

NET Framework Application Essentials. 03/30/; 2 minutes to read Contributors. all; In this article. This section of www.nxgvision.com Framework documentation provides information about basic application development tasks in www.nxgvision.com Framework.

With Safari, you learn the way you learn best. Get unlimited access to videos, live online training, learning paths, books, tutorials, and more. .NET Framework is a development framework that provides a new programming interface to Windows services and APIs, and integrates a number of technologies that emerged from Microsoft during the late s. .NET initiative in July In April , Version 1. .NET Framework was released. This book describes this updated version of the. .NET platform consists of four separate product groups: .NET; a set of development tools, including Visual Studio. These components collectively form the largest part of the. Web services An offering of commercial web services, specifically the. .NET Services initiative; for a fee, developers can use these services in building applications that require them. Specialized servers A set of. These provide specialized functionality for relational data storage, email, and B2B commerce. Future versions of these products will increasingly support the. .NET-enabled, non-PC devices, from cell phones to game boxes. .NET While the main strategy of. .NET is to enable software as a service,. .NET is much more than that. In addition to embracing the Web, Microsoft. .NET acknowledges and responds to the following trends within the software industry: Traditional distributed technologies require high vendor-affinity and are unable to interoperate with the Web. Componentization Simplifies the integration of software components developed by different vendors and supports development of distributed applications. .NET provides a simpler way to build and deploy components. Enterprise services Allow the development of scalable enterprise applications without writing code to manage transactions, security, or pooling. .NET continues to support COM and component services, since these services greatly reduce the development time and effort required to build large-scale applications. A key goal of Microsoft. .NET is to enable the sharing of functionality across the Web among different platforms, devices, and programming languages. Maturity of IT industry Lessons that the software industry has learned from developing large-scale enterprise and web applications. A commercial web application must support interoperability, scalability, availability, security, and manageability. .NET facilitates all these goals. Although these are the sources of many ideas embodied by Microsoft. .NET platform consists of five main components, as shown in Figure As part of the. .NET strategy, Microsoft has promised to deliver more. .NET device software to facilitate a new generation of smart devices. .NET platform On top of the operating system is a collection of specialized server products that shortens the time required to develop large-scale business systems. Since web services are highly reusable across the Web, Microsoft provides a number of building-block services officially called. .NET Services that applications developers can use, for a fee. .NET Services that Microsoft offers include. .NET Passport allows you to use a single username and password at all web sites that support Passport authentication. .NET Alert providers, such as a business, to alert their consumers with important or up-to-the-minute information. Microsoft plans to add newer services, such as calendar, directory, and search services. Third-party vendors are also creating new web services of their own. The top layer of the. .NET architecture is a development tool called Visual Studio. .NET , which makes possible the rapid development of web services and other applications. A successor to Microsoft Visual Studio 6. And at the center of. .NET is the Microsoft. .NET Frameworkâ€™the main focus of this book. .NET Framework is a development and runtime infrastructure that changes the development of business applications on the Windows platform. .NET Framework embodies design goals that are both practical and ambitious. In this section, we discuss the main design goals of the Microsoft. .NET Framework, including support for components, language integration, application interoperation across the Web, simplified development and deployment, improved reliability, and greater security. Component Infrastructure Prior to the introduction of COM technology, developers had no standard way to integrate binary libraries without referring to or altering their source code. With the advent of COM, programmers were able to integrate binary components into their applications, similar to the way we can plug-and-play hardware components into our desktop PCs. Although

