to find?

Allocated: [10,008,138,940 bytes; 300,323 objects] Collected: [10,007,906,686 bytes; 300,097 objects]







Plain List






Group by Interface

Group by Namespace

Group by Assembly

Filter:

Type	Allocated bytes	Allocated objects
 System.String	10,005,706,934	200,128
 System.Char	2,400,000	100,000
 System.Object[]	17,944	12
 System.Collections.Hashtable+bucket[]	2,016	7
 System.Char[]	1,726	24
 System.Int32[]	1,324	9

Function	Allocated bytes	Allocated objects
 String.Concat(String str0, String str1)	10,002,699,972	99,999
 Program.Reverse(String line)	10,002,699,972	99,999
 String.CtorCharCount(Char c, Int32 count)	3,000,026	100,001
 String.CtorCharPtrStartLength(Char ptr, Int32 startIndex, Int32 length)	1,628	32
 String.GetStringForStringBuilder(String value, Int32 startIndex, Int32 length)	1,244	12

String.Concat,
я этого не
заказывал!



String traffic. How to find?

```
for (int i = charArray.Length; i > 0; i--)  
{  
    stringResult += charArray[i - 1];  
}  
return stringResult;
```

Object allocation: string concatenation

String traffic. How to fix?

```
static string Reverse(string line)
{
    var sb = new StringBuilder();
    for (var i = line.Length - 1; i >= 0; i--)
        sb.Append(line[i]);

    return sb.ToString();
}
```

Parameters array

```
void LogMessage(string message, params object[] args)
{...}
```

```
LogMessage("WTF?");
```

Object allocation: parameters array 'args' creation

```
LogMessage("WTF?", new object[] {});
```

Parameters array. How to fix?

```
void LogMessage(string message) {...}  
void LogMessage(string message, params object[] args) {...}  
void LogMessage(string message, object arg0) {...}  
void LogMessage(string message, object arg0, object arg1) {...}
```

А у меня есть
5 реализаций



Блин, и еще 5

Boxing

```
double d = 0;  
object o = d; // Boxing takes place
```






Boxing allocation: conversion from value type 'double' to reference type 'object'

Allocated: [818,250,094 bytes; 7,064,747 objects] Collected: [725,876,058 bytes; 6,494,688 objects]

Plain List Group by Interface Group by Namespace Group by Assembly

Filter:

Clear

Type	Allocated bytes	Allocated objects	Collected bytes	Collected objects
 System.Collections.Specialized.NotifyC...	930,880	14,545	636,864	9,951
 System.Double	925,344	38,556	878,544	32,606
 JetBrains.dotTrace.SnapShotApi.Signat...	892,976	6,566	890,256	6,566
 System.Text.StringBuilder	892,896	18,602	892,368	18,602
 MS.Internal.TextFormatting.TextMetrics	857,024	3,826	857,024	3,826

Boxing, ты кто?



Boxing

How to fix?

```
String.Format("i = {0}", i);
```

Boxing allocation: conversion from value type 'int' to reference type 'object'

```
String.Format("i = {0}", i.ToString());
```

Argument conversion

```
void LogMessage(string message, object arg0) {...}
```

```
LogMessage("Wrong value", 123);
```

```
void LogMessage<T>(string message, T arg0) {...}
```

Collection resize

```
list.Add(item);
```

Ой, и тут
трафик?



Collection resize. How to find?

Allocated: [818,250,094 bytes; 7,064,747 objects] Collected: [725,876,058 bytes; 6,494,688 objects]

