

1. Record Nr.	UNISALENTO991003164399707536
Autore	Venuti, Ridolfino <1705-1763>
Titolo	Collectanea antiquitatum romanarum quas centum tabulis aeneis incisas et a Rodolphino Venuti ... notis illustratas exhibet Antonius Borioni
Pubbl/distr/stampa	(Romæ : ex Typographia Rochi Bernabò, 1736)
Descrizione fisica	xii, 80 p., [1], 104 [i.e 105] c. di tav. : ill.; fol.
Altri autori (Persone)	Borioni, Antonio <sec. XVIII>
Lingua di pubblicazione	Italiano
Formato	Microfilm
Livello bibliografico	Monografia
Note generali	Front. inciso. Frontoni, iniziali, finali. Incisioni calcografiche. Le c. di tav. precedono il testo Nota ms incollata sulla c. di guardia posteriore Riproduzione in microfiche dell'originale conservato presso la Biblioteca Apostolica Vaticana
Nota di bibliografia	Indicazioni bibliografiche nelle note.

2. Record Nr.	UNINA9910300467703321
Autore	Rahman Mohammad
Titolo	C# Deconstructed : Discover how C# works on the .NET Framework // by Mohammad Rahman
Pubbl/distr/stampa	Berkeley, CA : , : Apress : , : Imprint : Apress, , 2014
ISBN	1-4302-6671-6
Edizione	[1st ed. 2014.]
Descrizione fisica	1 online resource (165 p.)
Disciplina	005.2762
Soggetti	Microsoft software Microsoft .NET Framework Software engineering Microsoft and .NET Software Engineering/Programming and Operating Systems
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record.
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Contents at a Glance; Contents; About the Author; About the Technical Reviewer; Chapter 1: Introduction to Programming Language; Overview of the CPU; Instruction Set Architecture of a CPU; Memory: Where the CPU Stores Temporary Information; Concept of the OS; Concept of the Process; Concept of the Thread; What Is Virtualization?; Programming Language; Compilation and Interpretation; Birth of C# Language and JIT Compilation; // Microsoft (R) .NET Framework IL Disassembler Version 4.0.30319.1; The CLR; Road Map to the CLR; Tools Used in This Book; Son of Strike Debugging Extension DLL ConclusionFurther Reading; Chapter 2: The Virtual Machine and CLR; Virtual Machine; Problems with the Existing System; Optimization During Execution; Virtual Execution Environment; Components of the Virtual Execution Environment; CLR: Virtual Machine for .NET; CLR Supports Multiple Languages; Common Components of the CLR; Conclusion; Further Reading; Chapter 3: Assembly; What Is the Assembly?; Overview of Modules, Assemblies, and Files; Introduction to PE Files; Structure of the Assembly; Analysis of the Assembly; Section Header; .text Section; #~ stream; ModuleDef; TypeDef; MethodDef Reference TablesAssemblyRef; ModuleRef; TypeRef; MemberRef;

Assembly Loading; Inside the Bind, Map, Load Process; Binding to an Assembly; Consulting the Cache; Conclusion; Further Reading; Chapter 4: CLR Memory Model; Introduction; Memory Interaction between the CLR and OS; Windows Memory Management; Concept of the Process; Process Structure; Process Address Space; Concept of the Thread; Thread Address Space; Thread and Frames; Concept of the Virtual Memory; 32-bit and 64-bit Process Addressing; Virtual-to-Physical Address Mapping; Learn the Contents of a Particular Physical Memory Address

Find a Virtual Address and Its ContentsMemory-Mapped File; Conclusion; Further Reading; Chapter 5: CLR Memory Model II; CLR Memory Model: Application Domain; Finding an object in the Application Domain; Address Space of the Application Domain; Stack in the CLR; Address Space of the Stack; Heap; Heap and Address Space; objects; Garbage Collection; Generation 0; Generation 1; Generation 2; Conclusion; Further Reading; Chapter 6: CLR Execution Model; Overview of the CLR; The C# Program and the CLR; CLR Bootstrapping; CLR Address Space; Class Loader in the CLR; Locating the Main Entry Point

Stub Code for the ClassesVerification; Conclusion; Further Reading; Chapter 7: CLR Execution Model II; JIT Compilation; Method Stub of a Type; JIT-Compiled Status: NONE; JIT-Compiled Status: JIT; JIT-Compiled Status: PreJIT; How Many Times a Method Is JIT Compiled; Execution State of the CLR; Conclusion; Further Reading; Index

Sommario/riassunto

C# Deconstructed answers a seemingly simply question: Just what is going on, exactly, when you run C# code on the .NET Framework? To answer this question we will dig ever deeper into the structure of the C# language and the onion-skin abstraction layers of the .NET Framework that underpins it. We'll follow the execution thread downwards, first to MSIL (Microsoft Intermediate Language) then down through just-in-time compilation into Machine Code before finally seeing the results executed at the hardware level. The aim of this deep-dive is to provide you with a much more rounded knowledge of the environment within which you code exists. As a managed language, it's best-practice to let the Framework deal with device interaction but you'll find the experience of taking the cover off once in a while a very rewarding one that will greatly enrich your appreciate of the C# language and the way in which in functions.
