# Learning .NET High-Performance Programming

## Chapter 1 - Performance Thoughts

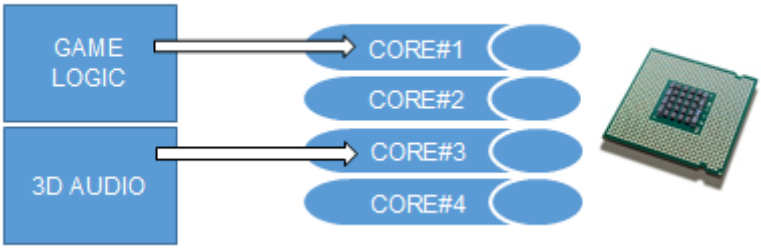
| Latency | Magnitude | | | | | |
|---|---|---|---|---|---|---|
| | 0% | 20% | 40% | 60% | 80% | 100% |
| Latency | | | | | | X |
| Throughput | | | X | | | |
| Resource usage | | | | | X | |
| Availability | | X | | | | |
| Scalability | X | | | | | |
| Efficiency | | | | X | | |

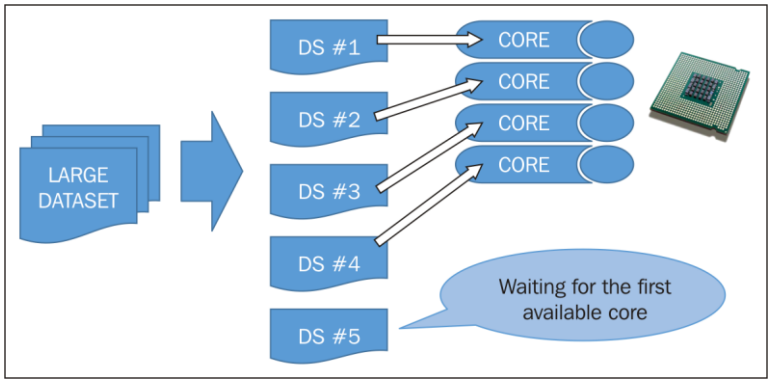

| Latency | Magnitude | | | | | |
|---|---|---|---|---|---|---|
| | 0% | 20% | 40% | 60% | 80% | 100% |
| Latency | | | | | | X |
| Throughput | | | X | | | |
| Resource usage | | | | | X | |
| Availability | | X | | | | |
| Scalability | X | | | | | |
| Efficiency | | | | X | | |

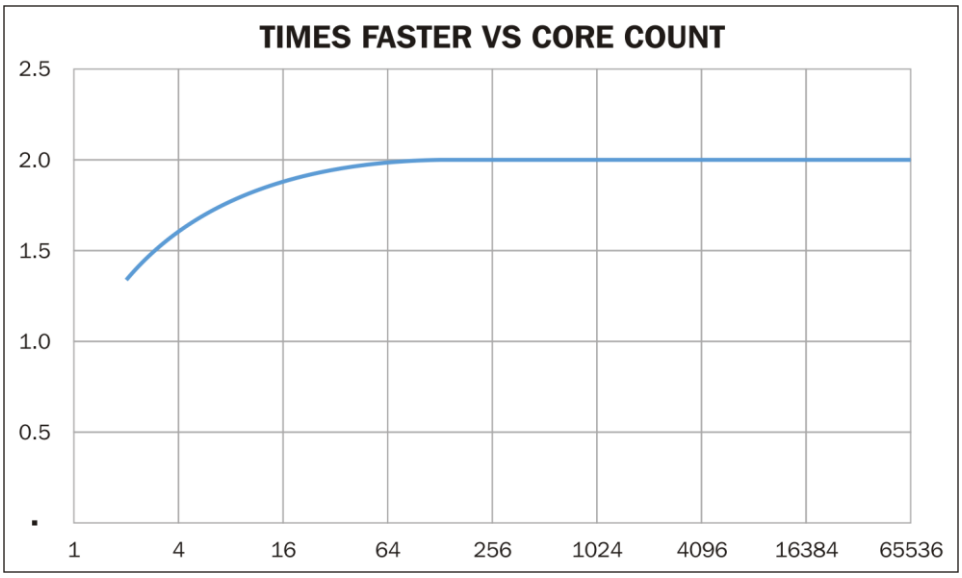

| Latency | Magnitude | | | | | |
|---|---|---|---|---|---|---|
| | 0% | 20% | 40% | 60% | 80% | 100% |
| Latency | | X | | | | |
| Throughput | | | | X | | |
| Resource usage | | | | | | X |
| Availability | | | X | | | |
| Scalability | X | | | | | |
| Efficiency | | | | X | | |

| Latency | Magnitude | | | | | |
|---|---|---|---|---|---|---|
| | 0% | 20% | 40% | 60% | 80% | 100% |
| Latency | X | | | | | |
| Throughput | | | | | | X |
| Resource usage | | | | | X | |
| Availability | | | X | | | |
| Scalability | | | | X | | |
| Efficiency | | X | | | | |

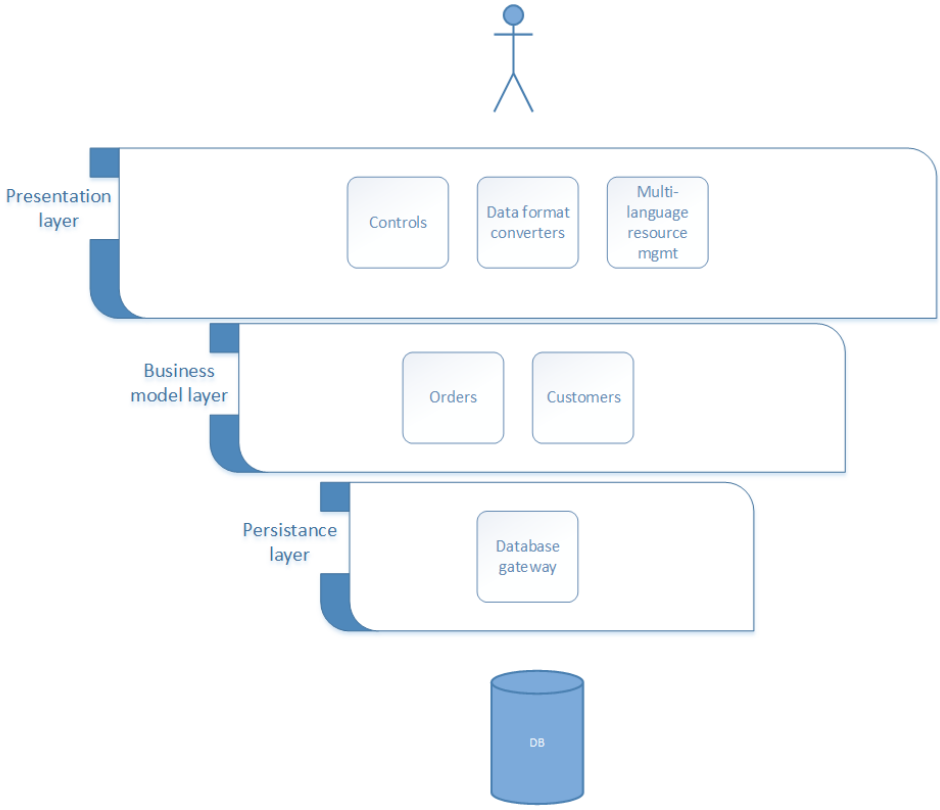
| Latency | Magnitude | | | | | |
|---|---|---|---|---|---|---|
| | 0% | 20% | 40% | 60% | 80% | 100% |
| Latency | | | | | | X |
| Throughput | | | | X | | |
| Resource usage | | | | | | X |
| Availability | | | X | | | |
| Scalability | | | | X | | |
| Efficiency | | X | | | | |

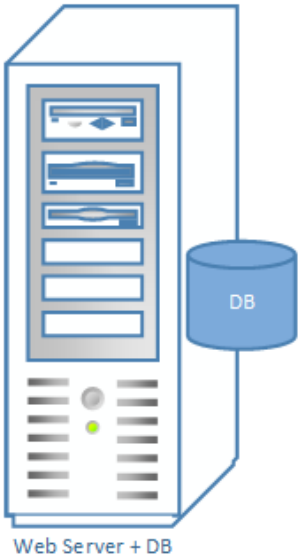


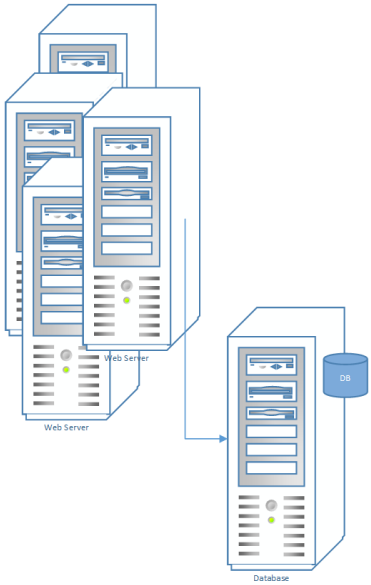



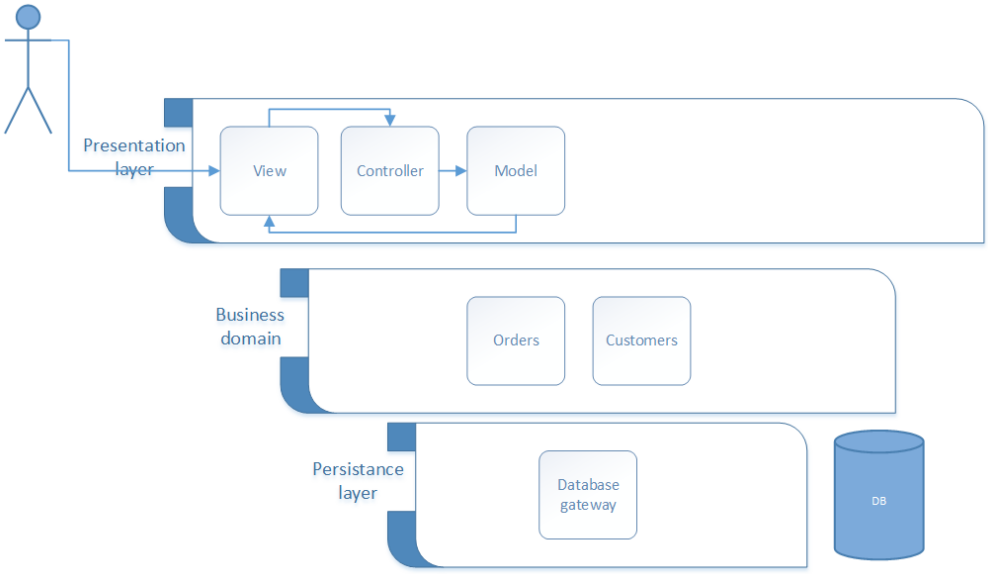

TIMES FASTER VS CORE COUNT
2.5
2.0
1.5
1.0
0.5
1
4
16
64
256
1024
4096
16384
65536

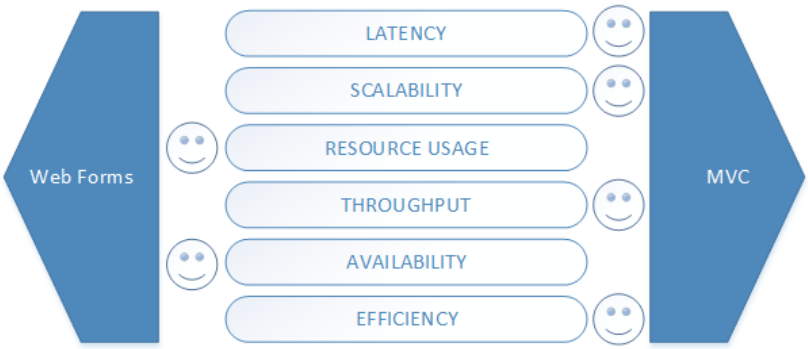
# Chapter 2 - Architecting High-performance .NET code

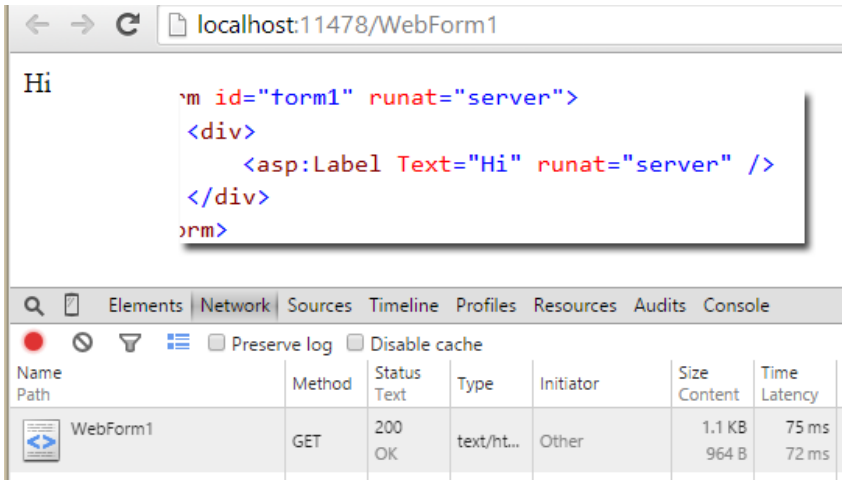



Web Server
Web Server
DB
Database

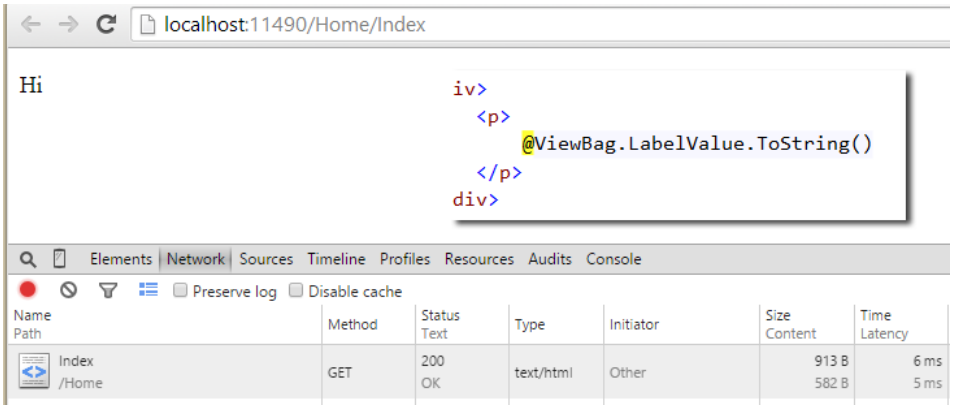

Presentation layer
View
Controller
Model
Business domain
Orders
Customers
Persistance layer
Database gateway
DB


Web Forms
LATENCY
SCALABILITY
RESOURCE USAGE
THROUGHPUT
AVAILABILITY
EFFICIENCY
MVC


localhost:11478/WebForm1
Hi
m id="form1" runat="server">

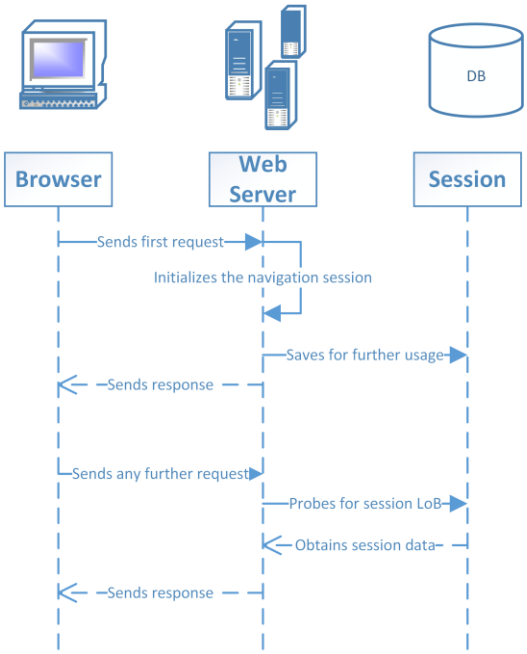
<asp:Label Text="Hi" runat="server" />

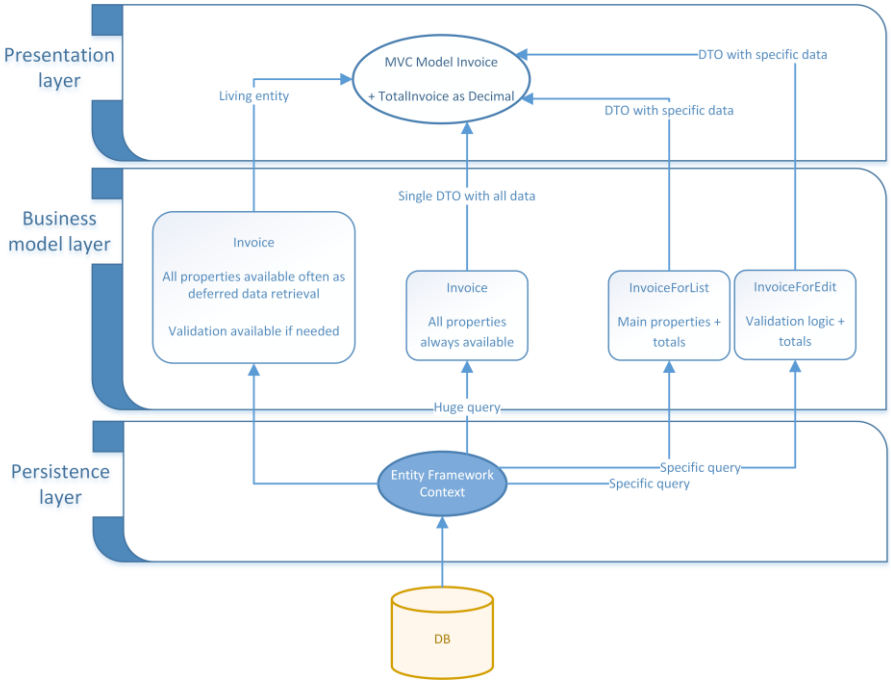
orm>
Elements Network Sources Timeline Profiles Resources Audits Console
Preserve log Disable cache
Name Path
Method
Status Text
Type
Initiator
Size Content
Time Latency
WebForm1
GET
200 OK
text/ht...
Other
1.1 KB 964 B
75 ms 72 ms