COM permits you to integrate binary components developed using any language, it does require you to obey the COM identity, lifetime, and binary layout rules. Realizing that these requirements result in frequent rewrites of similar code, .NET sets out to remove them. In the .NET world, all classes are ready to be reused at the binary level. You simply write a .NET class, which then becomes a part of an assembly to be discussed in Chapter 2 that inherently supports plug-and-play. In addition to providing a framework to make development easier, .NET removes the use of the registry for component registration and eliminates the requirements for extraneous plumbing code found in all COM components, including code to support IUnknown, class factories, component lifetime, registration, dynamic binding, and others. As long as your component meets all the rules spelled out in the COM specification, it can be instantiated and used by your applications. .NET supports not only language independence, but also language integration. This means that you can inherit from classes, catch exceptions, and take advantage of polymorphism across different languages. .NET components must support. For example, everything in .NET is an object of a specific class that derives from the root class called System. The CTS supports the general concepts of classes, interfaces, delegates which support callbacks, reference types, and value types. .NET base classes provide most of the base system types, such as those that support integer, string, and file manipulation. Because every language compiler must meet a minimum set of rules stipulated by the Common Language Specification CLS and generate code to conform to the CTS, different .NET languages can be used in the same application. In addition, the DCOM dynamic activation, protocol negotiation, and garbage collection facilities are proprietary, complex, and expensive. The solution is an open, simple, and lightweight protocol for distributed computing. Each time you set out to develop something in a different compiler, you had to learn a new API or a class library, because there is no consistency or commonality among these different libraries or interfaces. .NET solves this problem by providing a set of framework classes that every language uses. Such a framework removes the need for learning a new API each time you switch languages. Simplified Deployment Imagine this scenario: If you have ever encountered such a brutal "yet entirely possible" problem, you have entered DLL Hell. If you install an application that overwrites system DLLs, Windows will overwrite the added system DLLs with the original versions from the cache. In addition to this requirement, a shared DLL must have a unique hash value, public key, locale, and version number. This is possible because the executable that uses one of these DLLs is tightly bound to a specific version of the DLL during compilation. In addition to eradicating DLL Hell, .NET also removes the need for component-related registry settings. .NET stores all references and dependencies of .NET assemblies within a special section called a manifest see Chapter 2. In addition, assemblies can be either private or shared. Private assemblies are found using logical paths or XML-based application configuration files, and public assemblies are registered in the GAC; in both cases, the system will find your dependencies at runtime. If they are missing, you get an exception telling you exactly what happened. .NET brings back the concept of zero-impact installation and removal. This concept is the opposite of what you have to deal with in the world of COM. To set up a COM application, you have to register all your components after you have copied them over to your machine. Likewise, to uninstall the application, you should unregister your components to remove the registry entries prior to deleting your files. Again, if you fail to perform this step correctly, you will leave remnants in the registry that will be forever extant. .NET, you simply xcopy your files from one directory on a CD to another directory on your machine, and the application will run automatically. Reliability There are many programming languages and platforms in the commercial software industry, but few of them attempt to provide both a reliable language and a robust runtime or infrastructure.

Chapter 2 : .NET Overview - .NET Framework Essentials, 3rd Edition [Book]

Fully updated for version of www.nxgvision.com Framework,.NET Framework Essentials, 3rd Edition is an objective, concise, no-nonsense overview of the www.nxgvision.com Framework for developing web applications and services.

View all articles by John Spano It was written by Thuan Thai and Hoang Q. The book provides a very good general overview of that. NET is and what a programmer can accomplish with it. It also gives a good reference on the inner workings of assemblies and how the Visual Studio product creates them. The first few chapters start by defining the framework and what design goals went into it. The authors describe the two major sections of the framework, the common type system and the common language specification. After defining the framework, the authors describe how an assembly is created and what IL code is. They go into detail on how assemblies are loaded in to Windows and compare them to regular native code execution. The authors show how the common programming model allows for inheritance and code usage between individual languages. A quick overview of major framework classes is also given. For the remaining chapters, each gives an overview of a major area of. The first of these shows how to deploy. Message queues are also covered here. The authors detail how ADO. The different managed providers are also discussed. In chapter six an entirely new concept is covered called Web Services. Web Services provide a "Software as a Service" type of functionality. They allow public interaction through a web server to some data or functionality that can be directly used in other projects. The book shows how to discover, consume and write Web Services. The next two chapters describe the GUI interfaces available in. Both allow for very robust GUI for the web and regular windows client applications. Another very useful part of the book is the appendix. The appendix gives reference material to the different languages, acronyms, data types and common tools available to. I found the common utilities appendix invaluable. It provides a very quick reference on the options. I feel that this book should get two different ratings. I gave it a 10 of 10 for new to. NET platform yet, or have just started with it. It gives an invaluable overview of what. You should read this book. It will save you a huge amount of time later when you are trying to architect larger projects. When designing the more that is known about the design capabilities of the tools involved, the better the final design will be. For experienced programmers that have been programming with the. NET architecture, I give the book and 8. I found that the book is very good as a quick reference for seldom used tools and concepts in. NET much, you will find the chapters on them very informative.

