



GIESSWEIN
Software-Solutions

Ingolstädter .NET Developer Group

Internals of .NET CLR – Wie funktioniert der Motor?

Ing. Christian Giesswein, MSc.

www.software.tirol

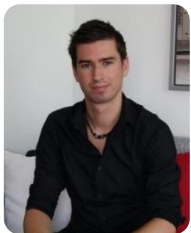
@giessweinweb

About Me



GIESSWEIN
Software-Solutions

- Ing. Christian Giesswein, MSc.
 - www.software.tirol
 - christian@software.tirol
- Consulting, Training und Entwicklung
- Technologisch am Puls der Zeit
 - Services & Technologien von Microsoft
 - C#, ASP.NET, WCF, WPF, ADO.NET, Entity Framework..
 - Desktopentwicklung
 - Datenbankentwicklung
 - Cloudentwicklung



Microsoft
CERTIFIED
Professional

Microsoft
CERTIFIED
Trainer

Microsoft
CERTIFIED
Solutions Developer
Windows Store Apps

Microsoft
CERTIFIED
Technology
Specialist

Microsoft
CERTIFIED
Professional
Developer

Microsoft
CERTIFIED
Solutions Developer
Web Applications



Sechs einfache Fragen die wir uns stellen:

- Wieviel DLLs werden während der Ausführung geladen?
- Wieviel Methoden hat die Klasse „Person“?
- Wann wird die Klasse „gejited“ ?
- Wieviel Bytes belegt die Klasse „Person“ ?
- Wieviel AppDomains/Threads gibt es?
- Wer räumt „p“ weg? Wer die Strings? Und Wie?

```
public class Program
{
    private static void Main()
    {
        Person p = new Person()
        {
            Firstname = "Christian",
            Lastname = "Giesswein"
        };
    }
}

internal class Person
{
    public string Firstname { get; set; }
    public string Lastname { get; set; }

    public void SayHello()
    {
        Console.WriteLine("SayHello has been called");
    }
}
```

Die CLR – Common Language Runtime



GIESSWEIN
Software-Solutions

.NET

- Entity Framework
- ASP.NET
- WPF
- Windows Forms
- ...

CLR

- Base Class Libraries (BCL)
- JIT & NGEN
- Garbage Collection
- Security Model (AppDomain)
- Exception Handling
- Loader von Assemblies

Wo die Reise beginnt...

- Doppelklick auf eine *.exe führt eine Anwendung aus, doch wer bestimmt ob diese .NET „fähig“ ist?
- → Portable Execution File (PE)
 - dumpbin.exe zeigt Details

