.NET e-Business Architecture

International Standard Book Number: 0-672-32219-6

Warning and Disclaimer

When reviewing corrections, always check the print number of your book. Corrections are made to printed books with each subsequent printing. To determine the printing of your book, view the copyright page. The print number is right-most number on the line below the "First Printing" line. For example, the following indicates the 4th printing of a title.

First Printing: November 2001

06 05 04 03 10 9 8 7 6 5 4

Misprint	Correction
Page 79, text in the Note: Microsoft XML Developer Center: **www.**msdn.microsoft.com/XML W3C XML: www.**w3c**.org.XML XML-Zone: **www.XML-Zone.com/**	Microsoft XML Developer Center: msdn.microsoft.com/XML W3C XML: www.**w3**.org.XML XML-Zone: **www.devx.com/xmlzone**
Page 84, third line of Note: SOAP Specification Version 1.1: **msdn.microsoft.com/xml/general/soapspec.asp** Web services Description Language 1.1: **msdn.microsoft.com/xml/general/wsdl.asp**	SOAP Specification Version 1.1: **www.w3.org/TR/SOAP** Web services Description Language 1.1: **www.w3.org/TR/wsdl**
Page 295, Note: The presentation services require the use of the business objects created for gasTIX. Both the assemblies and source code are available for download at www.gasTIX.**com**. Alternately, the assemblies are available for web service consumption at www.gasTIX.com**/WebServices**. You can reference them locally or through a web service...	The presentation services require the use of the business objects created for gasTIX. Both the assemblies and source code are available for download at www.gasTIX.**net**. Alternately, the assemblies are available for web service consumption at www.gasTIX.com**/participant.asmx**. You can reference them locally or through a web service...
Page 311, last sentence of the first paragraph under "Building the gasTIX Navigation User Control" Since the combined search functionality is only a small part of the functionality included in the gasTIX NavBar.ascx, the control built in this chapter will differ significantly from the control downloaded from **www.gasTIX.com**.	Since the combined search functionality is only a small part of the functionality included in the gasTIX NavBar.ascx, the control built in this chapter will differ significantly from the control downloaded from **www.gasTIX.net**.
Page 232, second paragraph, second line: ...application scope settings. The **global.asa** file can have any number of purposes, from setting...	...application scope settings. The **global.asax** file can have any number of purposes, from setting...

Page 346, first bullet on the page: • The developer is explicitly aware of the XML Web service being exposed and knows the URL for accessing that XML Web service. This is derived from a file with the .disco file extension or the SDL contract from a file with the **asmx** extension. For more information, see Discovery of Web services (DISCO) at **msdn.microsoft.com/xml/general/disco.asp**.	• The developer is explicitly aware of the XML Web service being exposed and knows the URL for accessing that XML Web service. This is derived from a file with the .disco file extension or the SDL contract from a file with the **.asmx** extension. For more information, see Discovery of Web services (DISCO) at **www.uddi.org/specification.html**.
Page 346, second to last paragraph from the bottom of the page, second sentence: For the gasBAGS example, enter the name of the gasTIX Web server, **www.gasTIX.com/**.	For the gasBAGS example, enter the name of the gasTIX Web server, **www.gasTIX.com/participant.asmx**.
Page 345, paragraph after the Tip: Now that we have a reference to our XML Web service, we can use it in our code. In our example, double-click on the WebForm1.ASPX file. Right-click on the WebForm1.ASPX file and click View Code. We need to add a reference to the XML Web service that we referenced from the **gasTIX XML Web server**...	Now that we have a reference to our XML Web service, we can use it in our code. In our example, double-click on the WebForm1.ASPX file. Right-click on the WebForm1.ASPX file and click View Code. We need to add a reference to the XML Web service that we referenced from the **gasTIX Web server**...
Page 348, first sentence of second paragraph: Select View Designer on the **WebForm1.ASPX** page.	Select View Designer on the **WebForm1.aspx** page.
Page 349, third bulleted item: • www.**gasTix**.com/Participant.asmx?SDL...	• www.**gasTIX**.com/Participant.asmx?SDL...
Page 369, bulleted list: • the database to use (dsn=gastix;uid=sa;**pwd"=**) • the table we are querying (event) • the column we are querying against **event_id** • the value we are using to query the database (mapped from the gasTIX **event_id.**)	• the database to use (dsn=gastix;uid=sa;**pwd=**) • the table we are querying (event) • the column we are querying against **(event_id)** • the value we are using to query the database (mapped from the gasTIX **event_id**)
Page 371, first line of the Note:	