Chapter 3 : Microsoft Download Center: Windows, Office, Xbox & More

www.nxgvision.com Framework edition of publisher O'Reilly® Essentials series is an excellent introduction to Microsoft's new software development framework. This book is suitable for any developer who wants to understand what this new initiative is all about without having to manually strip away the marketing hype.

NET is an advance in programming technology that greatly simplifies application development, both for traditional, proprietary applications and for the emerging paradigm of Web-based services. NET is a major new object-oriented programming language, C. But learning the new programming language is only part of the challenge. The much greater challenge is learning the. NET Framework and all its capabilities. NET training course covers the essentials of the. It pairs with the course "C Essentials" to form a five-day introduction to C and the. NET Framework for experienced programmers. The course is current to. It focuses on core portions of the. NET Framework that are common across many application areas. Separate courses are available in specific areas, such as ADO. NET and Web services. The course starts with an introduction to the architecture and key concepts of. It then discusses class libraries, assemblies, versioning, configuration, and deployment, which constitute a major advance in the simplicity and robustness of deploying Windows applications, ending the notorious "DLL hell. The following chapter continues the discussion of the. NET programming model, covering memory management, threading, asynchronous programming, application domains, marshal by value, marshal by reference, and. NET Security is introduced in some detail, including both code access security and role-based security. The next chapter covers interoperability of. The course concludes with an introduction to database programming using ADO. The course is practical, with many examples and a case study. The goal is to equip you to begin building significant applications using the. The student will receive a comprehensive set of materials, including course notes and all the programming examples. Course Goals Gain a thorough understanding of the philosophy and architecture of. NET Acquire a working knowledge of the. NET programming model and. NET Security Learn how to implement database applications using.

Chapter 4 : .NET Framework Essentials, 3rd Edition - O'Reilly Media

NET Framework Essentials is an objective, concise, and technical overview of the new www.nxgvision.com Framework for developing web applications and services. Specifically written for intermediate to advanced VB, C/C++, Java, and Delphi developers,.NET Framework Essentials is also useful to system architects and leaders who are assessing tools.

With Safari, you learn the way you learn best. Get unlimited access to videos, live online training, learning paths, books, tutorials, and more. No credit card required. NET Framework are many design goals that are practical yet extremely ambitious. In this section, we discuss the main design goals of the Microsoft. NET Framework, including better support for components, language integration, application interoperation across cyberspace, simple development and deployment, better reliability, and greater security. Component Infrastructure Prior to the existence of COM technology, Microsoft developers had no simple way to integrate binary libraries without referring to or altering their source code. With the advent of COM, programmers were able to integrate binary components into their applications, similar to the way we plug-and-play hardware components into our desktop PCs. While COM permits you to integrate binary components developed using any language, it does require you to obey the COM identity, lifetime, and binary layout rules. Realizing that all of these requirements result in frequent rewrites of similar code,. NET sets out to remove all of them. NET world, all classes are ready to be reused at the binary level. You simply write a .NET class, which then becomes a part of an assembly to be discussed in Chapter 2 , and it will support plug-and-play. NET removes the use of the registry for component registration and eliminates the requirements for extraneous plumbing code found in all COM components, including code to support IUnknown, class factories, component lifetime, registration, dynamic binding, and others. As long as your component meets all the rules spelled out in the COM specification, it can be instantiated and used by your applications. NET supports not only language independence, but also language integration. This means that you can inherit from classes, catch exceptions, and take advantage of polymorphism across different languages. NET components must support. For example, everything in .NET is an object of a specific class that derives from the root class called System. The CTS supports the general concepts of classes, interfaces, delegates which support callbacks , reference types, and value types. NET base classes provide most of the base system types, such as ones that support integer, string, and file manipulation. Because every language compiler must meet a minimum set of rules stipulated by the Common Language Specification CLS and generate code to conform to the CTS, different .NET languages can intermingle with one another. In addition, the DCOM dynamic activation, protocol negotiation, and garbage-collection facilities are proprietary, complex, and expensive. The solution is an open, simple, and lightweight protocol for distributed computing. Each time you set out to develop something in a different compiler, you had to learn a new API or a class library, because there is no consistency or commonality among these different libraries or interfaces. NET solution provides a set of framework classes and lets every language use it. Such a framework removes the need for learning a new API each time you switch languages. Simple Deployment Imagine this scenario: If you have ever encountered such a brutal “yet entirely possible” problem, you have entered DLL Hell. If you install an application that overwrites system DLLs, Windows will overwrite the added system DLLs with the original versions from the cache. In addition to this requirement, a shared DLL must have a unique hash value, public key, locale, and version number. Again, this is possible because the executable that uses one of these DLLs is tightly bound to the DLL during compilation. In addition to eradicating DLL Hell,. NET also removes the need for component-related registry settings. NET stores all references and dependencies of .NET assemblies within a special section called a manifest see Chapter 2. In addition, assemblies can be either private or shared. Private assemblies are found using logical paths or XML-based application configuration files, and public assemblies are registered in the GAC; in both cases the system will find your dependencies at runtime. If they are missing, you get an exception telling you exactly what happened. NET brings back the concept of zero-impact installation and removal. This concept is the opposite of what you have to deal with in the world of COM. To set up a COM application, you have to register all your components after you have copied them over to your machine. Likewise, to uninstall the