AppDomain



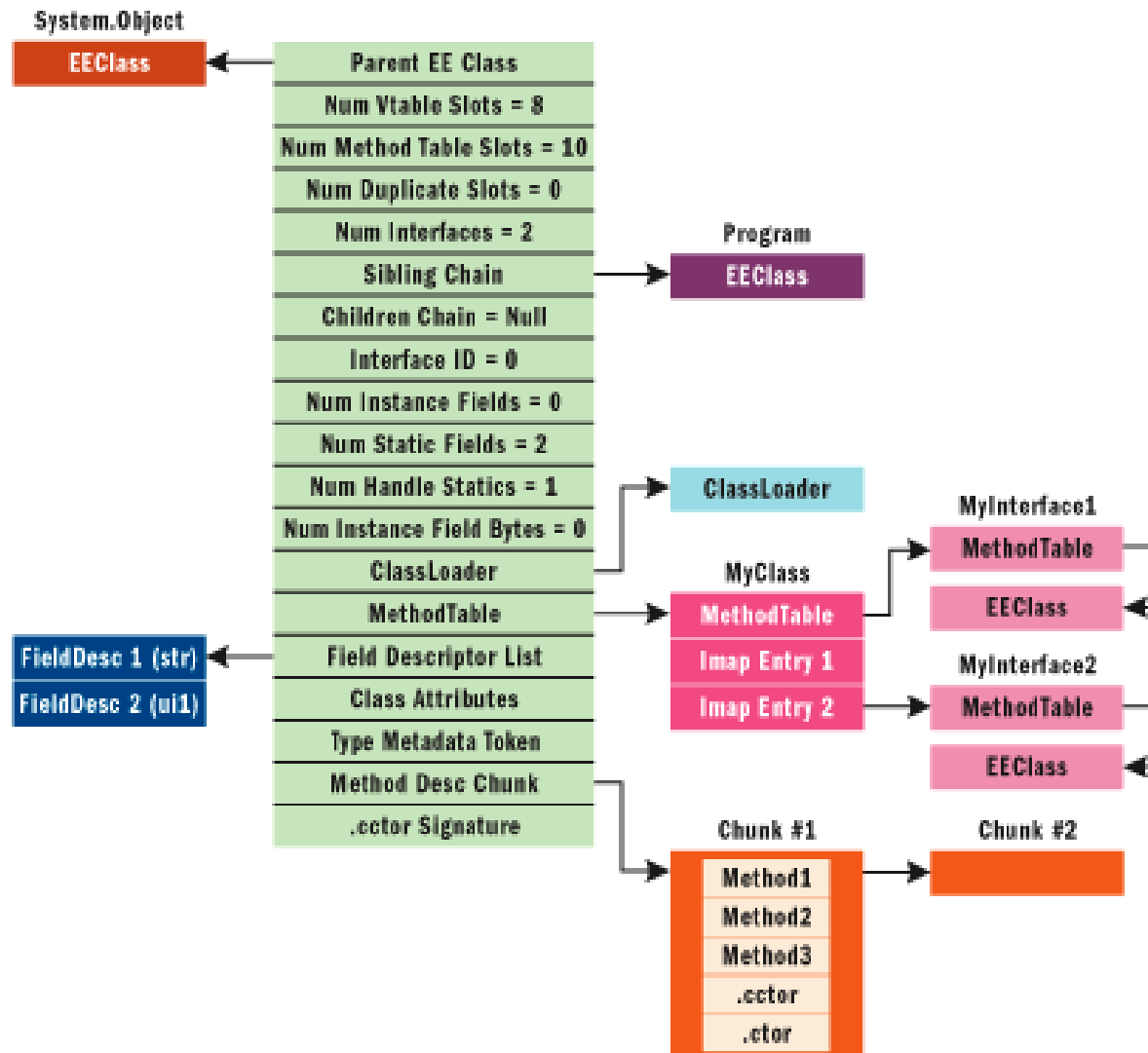
GIESSWEIN
Software-Solutions

Three Domains in a Managed Process			
System Domain (singleton)	Shared Domain (singleton)	Default AppDomain	
High-Frequency Heap	High-Frequency Heap	High-Frequency Heap	
Low-Frequency Heap	Low-Frequency Heap	Low-Frequency Heap	
Stub Heap	Stub Heap	Stub Heap	
Handle Table	Handle Table	Handle Table	
LOH Handle Table	LOH Handle Table	LOH Handle Table	
Interface Vtable Map	Interface Vtable Map	Interface Vtable Map	
Assembly Cache	Assembly Cache	Assembly Cache	
Context	Context	Context	
Security Descriptor	Security Descriptor	Security Descriptor	
Global Interface Vtable Map	Assembly Map	Interface Vtable Map	
Global String Literal Map	DLSRecords	String Literal Map	
Default Domain			
System Assembly			
Global Interface ID Table			
Process Heap	JIT Code Heap	GC Heap	Large Object Heap