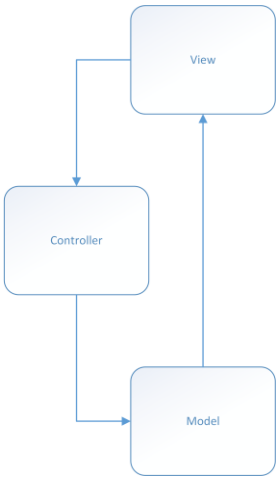

localhost:11490/Home/Index
Hi
iv>

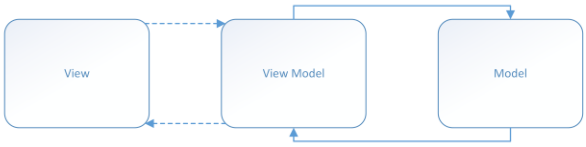
@ViewBag.LabelValue.ToString()

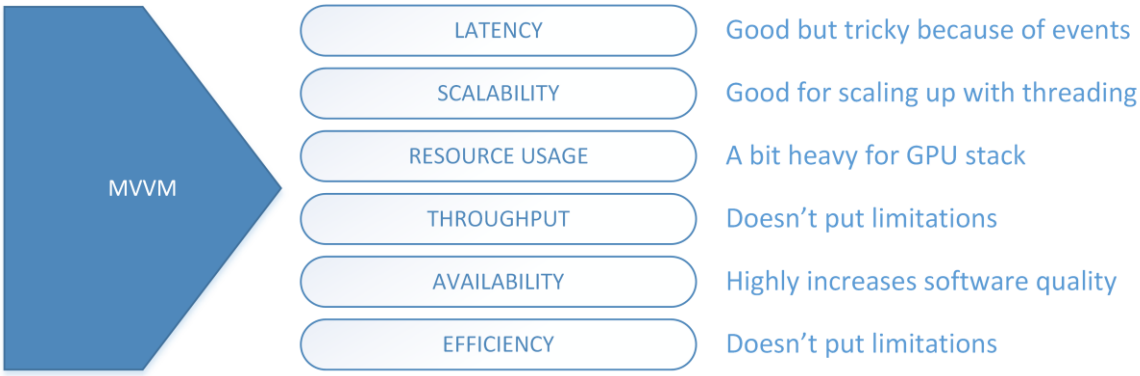
div>
Elements Network Sources Timeline Profiles Resources Audits Console
Preserve log Disable cache
Name Path
Method
Status Text
Type
Initiator
Size Content
Time Latency
Index /Home
GET
200 OK
text/html
Other
913 B 582 B
6 ms 5 ms

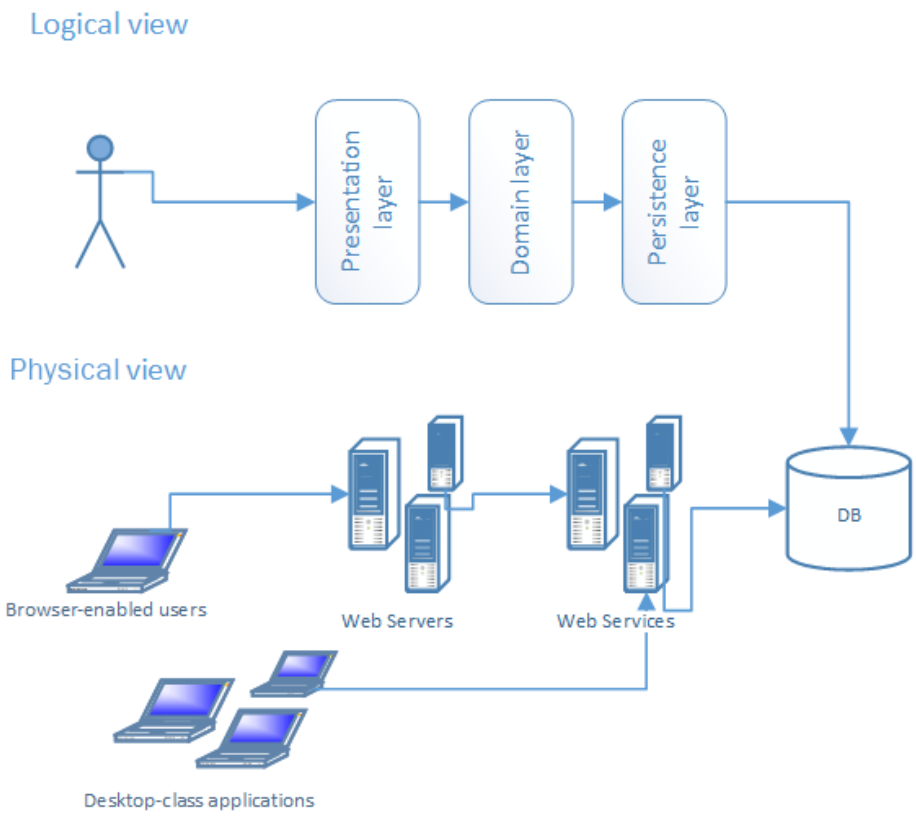

DB
Browser
Web Server
Session
Sends first request
Initializes the navigation session
Saves for further usage
Sends response
Sends any further request
Probes for session LoB
Obtains session data
Sends response

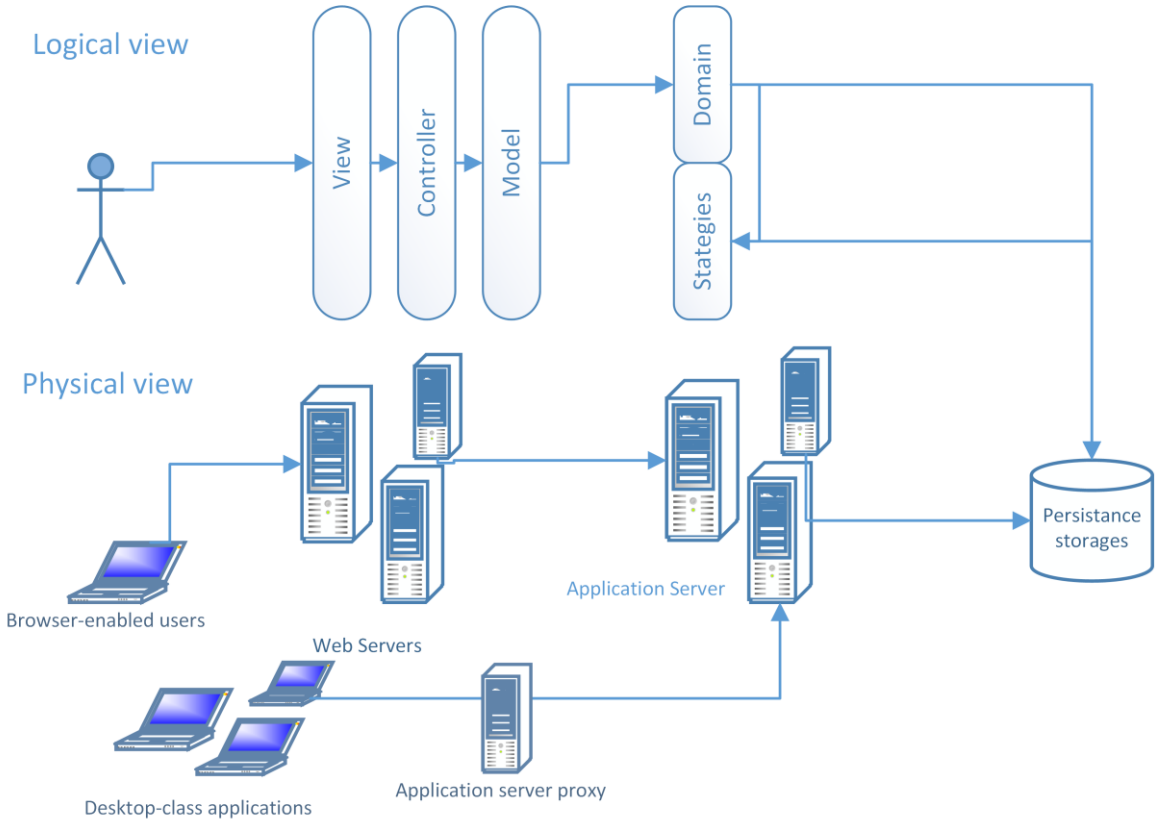

Presentation layer
MVC Model Invoice
+ TotalInvoice as Decimal
Living entity
DTO with specific data
DTO with specific data
Business model layer
Single DTO with all data
Invoice
All properties available often as deferred data retrieval
Validation available if needed
Invoice
All properties always available
InvoiceForList
Main properties + totals
InvoiceForEdit
Validation logic + totals
Huge query
Persistence layer
Entity Framework Context
Specific query
Specific query
DB

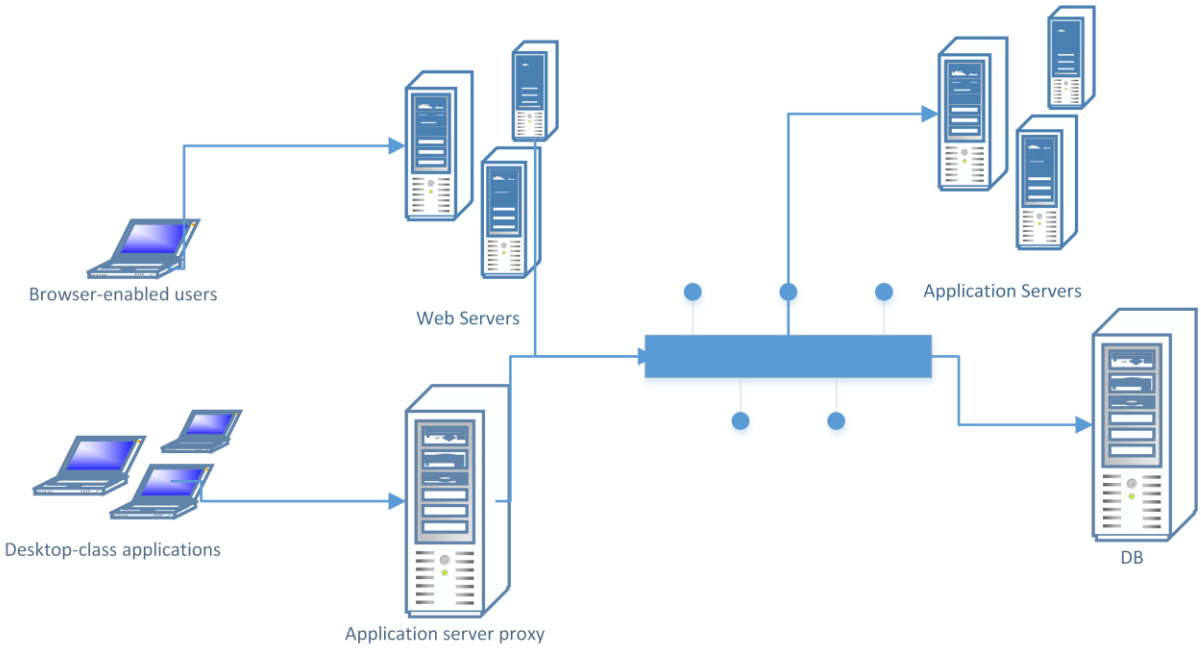

View
Controller
Model

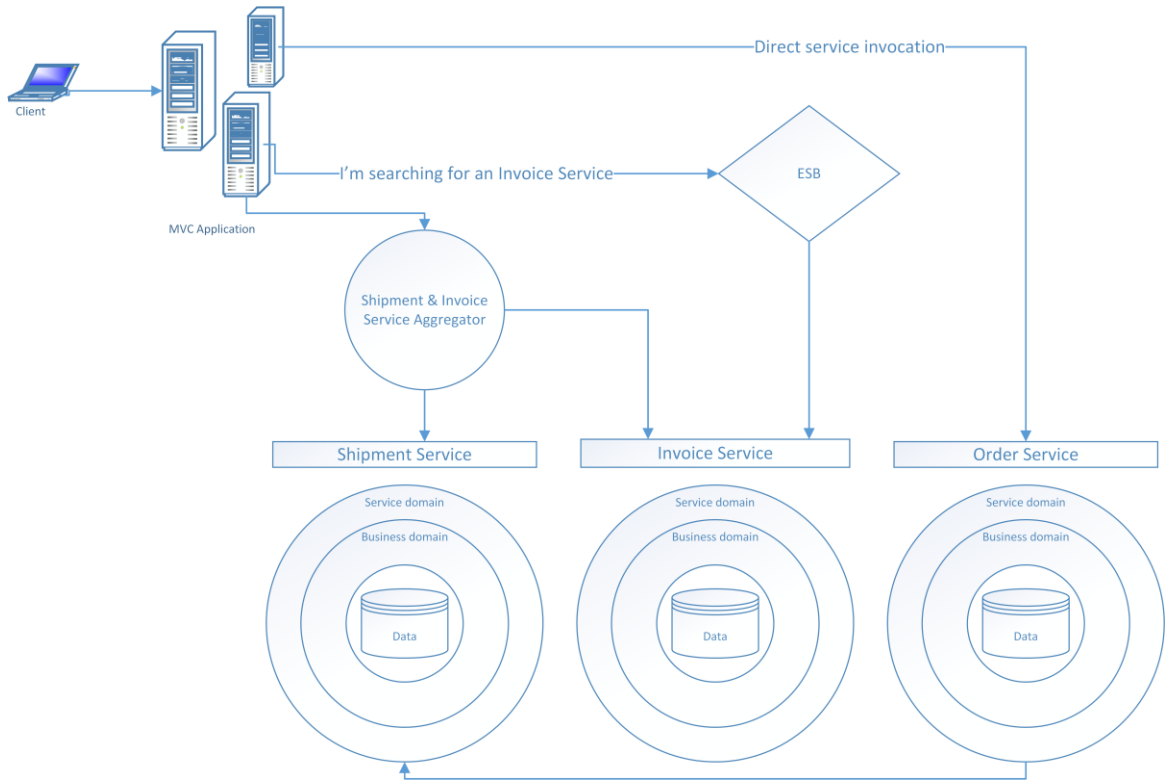

View
View Model
Model

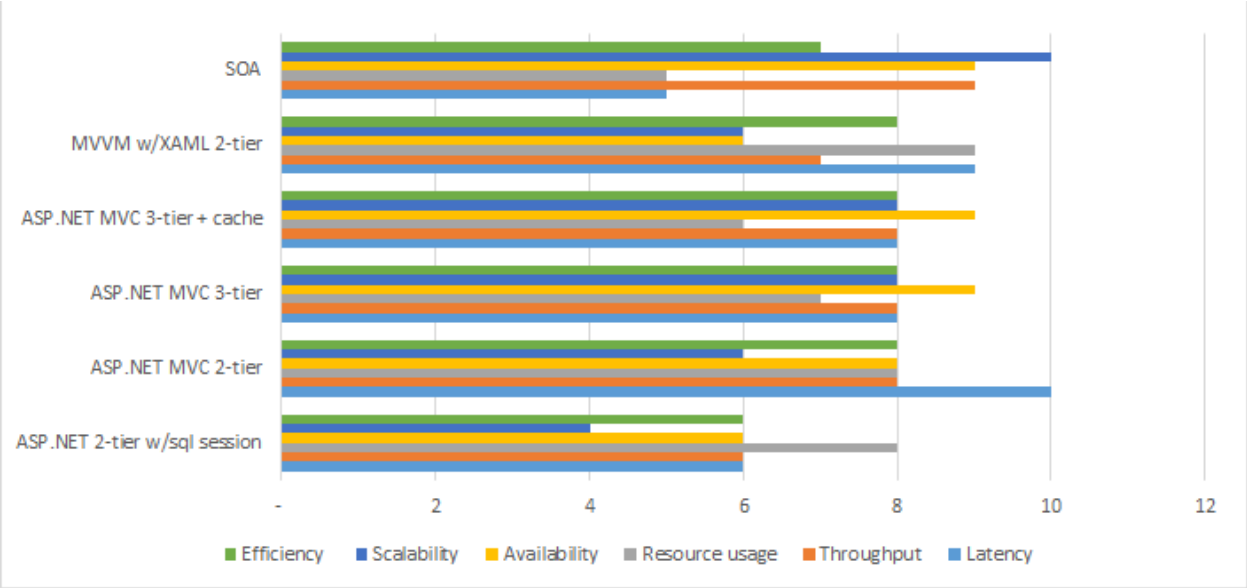

MVVM
LATENCY
Good but tricky because of events
SCALABILITY
Good for scaling up with threading
RESOURCE USAGE
A bit heavy for GPU stack
THROUGHPUT
Doesn't put limitations
AVAILABILITY
Highly increases software quality
EFFICIENCY
Doesn't put limitations