Plain List Group by Interface Group by Namespace Group by Assembly

Filter: Clear

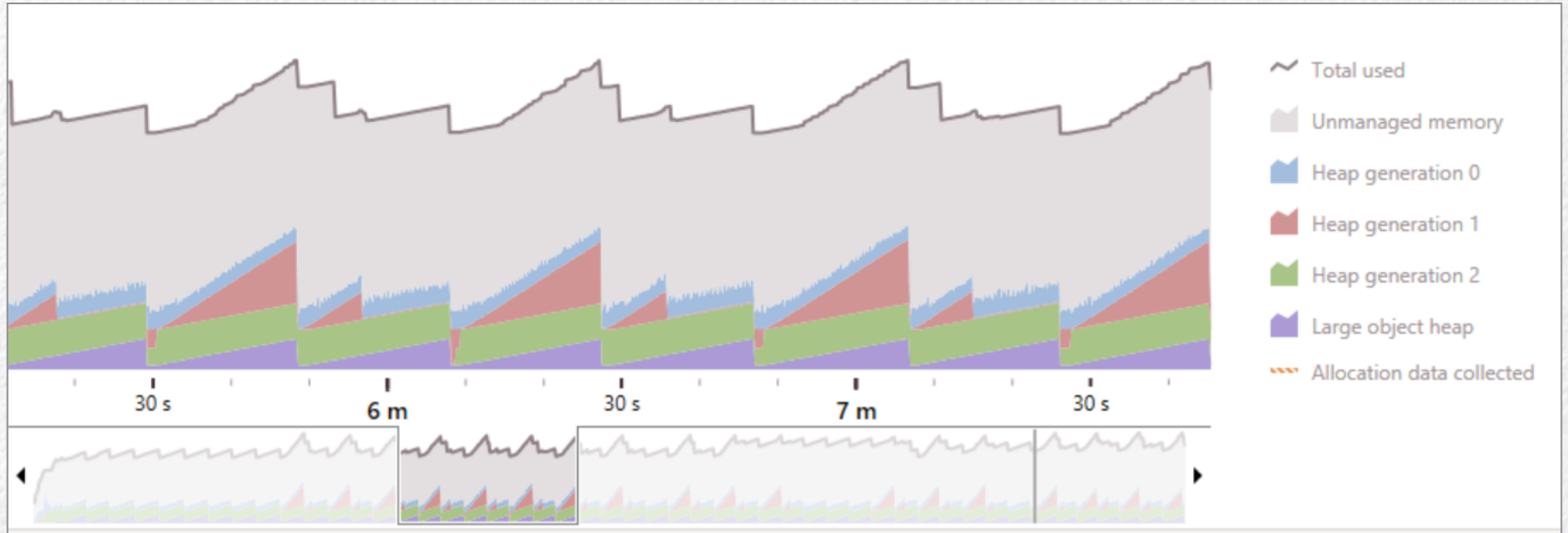
Type	Allocated bytes	Allocated objects	Collected bytes	Collected objects
System.Int32[]	22,744,352	163,362	21,330,552	154,925
System.Char[]	12,446,172	138,293	12,382,540	137,760
System.Collections.Generic.List+Enumerator<Na	10,817,360	270,434	10,817,360	270,434
System.Reflection.RuntimeMethodInfo[]	7,509,096	88,570	7,489,576	88,388

Function	Allocated bytes	Allocated objects	Collected bytes	Collected objects
StringBuilder.ctor(String value, Int32 startIndex, In	1,505,012	11,859	1,504,810	11,856
StringBuilder.ExpandByABlock(Int32 minBlockCha	1,392,638	6,392	1,392,548	6,391
Uri.GetLocalPath()	681,424	3,316	681,424	3,316
XmlTextReaderImpl.InitStreamInput(Uri baseUri, S	501,316	113	501,316	113
BinaryReader.ReadString()	458,354	1,629	431,564	1,534

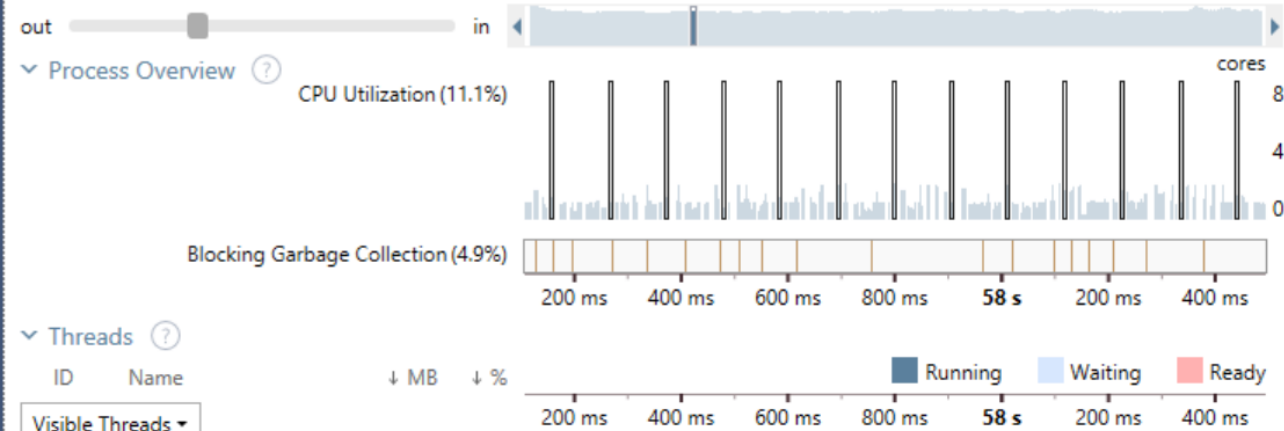
Collection resize. How to fix?

```
var list = new List<string>(10000);
```