EEClass



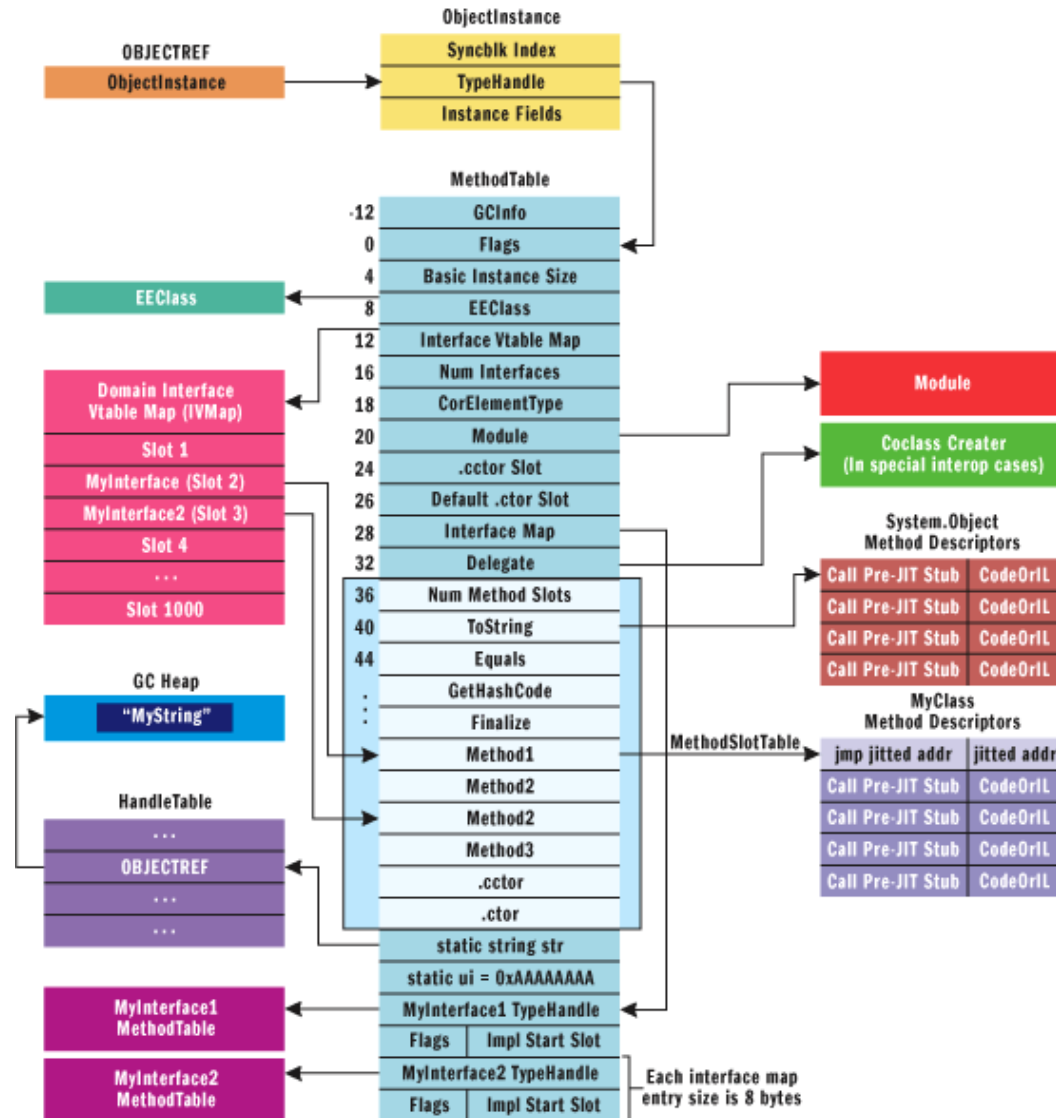
GIESSWEIN
Software-Solutions



MethodTable



GIESSWEIN
Software-Solutions



- Mit WinDbg wird es möglich in den „Motor“ hineinzuschauen und zu analysieren
 - <http://msdn.microsoft.com/en-us/windows/hardware/hh852365.aspx>
 - Herunterladen & Installieren der „Windows Debugging Tools“
- Für .NET gibt es eine wichtige Erweiterung SOS.dll (wird bei jeder Installation mitgeliefert)
 - SOS.dll – „Son Of Strike“.dll
 - <http://blogs.msdn.com/b/jasonz/archive/2003/10/21/53581.aspx>
 - C:\Windows\Microsoft.NET\Framework\v4.0.30319

- Vereinfacht gesagt gibt es „zwei“ Speicherbereiche:
 - Stack
 - Heap
- Der Stack stellt das „Ablagefach“ dar
 - Welche Variablen/Referenzen liegen herum?
 - Gültig bis zum Verlassen des aktuellen StackFrames

- !DumpMT
 - MethodTable
- !DumpMD
 - MethodDescription
- !DumpObj
 - Object
- !DumpHeap
 - Was liegt am Heap?

- Drei Generationen von Objekten
 - Generation 0:
 - "Frische" Instanzen
 - Generation 1:
 - Objekte eine Garbage Collection überlebt haben
 - Generation 2:
 - Objekte die „langlebig“ sind
- !GCRoot <Address>
 - Zeigt wer noch eine Referenz auf ein Objekt hat
- SOSEX (<http://www.stevetechspot.com>)
 - !dumpgen zeigt die Generation an

Kontakt

Telefon: +43 / 664 46 36 135

Email: christian@software.tirol

Web: www.software.tirol

Github.com/softwaretirol