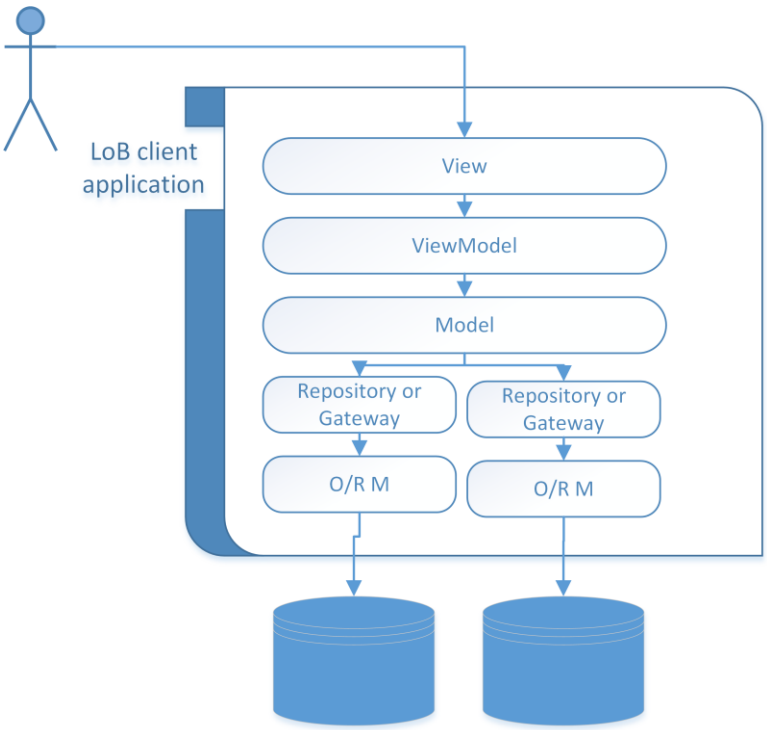

Logical view
Presentation layer
Domain layer
Persistence layer
Physical view
DB
Browser-enabled users
Web Servers
Web Services
Desktop-class applications

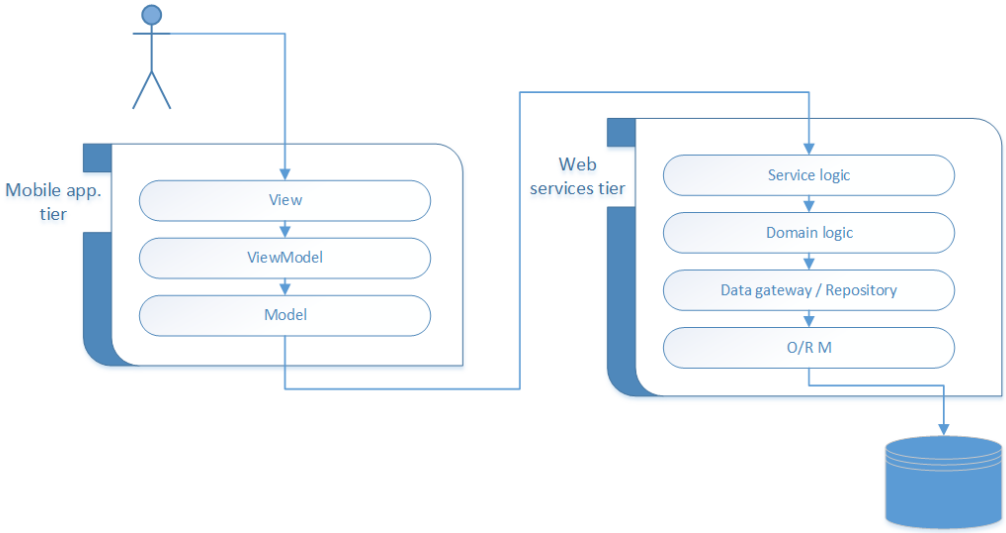

Logical view
View
Controller
Model
Domain
Stategies
Physical view
Browser-enabled users
Web Servers
Application Server
Persistance storages
Desktop-class applications
Application server proxy

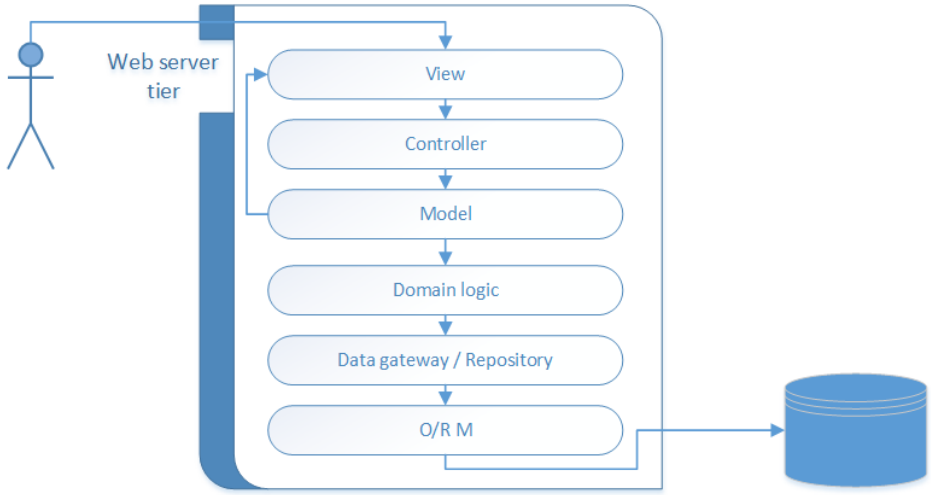

Browser-enabled users
Web Servers
Application Servers
Desktop-class applications
Application server proxy
DB

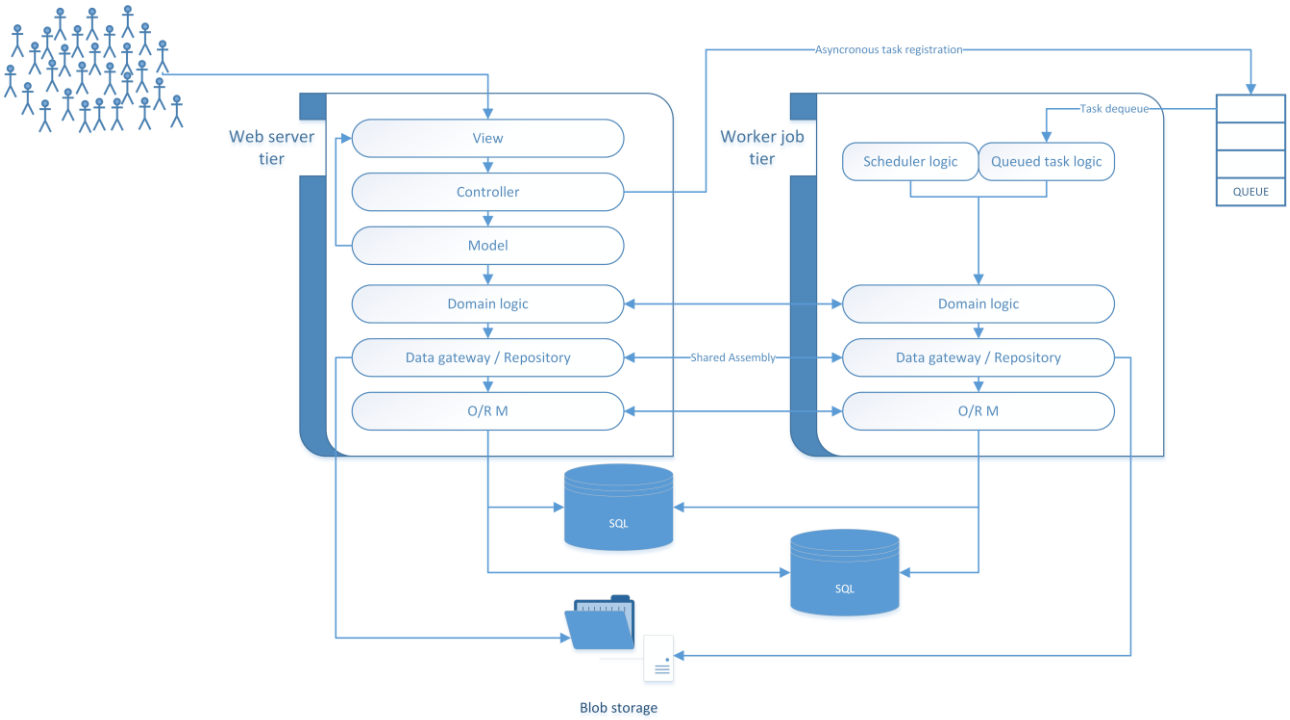

Client
MVC Application
Direct service invocation
I'm searching for an Invoice Service
ESB
Shipment & Invoice Service Aggregator
Shipment Service
Invoice Service
Order Service
Service domain
Business domain
Data
Service domain
Business domain
Data
Service domain
Business domain
Data

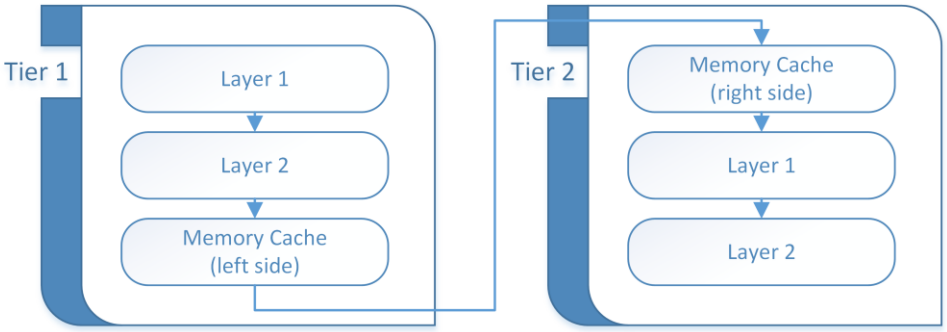

SOA
MVVM w/XAML 2-tier
ASP.NET MVC 3-tier + cache
ASP.NET MVC 3-tier
ASP.NET MVC 2-tier
ASP.NET 2-tier w/sql session
-
2
4
6
8
10
12
Efficiency
Scalability
Availability
Resource usage
Throughput
Latency


LoB client application
View
ViewModel
Model
Repository or Gateway
Repository or Gateway
O/R M
O/R M

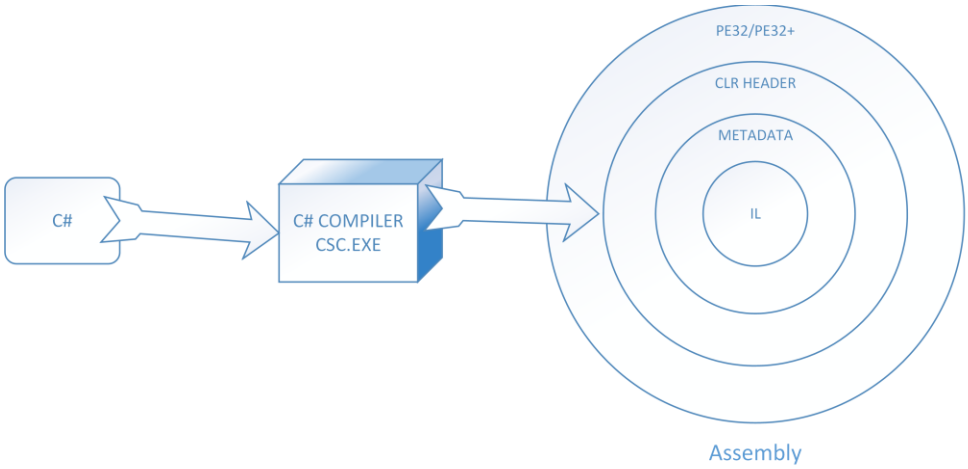

Mobile app. tier
View
ViewModel
Model
Web services tier
Service logic
Domain logic
Data gateway / Repository
O/R M

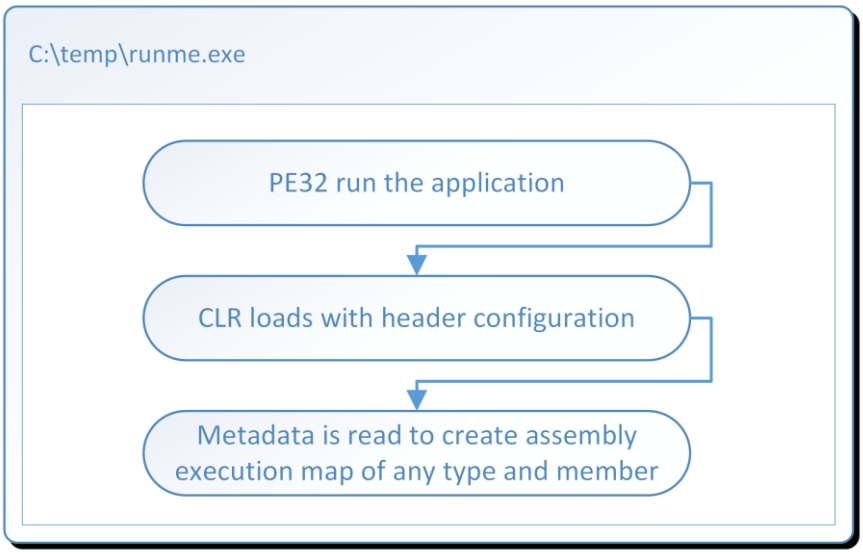

Web server tier
View
Controller
Model
Domain logic
Data gateway / Repository
O/R M

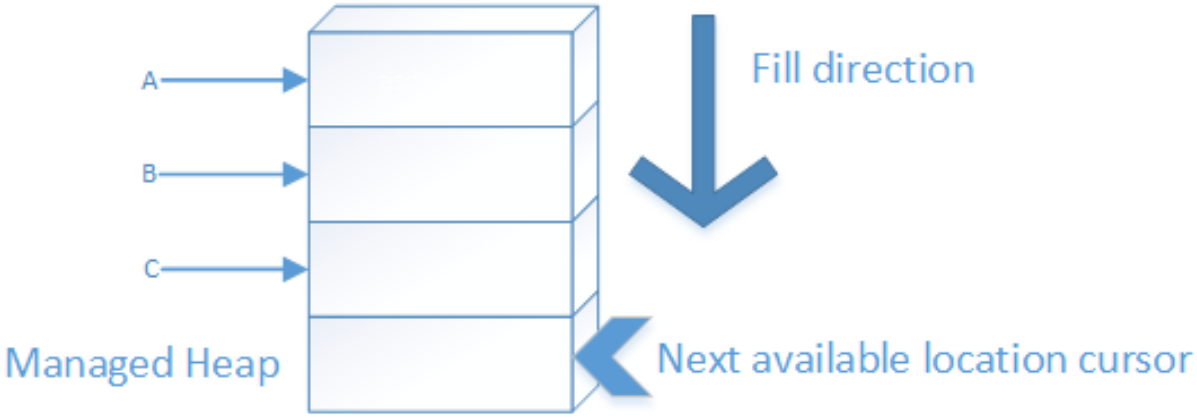

Asyncronous task registration
Web server tier
View
Controller
Model
Domain logic
Data gateway / Repository
O/R M
Worker job tier
Task dequeue
Scheduler logic
Queued task logic
QUEUE
Domain logic
Shared Assembly
Data gateway / Repository
O/R M
SQL
SQL
Blob storage