application, you should unregister your components to remove the registry entries prior to deleting your files. Again, if you fail to perform this step correctly, you will leave remnants in the registry that will be forever extant. NET, you simply xcopy your files from one directory on a CD to another directory on your machine, and the application will run automatically. Reliability There are many programming languages and platforms in the commercial software industry, but few of them attempt to provide both a reliable language and a robust runtime or infrastructure. NET as the next big thing. NET requires type safety. NET is derived from the mother of all classes, Object, which supports runtime type-identification features, content-dumping features, and so on. The CLR must recognize and verify types before they can be loaded and executed. This decreases the chances for rudimentary programming errors and prevents buffer overruns, which can be a security weakness. If you fail to do this, the allocated resources on your system will never be reclaimed even though they are no longer needed. Similar to Java, the .NET runtime tracks and garbage-collects all allocated objects that are no longer needed. Security When developing applications in the old days of DOS, Microsoft developers cared little about security because their applications ran on a single desktop with a single thread of execution. As soon as developers started developing client and server applications, things got a bit complicated: The problem became even more complex in the web environment, since you could unknowingly download and execute malicious applets on your machine. To mitigate these problems,. NET provides a number of security features. Unlike traditional security support whereby only access to the executable is protected,. NET goes further to protect access to specific parts of the executable code. For example, to take advantage of declarative security checks, you can prefix your method implementations with security attributes without having to write any code. To take advantage of imperative security checks, you write the code in your method to explicitly cause a security check. There are many other security facilities that .NET provides in an attempt to make it harder to penetrate your applications and system. In fact, if you use dumpbin. See Chapter 4 for more details. Get unlimited access to videos, live online training, learning paths, books, interactive tutorials, and more.

Chapter 5 : .NET Framework Application Essentials | Microsoft Docs

www.nxgvision.com Framework Essentials Second Edition. This books gives a very good overview of all major areas of the framework and shows the differences in the three main Microsoft languages that can target it, Visual Basic, C# and managed C++.

Chapter 6 : .NET Framework Essentials Using C# Training

www.nxgvision.com is an advance in programming technology that greatly simplifies application development, both for traditional, proprietary applications and for the emerging paradigm of Web-based services.

Chapter 7 : .Net Framework Essentials by Thuan L. Thai

This three-day course is designed to provide a sound introduction to the essentials of www.nxgvision.com Framework for programmers who already know the C# language and the fundamentals of Windows Forms.

Chapter 8 : Windows 8 How To: Install .NET and Windows Live Essentials â€” Dr. Z's Blog

At Microsoft our mission and values are to help people and businesses throughout the world realize their full potential.

Chapter 9 : .NET Framework Essentials, 2nd Edition - pdf - Free IT eBooks Download

NET Framework Essentials is an objective, concise, and technical overview of the new www.nxgvision.com Framework for developing web applications and services. Specifically written for intermediate to advanced VB, C/C++, Java, and

DOWNLOAD PDF NET FRAMEWORK ESSENTIALS

Delphi developers,.NET Framework Essentials is also useful to system architects and leaders who are assessing tools for future projects.