The port, IncomingPrintOrder, referred in the dialog in Figure **12.14**...	The port, IncomingPrintOrder, referred in the dialog in Figure **12.13**...
Page 394, third bulleted item in the Note: • The Microsoft public news groups at msnews.**Microsoft**.com have several good discussions on BizTalk. Paths include **Microsoft**.public.biztalk.general, **Microsoft**.public.biztalk.orchestration, and **Microsoft**.public.biztalk.setup	• The Microsoft public newsgroups at msnews.**microsoft**.com have several good discussions on BizTalk. Paths include **microsoft**.public.biztalk.general, **microsoft**.public.biztalk.orchestration, and **microsoft**.public.biztalk.setup
Page 416, Figure 14.2 caption: *Define DEBUG **constant** for C# .NET project.*	*Define DEBUG **Conditional Compilation Constant** for C# .NET project.*
Page 457, two paragraphs under "Tuning the .NET Business Tier with Visual Studio Analyzer" heading: One of the main tools used to tune the middle tier in .NET applications is Visual Studio Analyzer (**VSA**). **VSA** is designed to provide a picture of the lower-level operation of a Windows .NET application. Unlike debugging and tuning within a single language such as Visual Basic .NET, **VSA** shows us interactions between components, machines, and processes. **VSA** captures data through events, and unwanted data can be filtered out. Graphical displays allow a technical architect to track down problems and bottlenecks. **VSA** operates within the familiar Visual Studio .NET IDE, and is made available with the Enterprise Edition of VS .NET.	One of the main tools used to tune the middle tier in .NET applications is Visual Studio Analyzer (**Vista**). **Vista** is designed to provide a picture of the lower-level operation of a Windows .NET application. Unlike debugging and tuning within a single language such as Visual Basic .NET, **Vista** shows us interactions between components, machines, and processes. **Vista** captures data through events, and unwanted data can be filtered out. Graphical displays allow a technical architect to track down problems and bottlenecks. **Vista** operates within the familiar Visual Studio .NET IDE, and is made available with the Enterprise Edition of VS .NET.
Page 458, first paragraph: Once the **VSA** Project Wizard has been completed, we are placed in the Visual Studio .NET IDE. A new **VSA** Analyzer solution is created, and we can explore...	Once the **Vista** Project Wizard has been completed, we are placed in the Visual Studio .NET IDE. A new **Vista** Analyzer solution is created, and we can explore...
Page 458, Figure 15.15 caption: *The Solution Explorer in **VSA** shows all the components of a VSA solution.*	*The Solution Explorer in **Vista** shows all the components of a VSA solution.*

Page 458, first line of Tip: It is a good idea to run our ACT test scripts while recording in **VSA**, as the user is...	It is a good idea to run our ACT test scripts while recording in **Vista**, as the user is...
Page 459, Figure 15.16 caption: *Recording a **VSA** test run while stressing the system with ACT.*	*Recording a **Vista** test run while stressing the system with ACT.*
Page 459, Figure 15.17 caption: *The **VSA** Event details view shows us all the events captured during a recording session.*	*The **Vista** Event details view shows us all the events captured during a recording session.*
Page 460, second line of Caution: ...managed code, and vice versa. Interop is an expensive operation, however, and **VSA** is a...	...managed code, and vice versa. Interop is an expensive operation, however, and **Vista** is a...
Page 460, paragraphs below the Caution: **VSA** provides several other helpful views, including the Machine Diagram (helpful for a distributed application with multiple servers), the Process Diagram, and the Component Diagram. Click the New Process Diagram button to automatically generate a view of all the processes captured by the recorder. A sample **VSA** process diagram is shown in Figure 15.18.	**Vista** provides several other helpful views, including the Machine Diagram (helpful for a distributed application with multiple servers), the Process Diagram, and the Component Diagram. Click the New Process Diagram button to automatically generate a view of all the processes captured by the recorder. A sample **Vista** process diagram is shown in Figure 15.18.
Page 461, second paragraph, second line: ...components captured by the recorder. A sample **VSA** component diagram is shown in Figure 15.19.	...components captured by the recorder. A sample **Vista** component diagram is shown in Figure 15.19.
Page 461, second paragraph after the figure, second line: shows how a .NET application is really executing under the covers. **VSA** is a great tool for...	shows how a .NET application is really executing under the covers. **Vista** is a great tool for...

This errata sheet is intended to provide updated technical information. Spelling and grammar misprints are updated during the reprint process, but are not listed on this errata sheet.