Tier 1
Layer 1
Layer 2
Memory Cache
(left side)
Tier 2
Memory Cache
(right side)
Layer 1
Layer 2

# Chapter 3 - CLR Internals

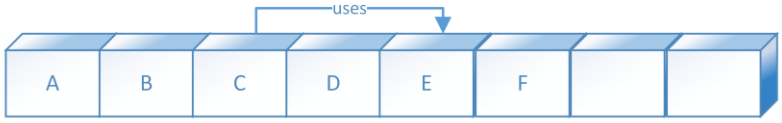




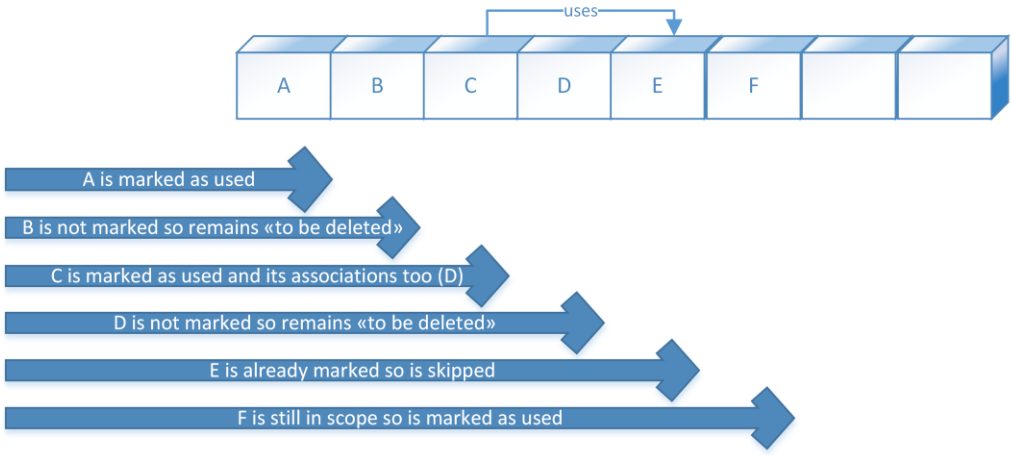

| RAM | Microsoft Windows (32\|64 bit) | | Application Address Space (32\|64 bit) | |
|---|---|---|---|---|
| 4GB | 3.25GB | 4GB | 1.5GB | 4GB |
| 8GB | 3.25GB | 8GB | 1.5GB | 8GB |
| 64GB | 3.25GB | 64GB | 1.5GB | 64GB |

1) Variables are instantiated

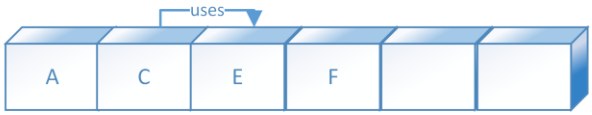



2) B and D become unreachable because their scope end

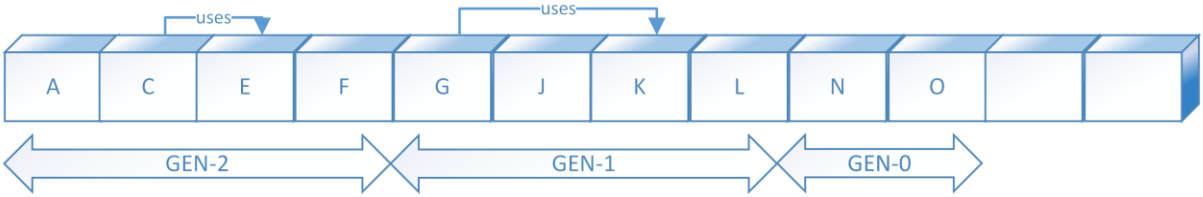
3) GC fires and start marking any object if still used (in FIFO fashion), elsewere the object will be deallocated from memory

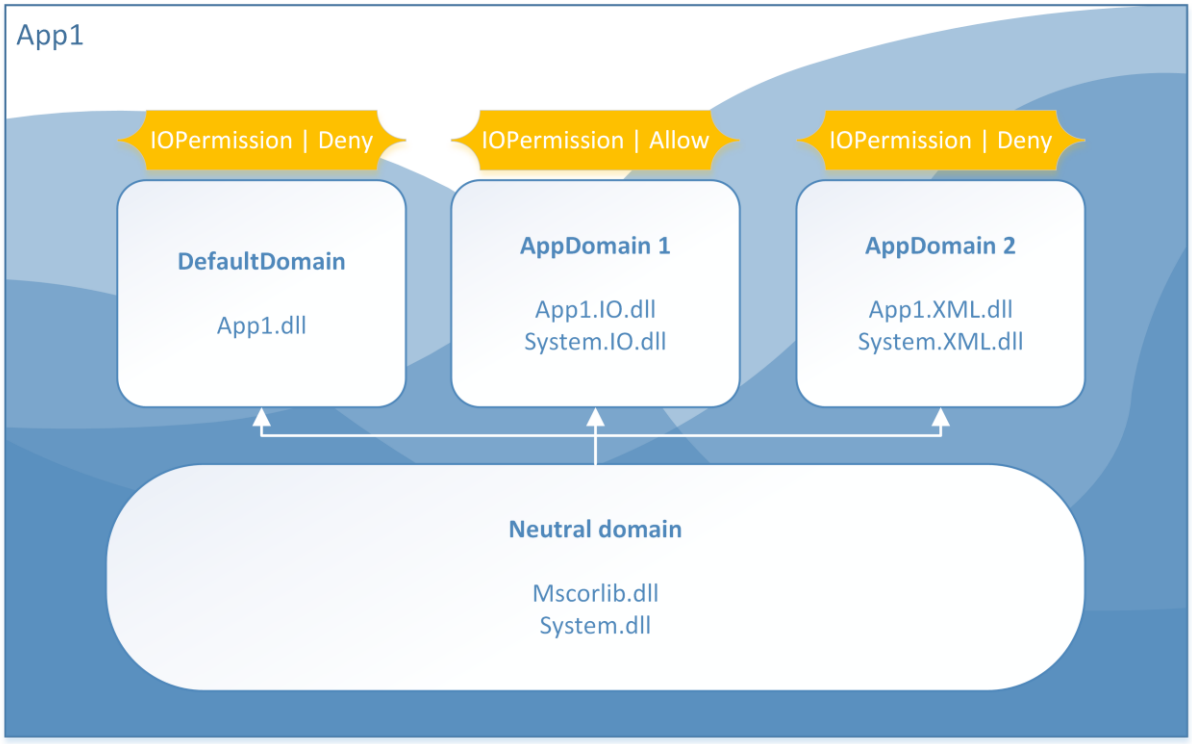


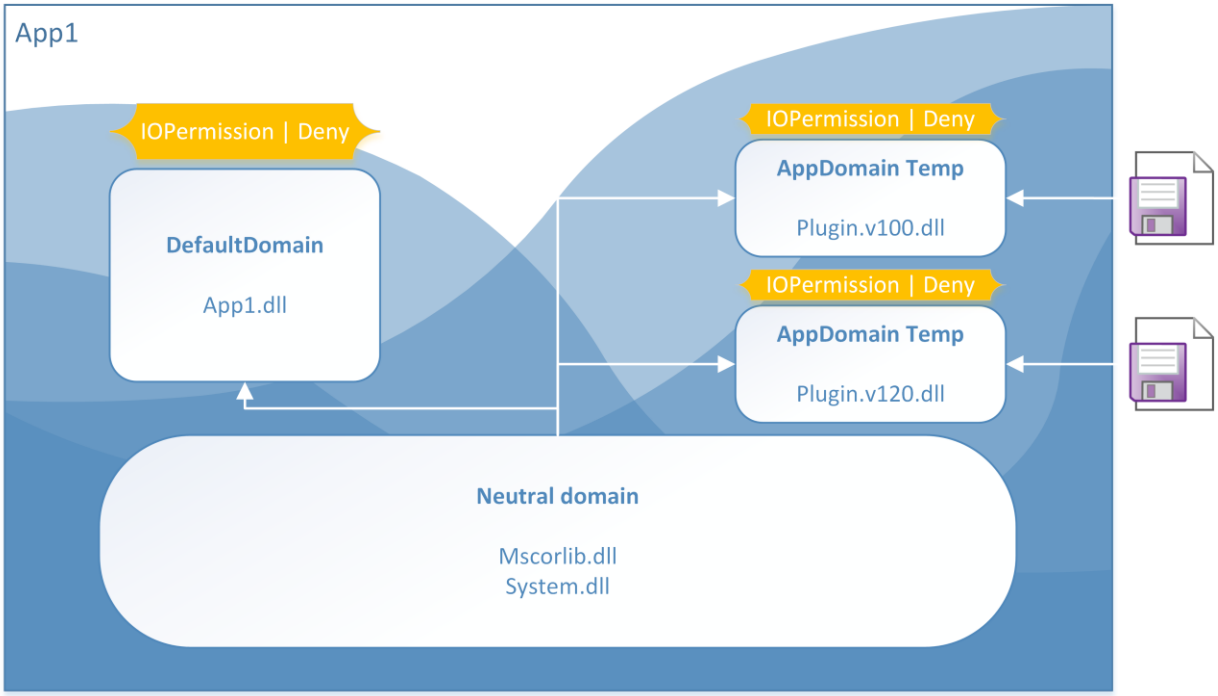

4) the GC ended Marking phase start the Compacting phase

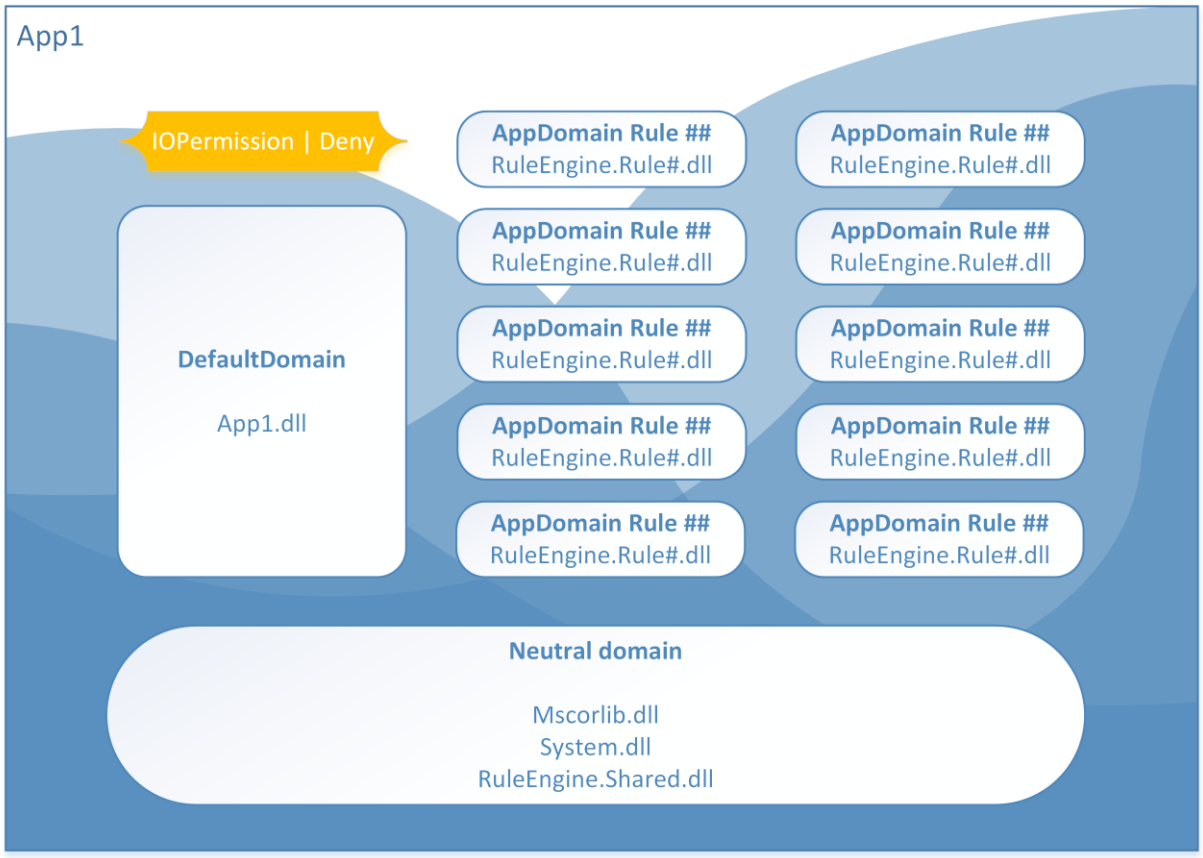


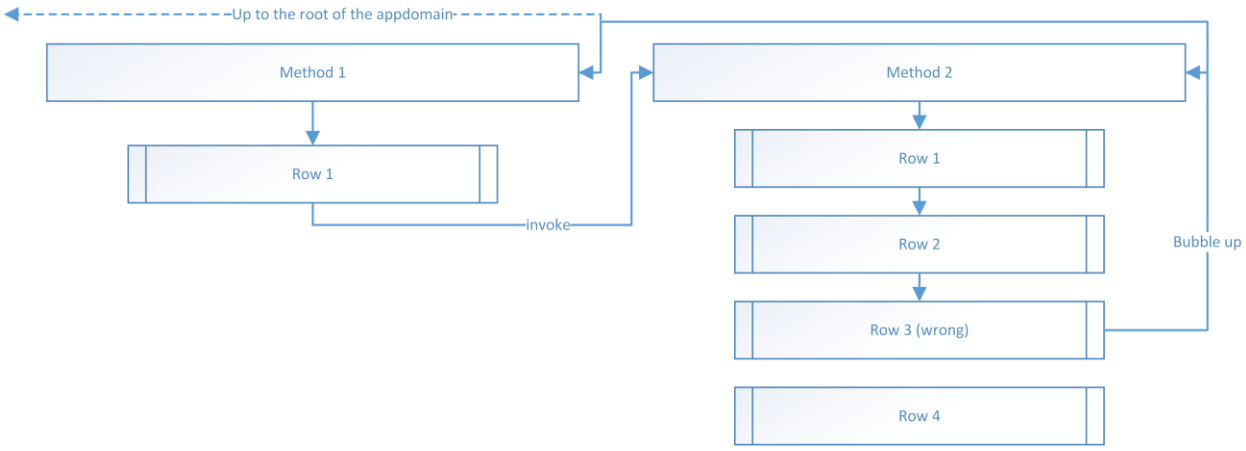

This makes objects nearest speeding their access time and also reduce whole address space



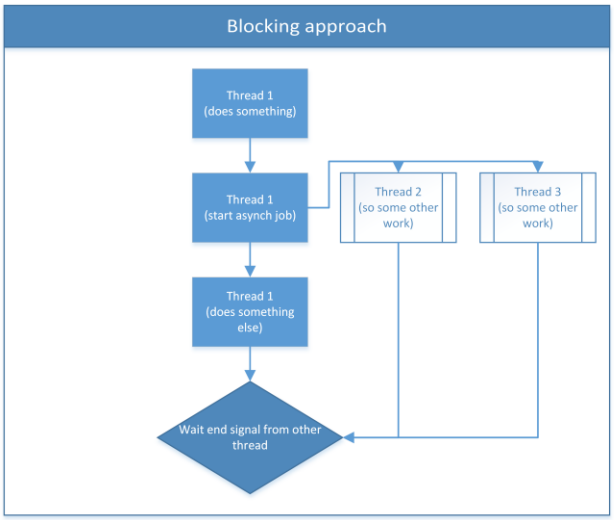

App1
IOPermission | Deny
IOPermission | Allow
IOPermission | Deny
DefaultDomain
App1.dll
AppDomain 1
App1.IO.dll
System.IO.dll
AppDomain 2
App1.XML.dll
System.XML.dll
Neutral domain
Mscorlib.dll
System.dll