LOH traffic



3 filters applied (clear all) : Visible 1 387 ms x Memory Allocation x Large Object Heap x



Filters X

Analysis Subject	clear
Time	24 972 ms
Memory Allocation	3.2 MB
Interval Filters	
Thread State	
Blocking GC	
Memory Allocation: Heap	clear
<input type="checkbox"/> Small Object Heap	MB % +187
<input checked="" type="checkbox"/> Large Object Heap	3.2

Top Methods Hide system methods

100 % Microsoft.Win32.RegistryKey.InternalGetValue(String, Object)

Call Tree Back Traces

- 100 % InternalGetValue • 3.2 MB • Microsoft.Win32.Regis
- 50.0 % get_NextValue • 1.6 MB • JetBrains.Common.Util.Per
- 50.0 % TryFillDataImpl • 1.6 MB • JetBrains.Common.Util.P
- 50.0 % TryFillData • 1.6 MB • JetBrains.Common.Util.Perf
- 50.0 % UpdateValue • 1.6 MB • JetBrains.Timeline.Frame
- 50.0 % OnSourceUpdated • 1.6 MB • JetBrains.Common.U
- 50.0 % UpdateValue • 1.6 MB • JetBrains.Common.U
- 50.0 % UpdateValue • 1.6 MB • JetBrains.Common.U
- 50.0 % RaiseChanged • 1.6 MB • JetBrains.Common.U
- 50.0 % UpdateValue • 1.6 MB • JetBrains.Common.U
- 50.0 % UpdateValue • 1.6 MB • JetBrains.Common.U
- 50.0 % UpdateValue • 1.6 MB • JetBrains.Common.U
- 50.0 % ThreadLogic • 1.6 MB • JetBrains.Time
- 50.0 % RunInternal • 1.6 MB • System.Threa
- <CreatePIDPerformanceCounter>b_0 • 1.6 MB • Jet

LOH traffic

How to fix?

- Anti-LOH collections
- Anti-LOH sorting



Collection enumeration

```
private void Enumerate(IEnumerable<string> stringList)
{
    foreach (var s in stringList)
    {
    }
}

public void Work()
{
    var list = new List<string>();

    for (var i = 0; i < 24000; i++)
        Enumerate(list);
}
```

Collection enumeration. How to find?

Allocated: [54,310 bytes; 1,306 objects] Collected: [25,764 bytes; 1,038 objects]

Plain List Group by Interface Group by Namespace Group by Assembly

Filter: Clear

Type	Allocated bytes	Allocated objects	Collected bytes	Collected objects
System.Collections.Generic.List+Enumerator<String>	24,000	1,000	24,000	1,000
System.Object[]	17,280	6	36	1
System.String	5,990	164	1,068	14
System.Globalization.CultureInfo	924	3		
System.Char[]	740	5	60	3





Function	Allocated bytes	Allocated objects	Collected bytes	Collected objects
List<T>.GetEnumerator()	24,000	1,000	24,000	1,000
EnumerableTest.Foo(IEnumerable<T> sList)	24,000	1,000	24,000	1,000
EnumerableTest.Goo()	24,000	1,000	24,000	1,000




Collection enumeration. How to find?

Allocated: [818,250,094 bytes; 7,064,747 objects] Collected: [725,876,058 bytes; 6,494,688 objects]

Plain List Group by Interface Group by Namespace Group by Assembly

Filter: Clear

Type	Allocated bytes	Allocated objects	Collected bytes	Collected objects
 System.SZArrayHelper+SZGenericArrayEnumerator<IRuleInfo>	3,041,952	95,061	3,041,952	95,061
 JetBrains.Application.Configuration.XmlExternalizableAttribute[]	3,037,768	84,284	3,037,768	84,284
 JetBrains.Util.Pair<MethodViewData, UInt64>[]	2,826,024	12,604	2,753,880	12,490
 System.Collections.Generic.List<Attribute>	2,816,120	70,403	2,816,120	70,403

Function	Allocated bytes	Allocated objects	Collected bytes	Collected objects
 SZArrayHelper.GetEnumerator<T>()	3,041,952	95,061	3,041,952	95,061
 SubsystemResolveHelpers.CheckFunctionSignature(ISubsystemIn	3,041,344	95,042	3,041,344	95,042
 SubsystemSettingsSerializationHelper.BuildSubsystemXml(XmlEl	608	19	608	19

Collection enumeration. How to fix?


```
private void Enumerate(List<string> stringList)
{
    foreach (var s in stringList)
    {
    }
}
```

Когда фиксировать трафик?

- Когда он реально большой!
- Влияет на отзывчивость интерфейса

Chapter III.