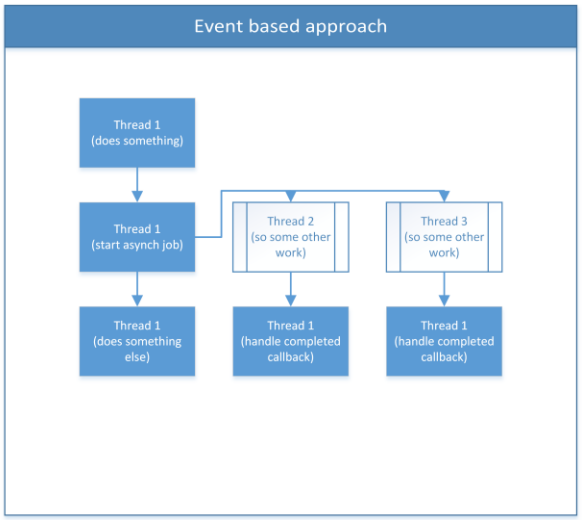

App1
IOPermission | Deny
DefaultDomain
App1.dll
IOPermission | Deny
AppDomain Temp
Plugin.v100.dll
IOPermission | Deny
AppDomain Temp
Plugin.v120.dll
Neutral domain
Mscorlib.dll
System.dll

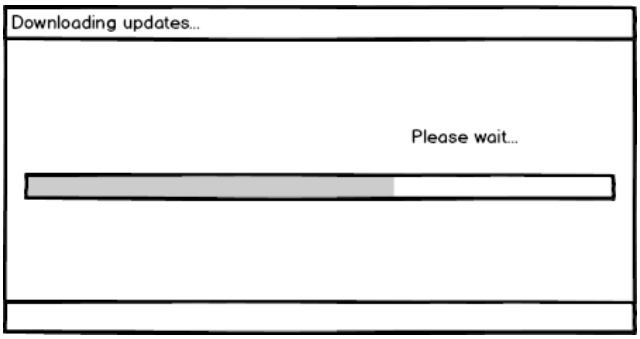

App1
IOPermission | Deny
DefaultDomain
App1.dll
AppDomain Rule ##
RuleEngine.Rule#.dll
Neutral domain
Mscorlib.dll
System.dll
RuleEngine.Shared.dll


Up to the root of the appdomain
Method 1
Row 1
invoke
Method 2
Row 1
Row 2
Row 3 (wrong)
Row 4
Bubble up

# Chapter 4 - Asynchronous Programming



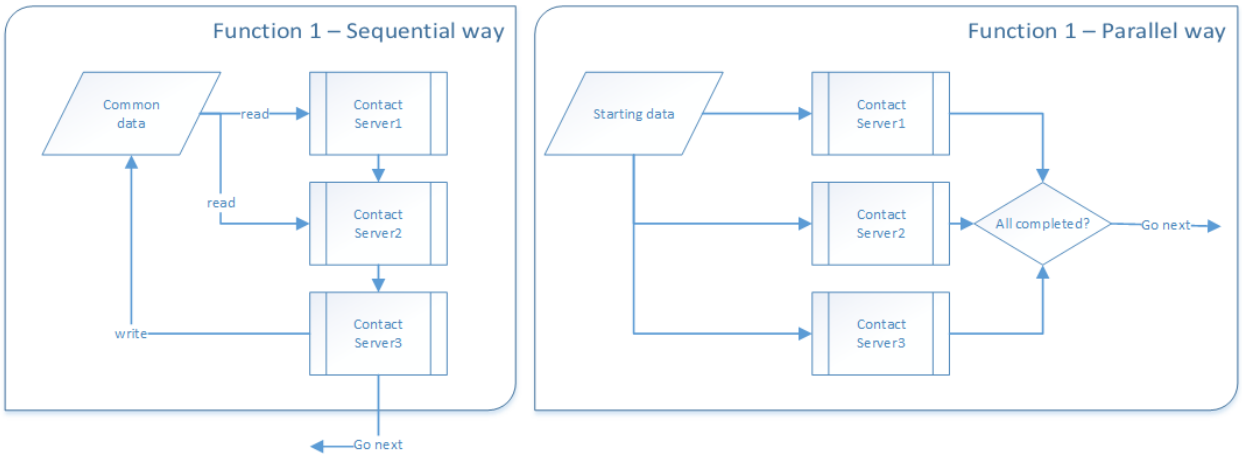



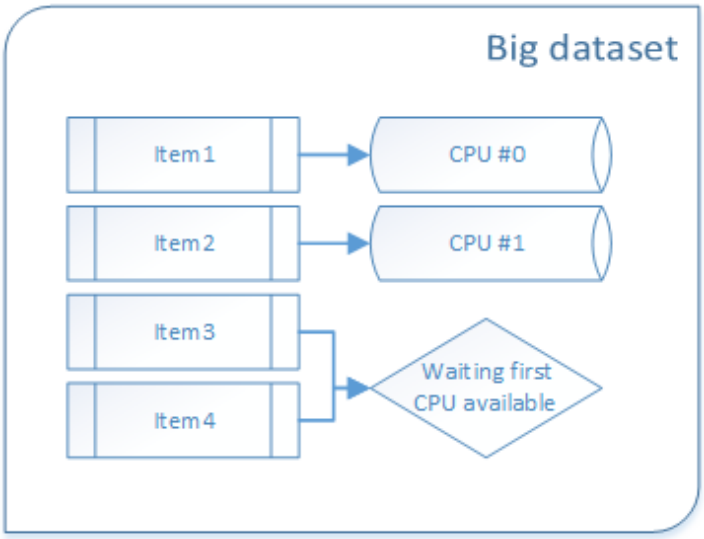

## Remarks

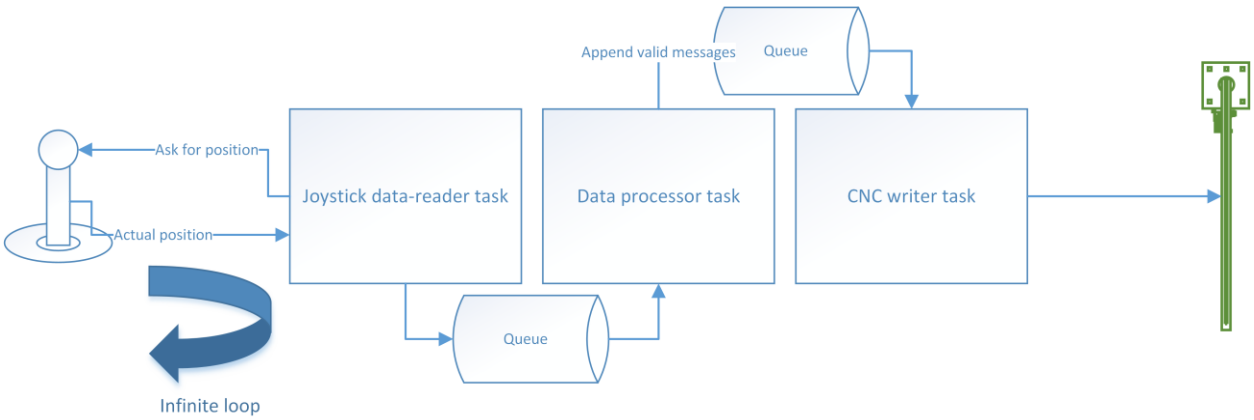
**Tip**

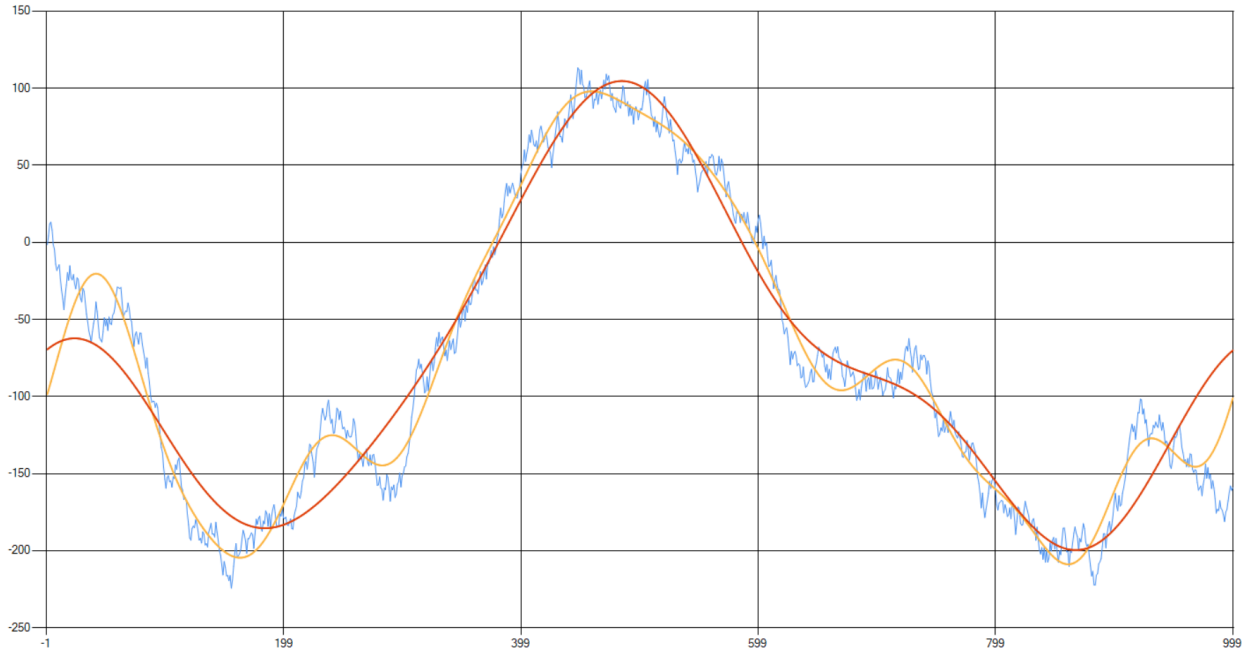
The FromAsync overloads that take an *asyncResult* parameter are not as efficient as the overloads that take a *beginMethod* parameter. If performance is an issue, use the overloads that provide the *beginMethod/endMethod* pattern.

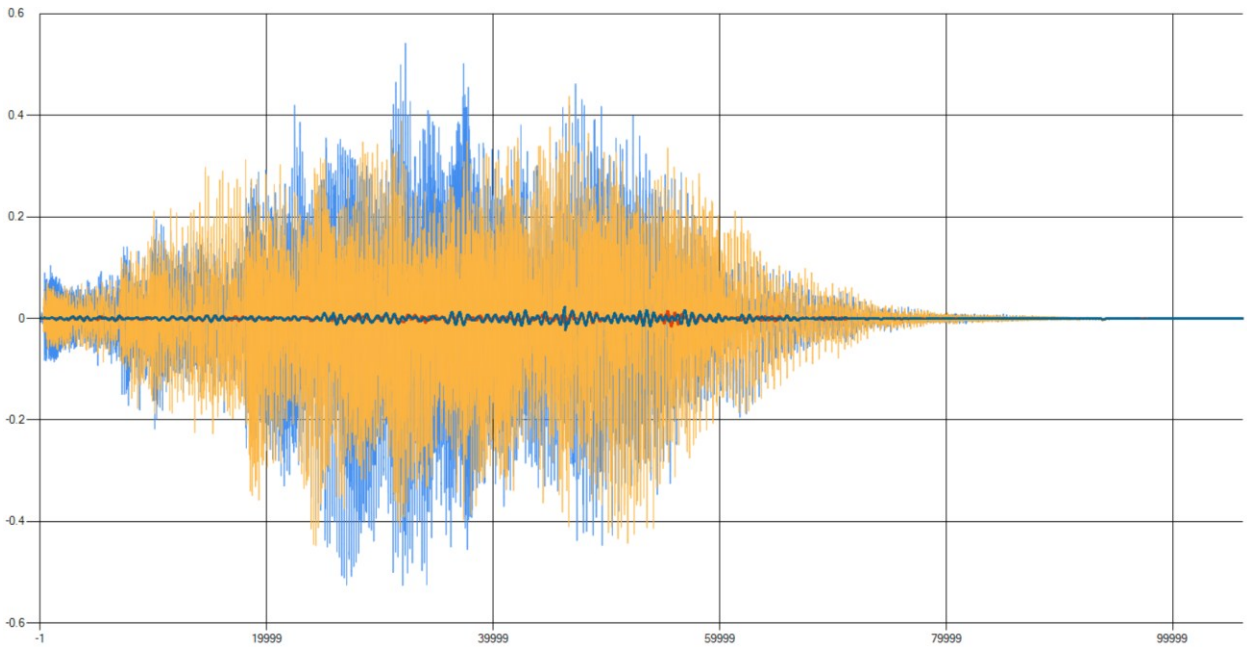
# Chapter 5 - Programming for Parallelism



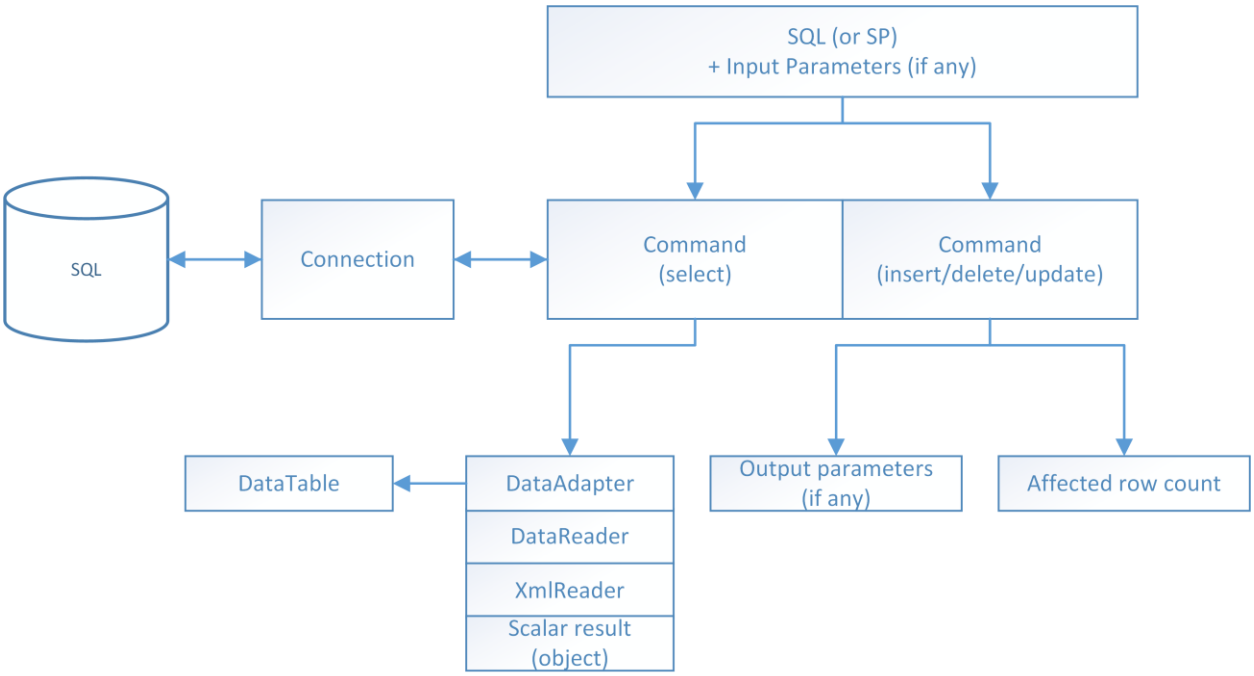



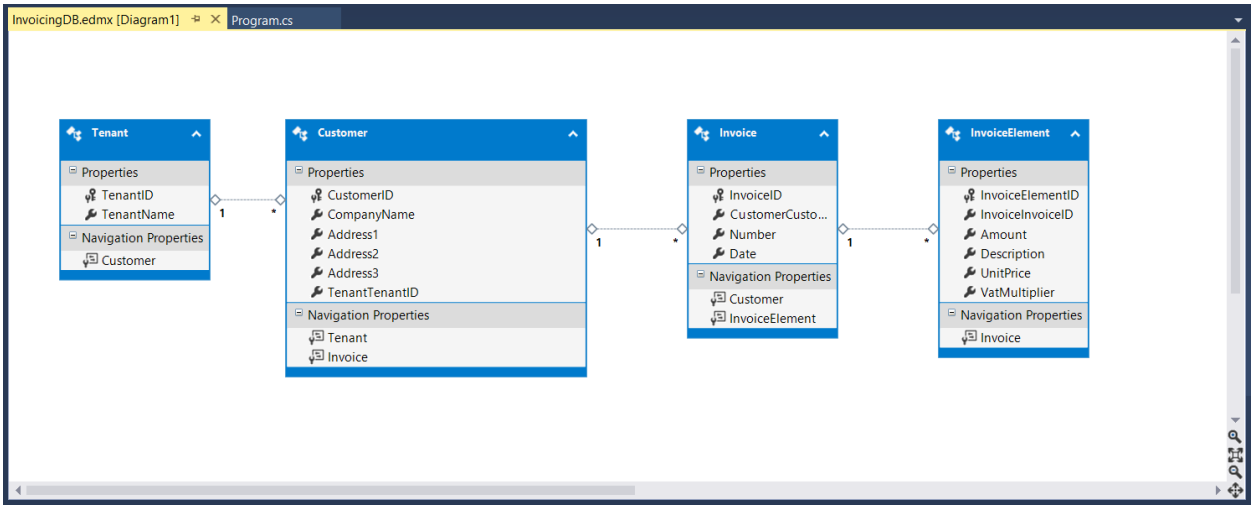

Append valid messages
Queue
Ask for position
Joystick data-reader task
Data processor task
CNC writer task
Actual position
Queue
Infinite loop


150
100
50
0
-50
-100
-150
-200
-250
-1
199
399
599
799
999

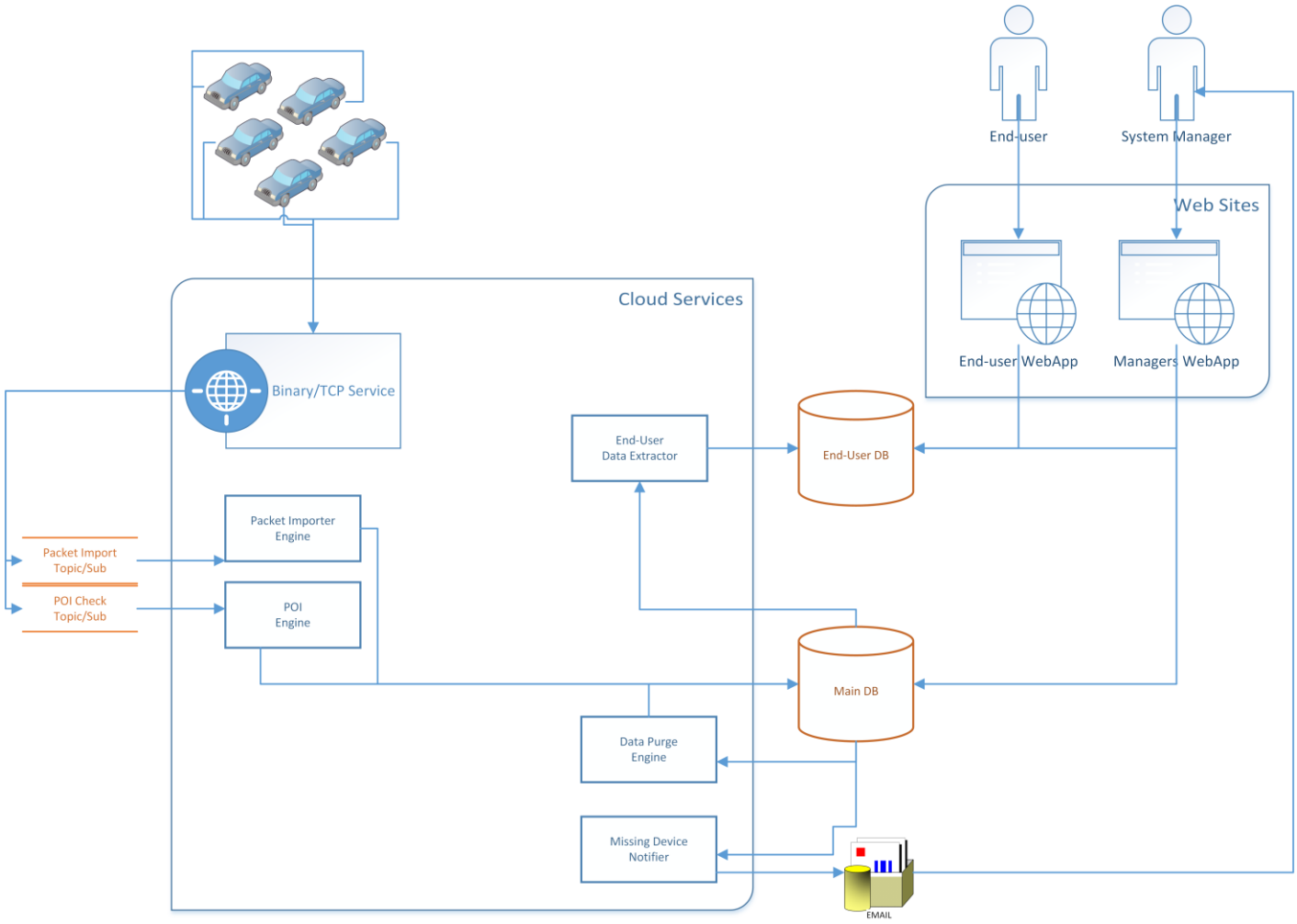

0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-1
19999
39999
59999
79999
99999

# Chapter 7 - Database Querying



# Chapter 8 - Programming for Big Data



Microsoft Azure

Subscriptions tonyexpo@msn.com

SQL DATABASES 3
STORAGE 9
HDINSIGHT 0
MEDIA SERVICES 0
SERVICE BUS 3
VISUAL STUDIO ONLINE 0
CACHE 0
BIZTALK SERVICES 0
RECOVERY SERVICES 0
CDN 0
AUTOMATION 0

## service bus

| NAMESPACE NAME | STATUS | TYPE | LOCATION | SUBSCRIPTION | CREATED DATE |
|---|---|---|---|---|---|
| tonysb1 | Active | Messaging | North Europe | tonyexpo@msn.com | 5/21/2015 10:52:59 PM |
| tonysb2 | Active | Messaging | North Europe | tonyexpo@msn.com | 5/27/2015 2:56:55 PM |
| tonysb4 | Active | Messaging | North Europe | tonyexpo@msn.com | 5/27/2015 4:37:15 PM |

NEW CREATE CONNECTION INFORMATION DELETE

## Access connection information

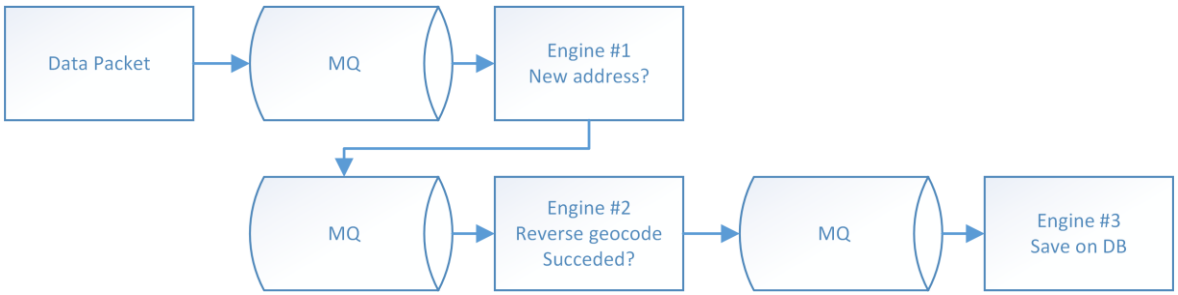
Use this connection information to manage namespace 'tonysb1'. You can also use authorization policies configured here to connect to all entities in this namespace.

**SAS**

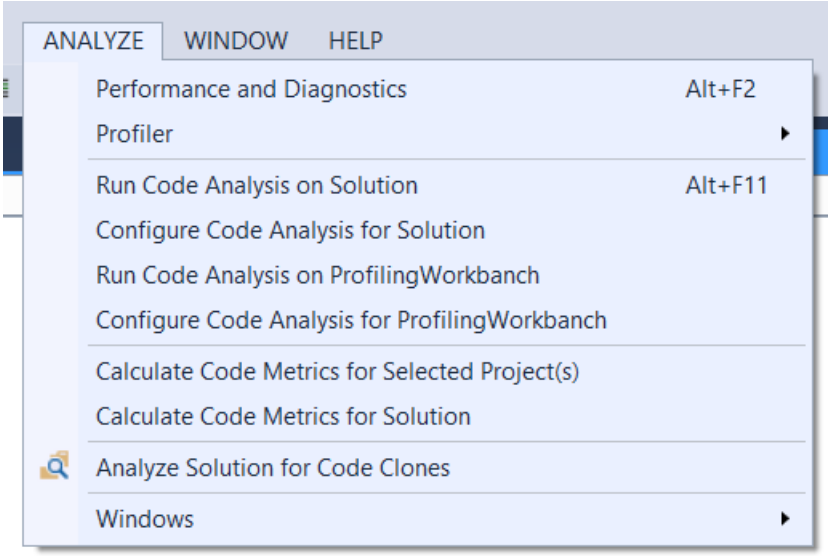
| NAME | CONNECTION STRING |
|---|---|
| RootManageSharedAccessKey | Endpoint=sb://tonysb1.servicebus.windows.net/;SharedAccessKeyName=RootManage |

**ACS**

Looking for ACS connection information? Please see here for more information regarding using ACS with Service Bus.


Data Packet
MQ
Engine #1
New address?
MQ
Engine #2
Reverse geocode
Succeded?
MQ
Engine #3
Save on DB

# Chapter 9 - Analyzing Code Performance



## Analysis Target

Change Target

**Startup Project**

ProfilingWorkbanch

## Available Tools

Show target specific tools

- [ ] CPU Usage
  See where the CPU is spending time executing your code. Useful when the CPU is the performance bottleneck
- [ ] Memory Usage
  Investigate application memory to find issues such as memory leaks
- [x] Performance Wizard
  CPU Sampling, Instrumentation, .NET Memory allocation, and Resource Contention

## Not Applicable Tools

- [ ] Energy Consumption
  Examine where energy is consumed in your application
- [ ] GPU Usage
  Examine GPU usage in your application. Useful to determine whether CPU or GPU is the performance bottleneck. Only available on supported graphics cards.
- [ ] HTML UI Responsiveness
  Examine where time is spent in your website or application
- [ ] JavaScript Function Timing
  See the elapsed time and call counts of your JavaScript functions. Useful when your code is waiting on I/O or other non-CPU intensive operations
- [ ] JavaScript Memory
  Investigate the JavaScript heap to help find issues such as memory leaks
- [ ] XAML UI Responsiveness
  Examine where time is spent in your application

Performance Wizard -- Page 1 of 3

**Specify the profiling method**

Profiling your application can help diagnose performance problems and identify the most common expensive methods in your application. To begin, choose a profiling method from the options below.

**What method of profiling would you like to use?**

- (●) **CPU sampling (recommended)**
  Monitor CPU-bound applications with low overhead
- ( ) **Instrumentation**
  Measure function call counts and timing
- ( ) **.NET memory allocation**
  Track managed memory allocation
- ( ) **Resource contention data (concurrency)**
  Detect threads waiting for other threads

Read more about profiling methods

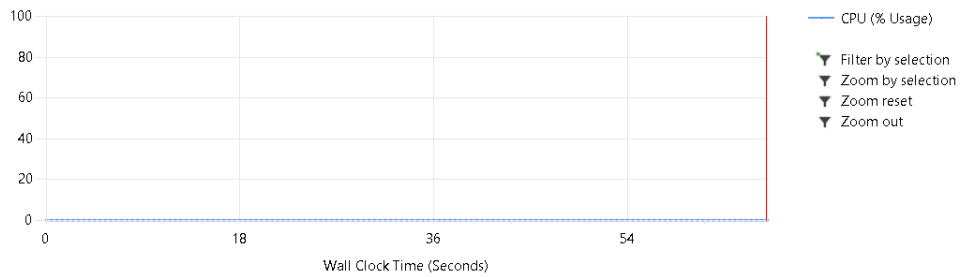
< Previous | Next > | Finish | Cancel

ProfilingWorkbanch150329.vspx | MainViewModel.cs | MainWindow.xaml | MainWindow.xaml.cs | SimpleModel.cs

Current View: Summary

**Sample Profiling Report**

123 total samples collected




**Notifications**

- Show All Code

**Report**

- Show Marks
- Show Trimmed Call Tree
- Show Hot Lines
- Compare Reports...
- Export Report Data...
- Save Analyzed Report...
- Filter Report Data
- Toggle Full Screen
- Set Symbol Paths...

**Hot Path**

| Function Name | Inclusive Samples % | Exclusive Samples % |
|---|---|---|
| ProfilingWorkbanch.exe | 100.00 | 0.00 |
| ProfilingWorkbanch.App.Main | 100.00 | 0.00 |
| **[PresentationFramework.ni.dll]** | **100.00** | **100.00** |

Related Views: Call Tree Functions

**Functions Doing Most Individual Work**

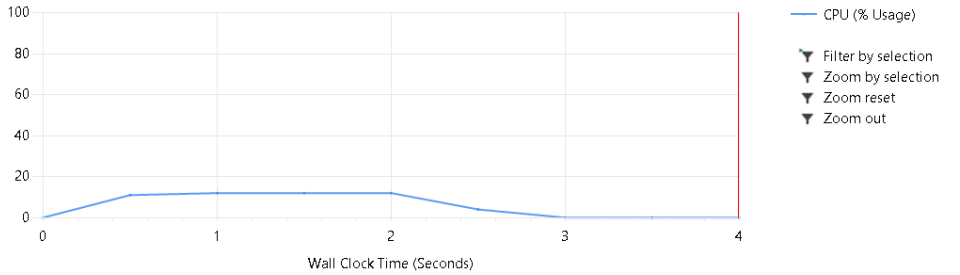
| Name | Exclusive Samples % |
|---|---|
| [PresentationFramework.ni.dll] | 100.00 |
| ProfilingWorkbanch.App.Main | 0.00 |

BindableBase.cs | ProfilingWorkbanch150329(4).vspx | SimpleModel.cs | MainViewModel.cs | MainWindow.xaml | MainWindow.xaml.cs

Current View: Summary

**Sample Profiling Report**

2,129 total samples collected




**Notifications**

- Show All Code

**Report**

- Show Marks
- Show Trimmed Call Tree
- Show Hot Lines
- Compare Reports...
- Export Report Data...
- Save Analyzed Report...
- Filter Report Data
- Toggle Full Screen
- Set Symbol Paths...

**Hot Path**

| Function Name | Inclusive Samples % | Exclusive Samples % |
|---|---|---|
| ProfilingWorkbanch.exe | 100.00 | 0.00 |
| ProfilingWorkbanch.MainViewModel.<.ctor>b__0 | 85.35 | 0.00 |
| ProfilingWorkbanch.SimpleModel.GenerateMocks | 85.35 | 0.00 |
| **[System.Core.ni.dll]** | **85.35** | **85.35** |

Related Views: Call Tree Functions

**Functions Doing Most Individual Work**

| Name | Exclusive Samples % |
|---|---|
| [System.Core.ni.dll] | 85.35 |
| [PresentationFramework.ni.dll] | 14.51 |

ProfilingWorkbanch150329(5).vspx

Current View: Function Details

**ProfilingWorkbanch.SimpleModel.GenerateMocks**

ProfilingWorkbanch.exe

| Calling functions | Current function | Called functions |
|---|---|---|
| ctor>b__0 86.9% | GenerateMocks 86.9%<br>Function Body < 0.1% | [System.Core.ni.dll] 86.9% |

Related Views: Caller/Callee Functions

Performance metric: Inclusive Samples %

**Function Code View**

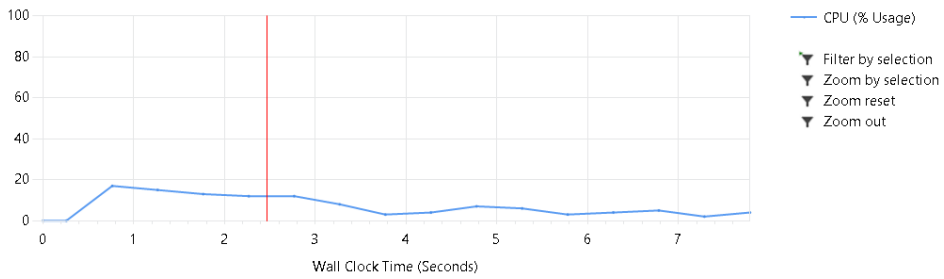
c:\Temp\vs2013\ProfilingWorkbanch\ProfilingWorkbanch\SimpleModel.cs

```
{
    var r = new Random(DateTime.Now.GetHashCode());
86.9 %  return Enumerable.Range(1, 1000).Select(i =>
        {
            Thread.SpinWait(r.Next(100000, 1000000));
            return new SimpleModel
            {
                Name = string.Format("ITEM #{0}", i),
                Value = r.NextDouble() * (double)i,
            };
        })
        .ToArray();
}
    }
}
```

100 %

ProfilingWorkbanch150330.vsp

Current View: Summary




**Notifications**

View Guidance

**Report**

Show Trimmed Call Tree
Compare Reports...
Export Report Data...
Save Analyzed Report...
Filter Report Data
Toggle Full Screen
Set Symbol Paths...