Высокая статической память



Что-то у
нас
мусора
много

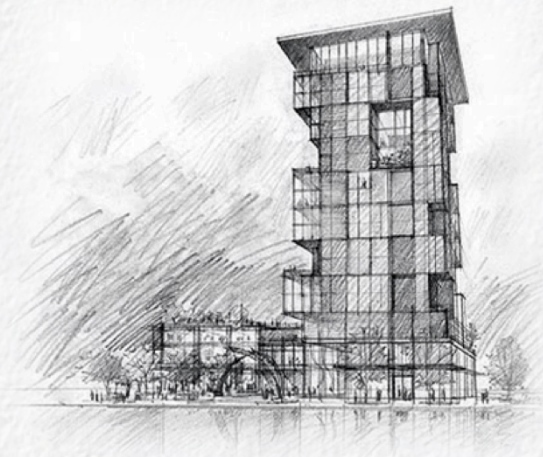


Причины

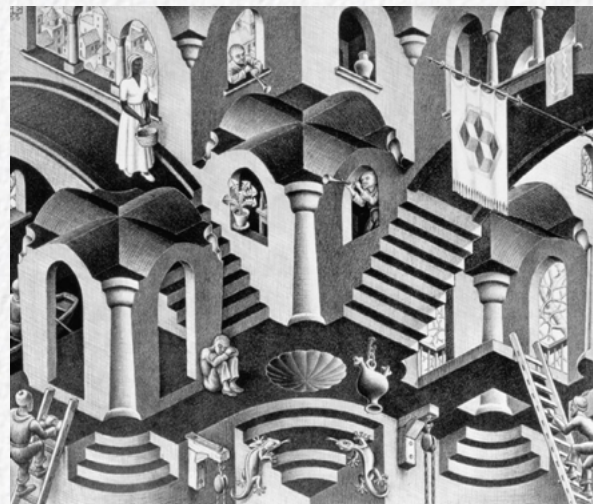
- Алгоритмические
- Фрагментация
- Дублирование

Архитектура

Ожидание



Реальность



Группировки

Type List:

Plain List

Group by Namespace

Group by Assembly

Group by Interface

Group by Dominators

Group by Similar Retention

Instances

Group by Creation Stack Trace:

Call Tree

Call Tree as Icicle Chart

Back Traces

Group by Generations

Group by Shortest Path



Fragmentation

Heap Fragmentation

Large Object Heap 3 heaps 37.49 MB total 70 % fragmentation 0 B pinned 33.06 MB used 4.43 MB free 823.12 KB CLR aux

Generation 2 50 heaps 722.71 MB total 88 % fragmentation 40.08 MB pinned 640.92 MB used 41.72 MB free 456 B CLR aux

Generation 1 1 heap 5.07 MB total 41 % fragmentation 36 B pinned 4.87 KB used 5.07 MB free 0 B CLR aux




Sparse Arrays











Sparse arrays

16.38 MB wasted by [843,713 objects](#)

Wasted

Fill

Partially filled arrays are inefficient from the point of memory usage. 

  <code>KeyValuePair<CompiledType, ValueRefEntry>[]</code>	2.11 MB	47.2 %
  <code>KeyValuePair<String, ValueRefEntry>[]</code>	376.11 KB	26.5 %
  <code>KeyValuePair<Dependency, JetHashSet<SourceFilePtr>>[]</code>	156.25 KB	0.0 %
  <code>KeyValuePair<String, FileSystemPath>[]</code>	143.19 KB	81.7 %
  <code>KeyValuePair<ITypeElement, ValueRefEntry>[]</code>	128.41 KB	49.8 %
<i>others</i>	13.48 MB	66.0 %

String duplicates

Duplicate strings from Snapshot #1

Wasted Size: 61.17 MB

Objects: 925,418

Filter:

Clear

List contains only first 10000 of 177959 entries.

String value	Length	Objects count	Wasted Size
Property.VsWindowFrame::IsVisibleRaw.Unguarded	46	22473	2.27 MB
Property.WindowFrame.SwitchToFrameRequested	43	22500	2.15 MB
Property.WindowFrame.IsVisibleOnScreen	38	22500	1.93 MB
Property.IsDisposedUnguarded	28	22558	1.51 MB
Property.WindowFrame.Closed	27	22500	1.46 MB
Property.RootParentHwnd	23	22500	1.29 MB
Property.UiElement	18	22500	1.07 MB
Microsoft.Expression.DesignHost.Isolation.Remoting.IRemoteObject, Microsoft.Expression.DesignHost, Version=12.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a	165	870	291.93 KB
.ctor	5	10985	257.44 KB

Questions & Answers

Profiling is fun[™]

Валим!