### Hot Path

| Name | Inclusive % | Exclusive % |
|---|---|---|
| ProfilingWorkbanch.exe | 100.00 | 0.00 |
| **ProfilingWorkbanch.App.Main()** | **52.92** | **0.06** |
| **ProfilingWorkbanch.MainViewModel.<.ctor>b__0()** | **47.08** | **0.07** |

Related Views: Call Tree Functions

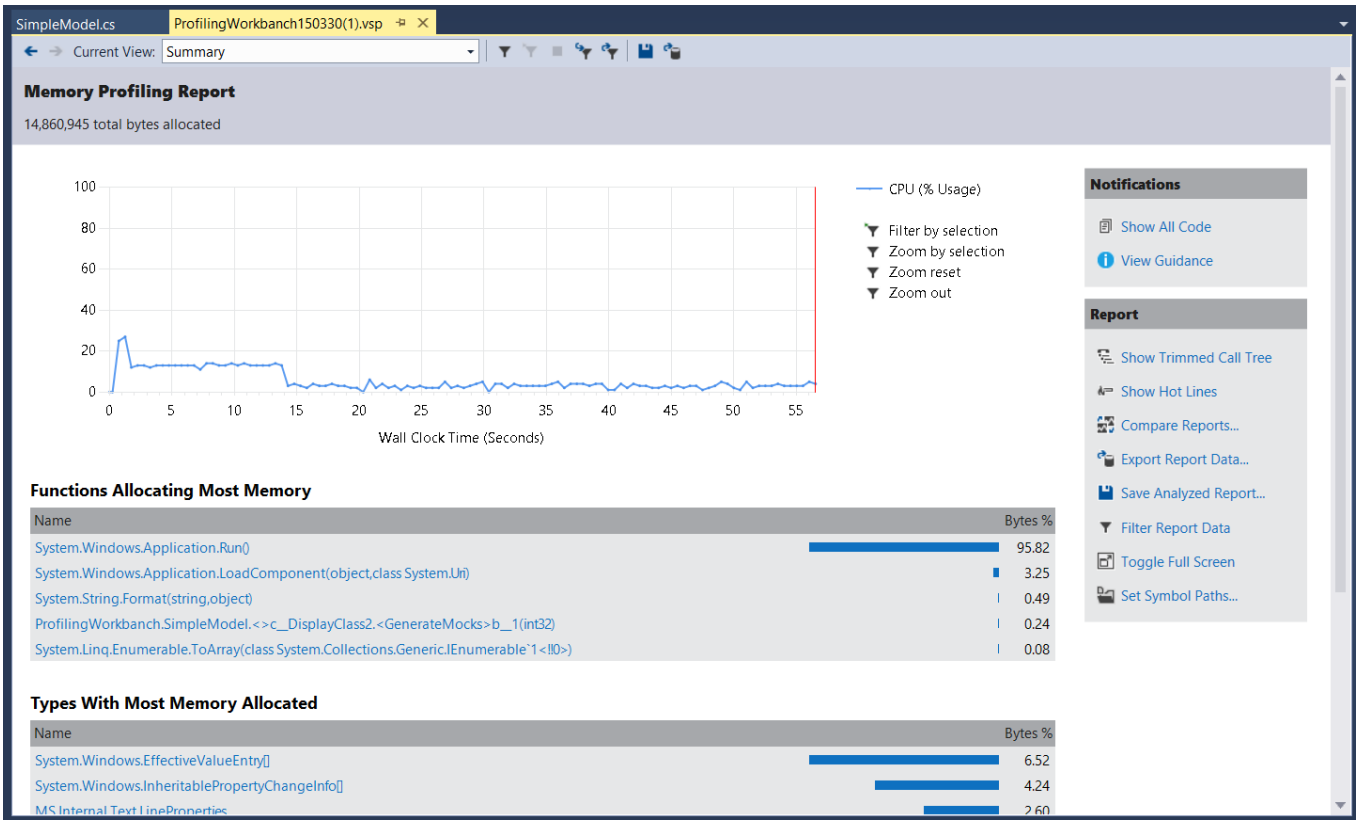
### Functions With Most Individual Work

| Name | Exclusive Time % |
|---|---|
| System.Windows.Application.Run() | 51.85 |
| System.Threading.Thread.SpinWait(int32) | 46.86 |
| System.Windows.Application.LoadComponent(object,class System.Uri) | 0.90 |
| System.Threading.Tasks.TaskFactory.StartNew(class System.Action) | 0.08 |
| System.Linq.Enumerable.Select(class System.Collections.Generic.IEnumerable`1<!!0>,class System.Func`2<!!0,!!1>) | 0.07 |

ProfilingWorkbanch150330.vsp

Current View: Call Tree

| Function Name | Number of Calls | Elapsed Inclusive Time % | Elapsed Exclusive Time % | Avg Elaps |
|---|---|---|---|---|
| ProfilingWorkbanch.exe | 0 | 100.00 | 0.00 | |
| ProfilingWorkbanch.App.Main() | 1 | 52.92 | 0.06 | |
| System.Windows.Application.Run() | 1 | 52.85 | 51.85 | |
| ProfilingWorkbanch.MainWindow..ctor() | 1 | 0.99 | 0.01 | |
| ProfilingWorkbanch.BindableBase.add_PropertyChanged(class System.ComponentModel.PropertyChange | 1,000 | 0.01 | 0.00 | |
| ProfilingWorkbanch.App.InitializeComponent() | 1 | 0.01 | 0.01 | |
| ProfilingWorkbanch.MainViewModel.<.ctor>b__0() | 1 | 47.08 | 0.07 | |
| ProfilingWorkbanch.SimpleModel.GenerateMocks() | 1 | 47.00 | 0.00 | |
| System.Linq.Enumerable.ToArray(class System.Collections.Generic.IEnumerable`1<!!0>) | 1 | 46.92 | 0.03 | |
| ProfilingWorkbanch.SimpleModel.<>c__DisplayClass2.<GenerateMocks>b__1(int32) | 1,000 | 46.89 | 0.01 | |
| System.Threading.Thread.SpinWait(int32) | 1,000 | 46.86 | 46.86 | |
| System.String.Format(string,object) | 1,000 | 0.01 | 0.01 | |
| ProfilingWorkbanch.SimpleModel.set_Name(string) | 1,000 | 0.00 | 0.00 | |
| ProfilingWorkbanch.SimpleModel.set_Value(float64) | 1,000 | 0.00 | 0.00 | |
| System.Random.Next(int32,int32) | 1,000 | 0.00 | 0.00 | |
| System.Random.NextDouble() | 1,000 | 0.00 | 0.00 | |
| ProfilingWorkbanch.SimpleModel..ctor() | 1,000 | 0.00 | 0.00 | |
| System.Linq.Enumerable.Select(class System.Collections.Generic.IEnumerable`1<!!0>,class System.Func`2< | 1 | 0.07 | 0.07 | |
| System.DateTime.get_Now() | 1 | 0.01 | 0.01 | |
| System.Linq.Enumerable.Range(int32,int32) | 1 | 0.00 | 0.00 | |
| System.Random..ctor(int32) | 1 | 0.00 | 0.00 | |
| System.Object.GetHashCode() | 1 | 0.00 | 0.00 | |
| System.Func`2..ctor(object,native int) | 1 | 0.00 | 0.00 | |
| ProfilingWorkbanch.BindableBase.Notify(string) | 1 | 0.00 | 0.00 | |
| System.ComponentModel.PropertyChangedEventHandler.Invoke(object,class System.ComponentModel.Pr | 1 | 0.00 | 0.00 | |
| System.ComponentModel.PropertyChangedEventArgs..ctor(string) | 1 | 0.00 | 0.00 | |


SimpleModel.cs
ProfilingWorkbanch150330(1).vsp
Current View: Summary
Memory Profiling Report
14,860,945 total bytes allocated
100
80
60
40
20
0
0 5 10 15 20 25 30 35 40 45 50 55
Wall Clock Time (Seconds)
CPU (% Usage)
Filter by selection
Zoom by selection
Zoom reset
Zoom out
Notifications
Show All Code
View Guidance
Report
Show Trimmed Call Tree
Show Hot Lines
Compare Reports...
Export Report Data...
Save Analyzed Report...
Filter Report Data
Toggle Full Screen
Set Symbol Paths...
Functions Allocating Most Memory
Name Bytes %
System.Windows.Application.Run() 95.82
System.Windows.Application.LoadComponent(object,class System.Uri) 3.25
System.String.Format(string,object) 0.49
ProfilingWorkbanch.SimpleModel.<>c__DisplayClass2.<GenerateMocks>b__1(int32) 0.24
System.Linq.Enumerable.ToArray(class System.Collections.Generic.IEnumerable`1<!!0>) 0.08
Types With Most Memory Allocated
Name Bytes %
System.Windows.EffectiveValueEntry[] 6.52
System.Windows.InheritablePropertyChangeInfo[] 4.24
MS.Internal.TextLineProperties 2.60

SimpleModel.cs | ProfilingWorkbanch150330(1).vsp

Current View: Call Tree

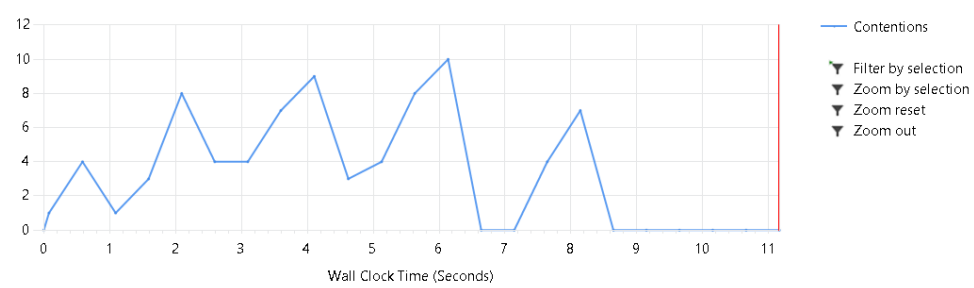
| Function Name | Module Name | Inclusive Allocations | Exclusive Allocations | Inclusive Bytes | Exclusive Bytes |
|---|---|---|---|---|---|
| ProfilingWorkbanch.exe | | 432,419 | 0 | 14,860,945 | 0 |
| ProfilingWorkbanch.App.Main() | ProfilingWorkbanch.exe | 427,330 | 0 | 14,738,623 | 0 |
| System.Windows.Application.Run() | PresentationFramework. | 427,330 | 412,837 | 14,738,623 | 14,239,826 |
| ProfilingWorkbanch.MainWindow..ctor() | ProfilingWorkbanch.exe | 14,493 | 1 | 498,797 | 104 |
| ProfilingWorkbanch.MainWindow.Initia | ProfilingWorkbanch.exe | 14,149 | 1 | 488,091 | 40 |
| System.Windows.Application.LoadC | PresentationFramework. | 14,146 | 14,019 | 487,845 | 482,474 |
| ProfilingWorkbanch.MainViewM | ProfilingWorkbanch.exe | 127 | 1 | 5,371 | 32 |
| System.Threading.Tasks.Task | mscorlib.ni.dll | 126 | 126 | 5,339 | 5,339 |
| System.Uri..ctor(string,valuetype Sys | System.ni.dll | 2 | 2 | 206 | 206 |
| System.Windows.Window..ctor() | PresentationFramework. | 343 | 343 | 10,602 | 10,602 |
| ProfilingWorkbanch.MainViewModel.<.ctor>b__0 | ProfilingWorkbanch.exe | 5,089 | 1 | 122,322 | 28 |
| ProfilingWorkbanch.SimpleModel.GenerateM | ProfilingWorkbanch.exe | 5,068 | 3 | 121,742 | 64 |
| ProfilingWorkbanch.BindableBase.Notify(strin | ProfilingWorkbanch.exe | 15 | 1 | 416 | 12 |
| [clr.dll] | clr.dll | 5 | 5 | 136 | 136 |

ConcurrencyWorkbanch150330(1).vsp | Program.cs

Current View: Summary

**Concurrency Profiling Report**

74 total contentions




**Notifications**

- Show All Code
- View Guidance

**Report**

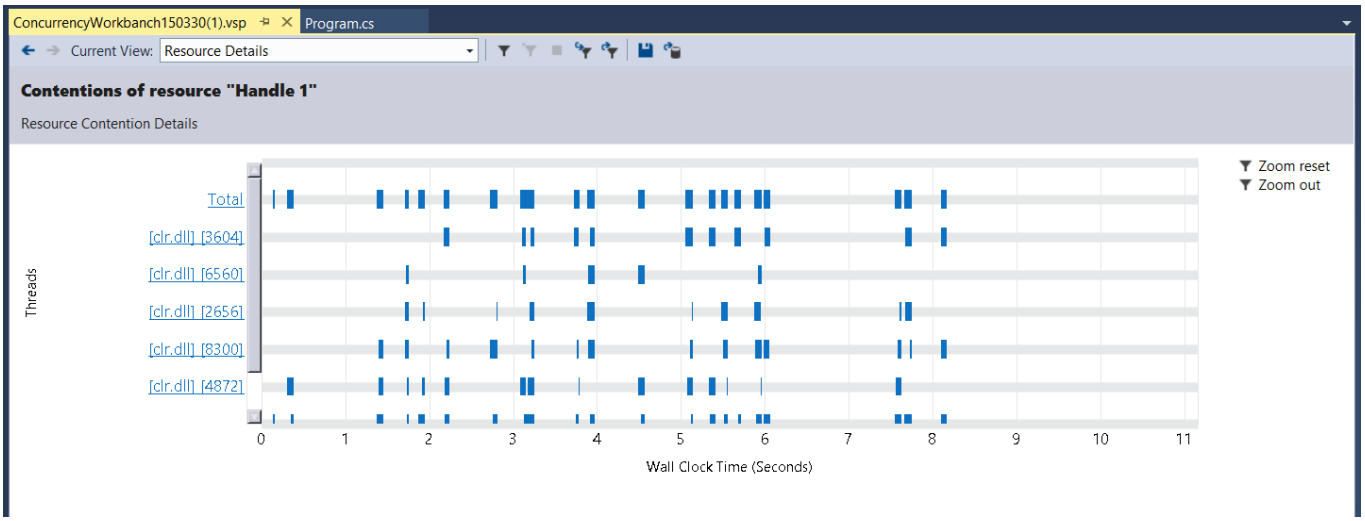
- Show Trimmed Call Tree
- Show Hot Lines
- Compare Reports...
- Export Report Data...
- Save Analyzed Report...
- Filter Report Data
- Toggle Full Screen
- Set Symbol Paths...

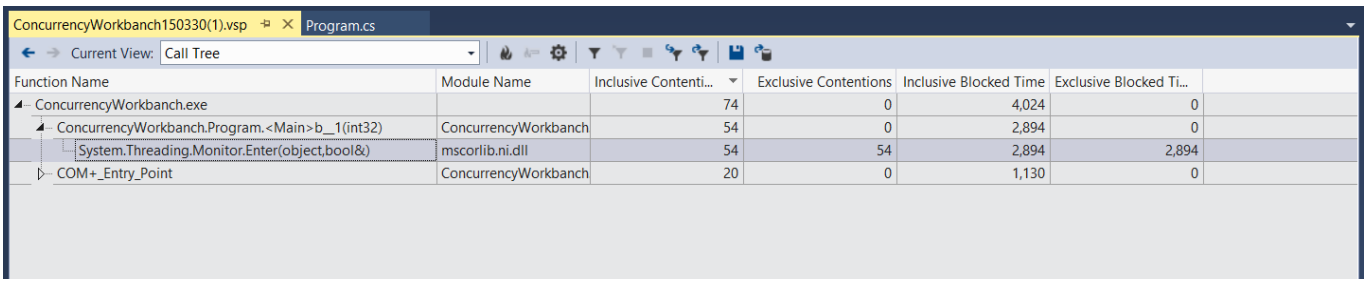
**Most Contended Resources**

| Name | Contentions % | Contentions |
|---|---|---|
| Handle 1 | 100.00 | 74 |

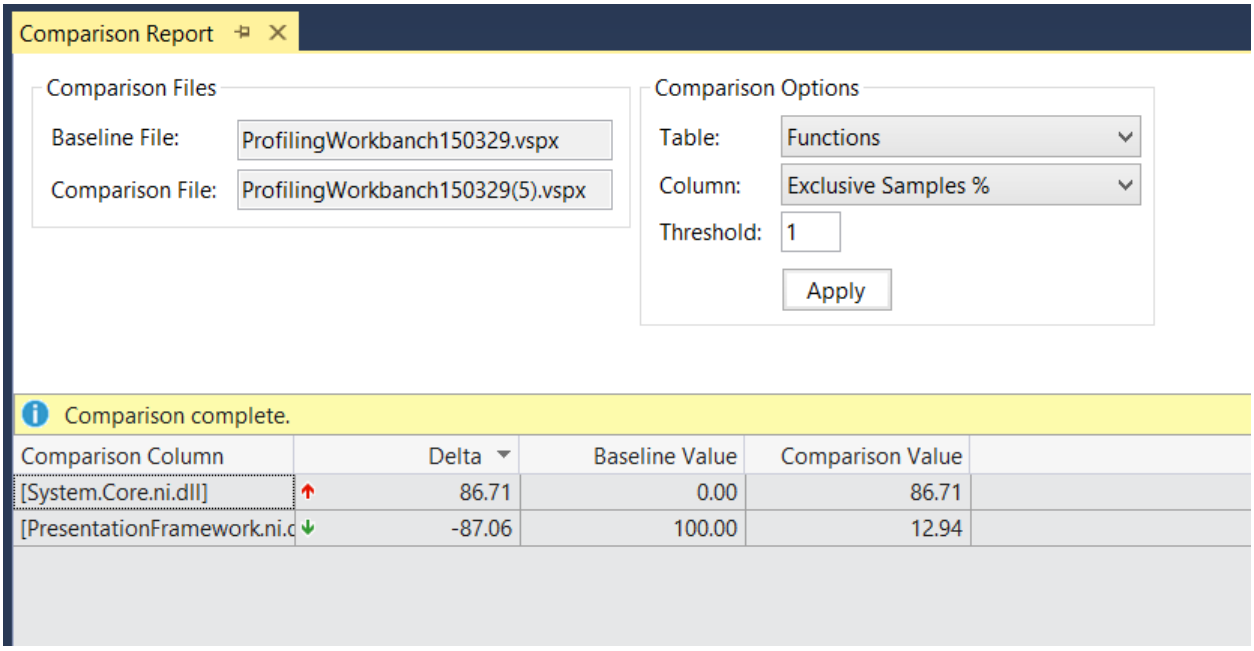
**Most Contended Threads**

| ID | Name | Contentions % | Contentions |
|---|---|---|---|
| 6228 | COM+_Entry_Point | 27.03 | 20 |
| 4872 | [clr.dll] | 18.92 | 14 |
| 8300 | [clr.dll] | 18.92 | 14 |
| 3604 | [clr.dll] | 14.86 | 11 |
| 2656 | [clr.dll] | 13.51 | 10 |

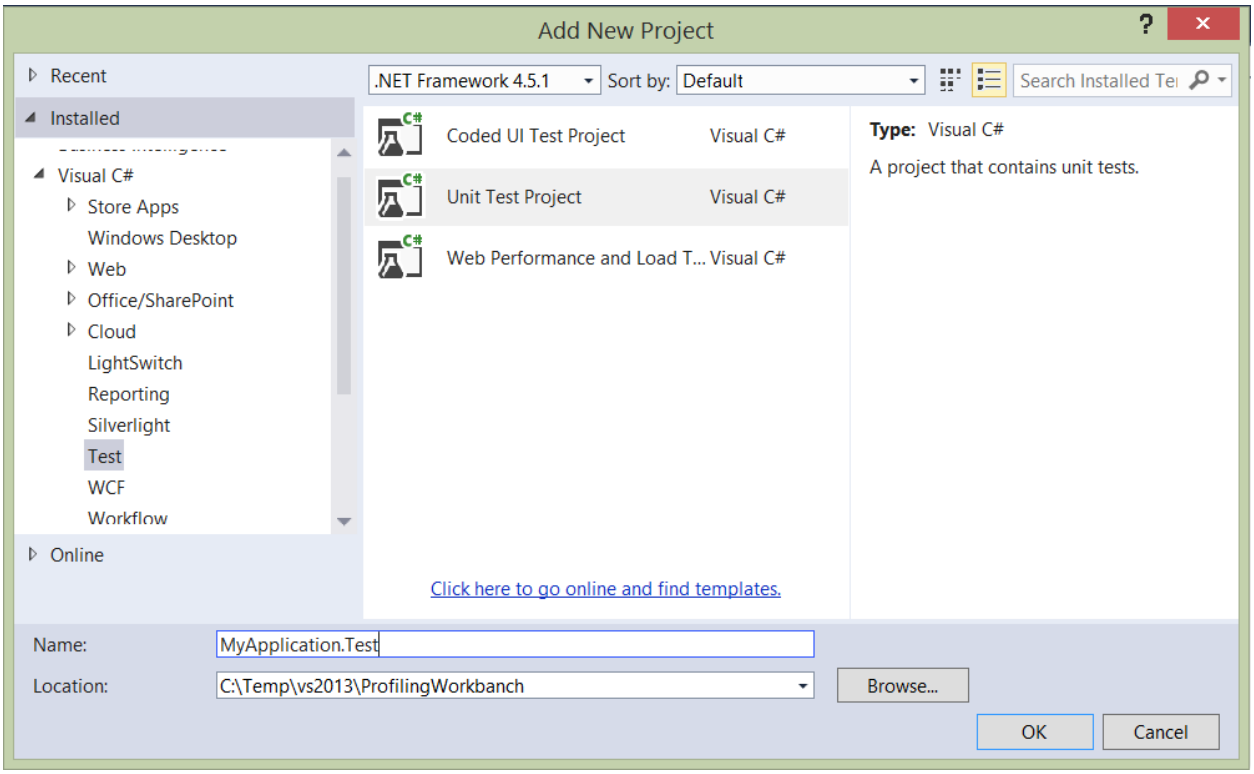


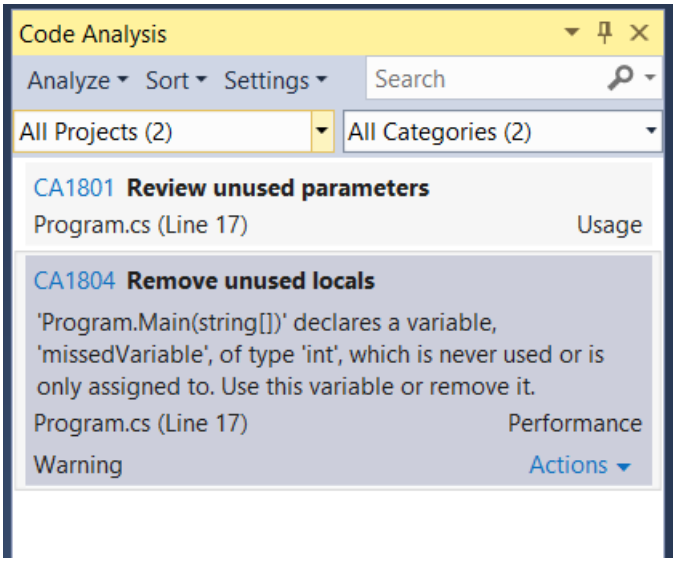



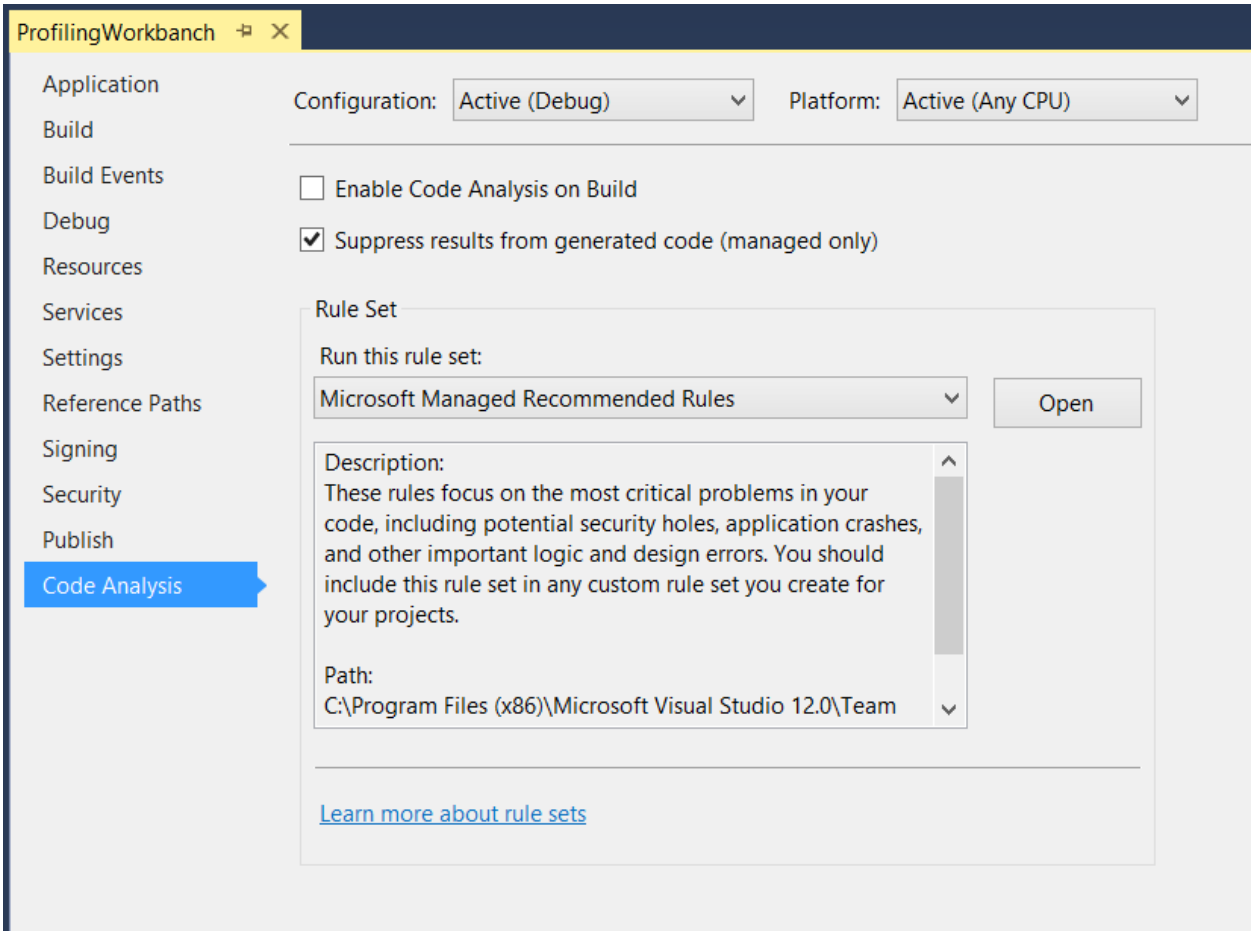

| Function Name | Module Name | Inclusive Contenti... | Exclusive Contentions | Inclusive Blocked Time | Exclusive Blocked Ti... |
|---|---|---|---|---|---|
| ConcurrencyWorkbanch.exe | | 74 | 0 | 4,024 | 0 |
| ConcurrencyWorkbanch.Program.<Main>b__1(int32) | ConcurrencyWorkbanch | 54 | 0 | 2,894 | 0 |
| System.Threading.Monitor.Enter(object,bool&) | mscorlib.ni.dll | 54 | 54 | 2,894 | 2,894 |
| COM+_Entry_Point | ConcurrencyWorkbanch | 20 | 0 | 1,130 | 0 |

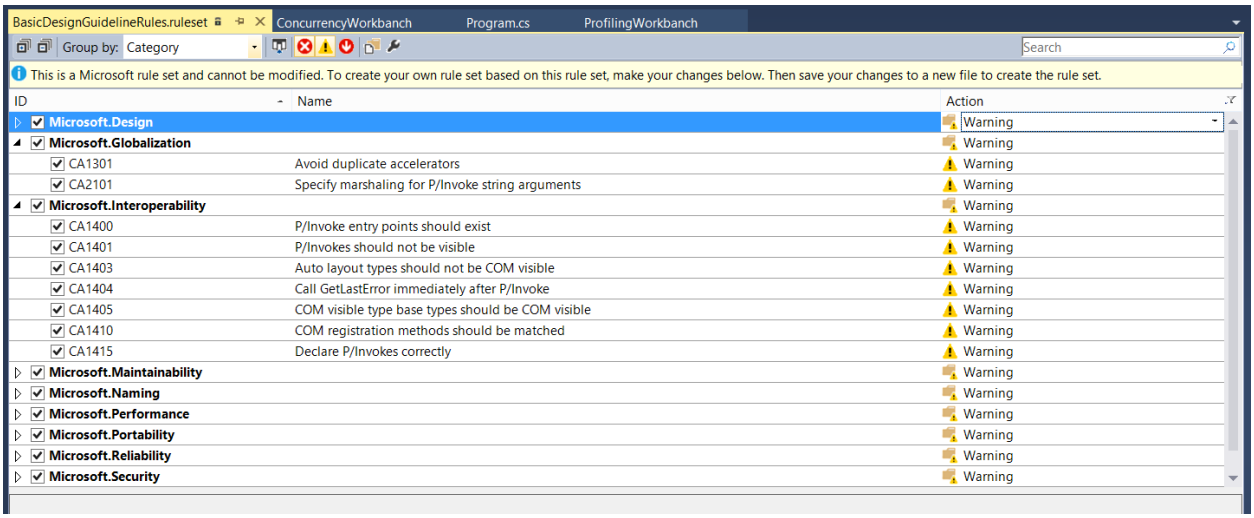



| Comparison Column | Delta | Baseline Value | Comparison Value |
|---|---|---|---|
| [System.Core.ni.dll] | 86.71 | 0.00 | 86.71 |
| [PresentationFramework.ni.c | -87.06 | 100.00 | 12.94 |


Add New Project
Recent
Installed
Visual C#
Store Apps
Windows Desktop
Web
Office/SharePoint
Cloud
LightSwitch
Reporting
Silverlight
Test
WCF
Workflow
Online
.NET Framework 4.5.1
Sort by: Default
Search Installed Te
Coded UI Test Project Visual C#
Unit Test Project Visual C#
Web Performance and Load T... Visual C#
Type: Visual C#
A project that contains unit tests.
Click here to go online and find templates.
Name: MyApplication.Test
Location: C:\Temp\vs2013\ProfilingWorkbanch
Browse...
OK
Cancel


Code Analysis
Analyze
Sort
Settings
Search
All Projects (2)
All Categories (2)
CA1801 Review unused parameters
Program.cs (Line 17)
Usage
CA1804 Remove unused locals
'Program.Main(string[])' declares a variable, 'missedVariable', of type 'int', which is never used or is only assigned to. Use this variable or remove it.
Program.cs (Line 17)
Performance
Warning
Actions


ProfilingWorkbanch
Application
Build
Build Events
Debug
Resources
Services
Settings
Reference Paths
Signing
Security
Publish
Code Analysis
Configuration: Active (Debug)
Platform: Active (Any CPU)
Enable Code Analysis on Build
Suppress results from generated code (managed only)
Rule Set
Run this rule set:
Microsoft Managed Recommended Rules
Open
Description:
These rules focus on the most critical problems in your code, including potential security holes, application crashes, and other important logic and design errors. You should include this rule set in any custom rule set you create for your projects.
Path:
C:\Program Files (x86)\Microsoft Visual Studio 12.0\Team
Learn more about rule sets


BasicDesignGuidelineRules.ruleset
ConcurrencyWorkbanch
Program.cs
ProfilingWorkbanch
Group by: Category
Search
This is a Microsoft rule set and cannot be modified. To create your own rule set based on this rule set, make your changes below. Then save your changes to a new file to create the rule set.
ID
Name
Action
Microsoft.Design Warning
Microsoft.Globalization Warning
CA1301 Avoid duplicate accelerators Warning
CA2101 Specify marshaling for P/Invoke string arguments Warning
Microsoft.Interoperability Warning
CA1400 P/Invoke entry points should exist Warning
CA1401 P/Invokes should not be visible Warning
CA1403 Auto layout types should not be COM visible Warning
CA1404 Call GetLastError immediately after P/Invoke Warning
CA1405 COM visible type base types should be COM visible Warning
CA1410 COM registration methods should be matched Warning
CA1415 Declare P/Invokes correctly Warning
Microsoft.Maintainability Warning
Microsoft.Naming Warning
Microsoft.Performance Warning
Microsoft.Portability Warning
Microsoft.Reliability Warning
Microsoft.Security Warning

Code Metrics Results

Filter: None  Min:  Max:

| Hierarchy | Maintainability Index | Cyclomatic Complexity | Depth of Inheritance | Class Coupling | Lines of Code |
|---|---|---|---|---|---|
| ConcurrencyWorkbanch (Debug) | 70 | 7 | 1 | 8 | 15 |
| ProfilingWorkbanch (Debug) | 88 | 22 | 9 | 21 | 34 |
| TestingWorkbanch (Debug) | 97 | 4 | 1 | 1 | 4 |
| TestingWorkbanch.Test (Debug) | 82 | 2 | 1 | 5 | 5 |