

Contents

<i>Foreword</i>	<i>xvii</i>
<i>About the Authors</i>	<i>xix</i>
<i>Acknowledgments</i>	<i>xxi</i>

Part One	Overview of ATL Server	
	Features and Services	<i>1</i>

Chapter 1	Introduction to ATL Server	<i>3</i>
------------------	---	----------

<i>What Is ATL Server?</i>	<i>3</i>
<i>Who ATL Server Is For</i>	<i>4</i>
<i>What About Web Services?</i>	<i>4</i>
<i>How ATL Server Compares with Traditional</i>	
<i>Solutions</i>	<i>5</i>
CGI	<i>6</i>
ASP	<i>6</i>
ISAPI	<i>6</i>
Other Solutions	<i>6</i>
ATL Server	<i>7</i>
<i>How ATL Server Works</i>	<i>8</i>
SRF Basics	<i>11</i>
Application Basics	<i>12</i>
ISAPI Basics	<i>14</i>
Other Features of ATL Server	<i>15</i>
<i>More on Performance</i>	<i>17</i>
<i>Basic Deployment</i>	<i>18</i>
<i>Debugging Web Applications</i>	<i>19</i>
<i>Attribute Basics</i>	<i>19</i>
<i>Conclusion</i>	<i>20</i>

Chapter 2	SRF Files	<i>21</i>
------------------	------------------------	-----------

<i>Flow Control</i>	<i>22</i>
If/Else Statement	<i>23</i>
While Statement	<i>25</i>
Nesting Statements	<i>27</i>

<i>Using More Than One Handler</i>	27
<i>Include Statements</i>	29
<i>Parameters</i>	29
<i>Why No Variables?</i>	31
Advantages of ATL Server	31
<i>Comment Tags</i>	32
<i>Conclusion</i>	32
 Chapter 3 Request Handlers	33
 <i>Creating a Request Handler</i>	33
<i>Using Attributes</i>	36
<i>Web Jukebox 1.0</i>	38
Getting Started	38
<i>Conclusion</i>	43
 Chapter 4 Introduction to ISAPI Services	45
 <i>What to Put in Your ISAPI DLL</i>	45
<i>Designing Your ISAPI Services</i>	46
<i>Creating an ISAPI Service</i>	47
Creating a Simple Counter	48
Exposing the Counter As a Service (from the ISAPI DLL)	49
Using the ISAPI Service	51
<i>Conclusion</i>	53
 Chapter 5 Inside the ATL Server Architecture	55
 <i>Key Components in ATL Server</i>	55
<i>Processing an Incoming Request</i>	57
<i>Key ATL Server Classes</i>	59
Key ISAPI Classes	59
Key Request Handler (Application DLL) Classes	60
<i>Key Classes in Web Service Requests</i>	62
Key Web Service Client Classes	62
Key Web Service Server Classes	63
<i>Conclusion</i>	64

Chapter 6	Getting User Input	65
	Query Strings and Form Variables	65
	The ValidateAndExchange Method	68
	Validating Input	72
	Exchanging Data Types	74
	Using a Validation Context	76
	Conclusion	82
Chapter 7	Deploying an ATL Server Web Application	83
	Understanding the Administration Tool	83
	Virtual Directory Tab	84
	Other Tabs	85
	Modifying Your ISAPI DLL	87
	Understanding the Visual Studio .NET Deployment Utility	88
	Conclusion	91
Chapter 8	Cookies and Session State	93
	Session State Service	93
	Using Session State	96
	Using Database-Backed Session State	99
	Conclusion	103
Chapter 9	Perfmon	105
	Using Perfmon	106
	Out-of-the-Box Counters	107
	Understanding the Perfmon System	107
	Creating a Perfmon Counter	108
	Exposing a Perfmon Counter As an ISAPI Service	111
	Some Other Types of Perfmon Counters	116
	Conclusion	118
Chapter 10	Web Services	119
	Introducing Web Services	119
	SOAP	120
	WSDL	121

<i>Creating a Web Service</i>	123
Creating a Web Service by Hand	124
Creating a Web Service with ATL Server	124
<i>Using ATL Server Web Services</i>	128
ATL Server Web Service Architecture	129
SAX	129
Implementing Web Services	130
<i>Conclusion</i>	157

Part Two Advanced Concepts and Techniques

159

Chapter 11 Using ATL Server Components in Stand-Alone Applications

161

<i>Reusing the ATL Server Thread Pool</i>	161
General Considerations	162
The ATL Server Thread Pool Class	162
ATL Server Thread Pool Class Sample	164
Results and Conclusions	167
<i>Reusing the ATL Server Stencil Processor</i>	168
How the ATL Server Stencil Processor Works	169
Stencil Processor Sample	172
<i>Conclusion</i>	175

Chapter 12 Caching

177

<i>Types of Caches</i>	177
CMemoryCacheBase	177
CMemoryCache	177
CBlobCache	178
CDllCache	178
CStencilCache	178
CFileCache	178
CDataSourceCache	178
<i>When to Use Caching</i>	179
Caching Support Example	179
Key Types and Data Types	184
Cache Parameters	195
Cache Statistics	195
<i>Conclusion</i>	196

Chapter 13	<i>Application Tuning: Real World Tips and Tricks</i>	197
<i>COM</i>		197
OLE Automation Types		198
Marshaling		198
Security and Process Model		199
COM+		199
<i>Databases</i>		200
<i>Using the Right Tools</i>		201
Profiler		202
Perfmon		203
Task Manager		203
Web Stress Tool		203
Putting It All Together		204
<i>Common Symptoms and the Problems That Cause Them</i>		205
Low CPU, Low Requests per Second		205
High CPU, Low Requests per Second		205
Low Requests per Second, High Network Usage		205
Low Requests per Second, High Database Usage		206
Low CPU, High Requests per Second		206
<i>Conclusion</i>		206
 Chapter 14	 <i>ISAPI Extensions As Service Providers</i>	 209
<i>A Simple ISAPI Service</i>		209
<i>Built-in Global ISAPI Services</i>		213
<i>Built-in per-Thread ISAPI Services</i>		215
Using per-Thread Services		215
How to Implement the Actual Service		216
How to Make a Service per-Thread		217
Optimization		218
Per-Thread ISAPI Services Conclusion		219
<i>Dynamically Adding Services to the ISAPI Application</i>		219
<i>Conclusion</i>		221

Chapter 15	<i>Debugging ATL Server Applications</i>	223
	<i>Server-Side Debugging</i>	223
	ISAPI Initialization	223
	Request Loading and Processing	227
	Finding Errors in SRF Files	230
	Identifying a Failing Replacement Method	232
	<i>Client-Side SOAP Debugging</i>	234
	<i>Debugging Helper: The ISAPI WebDbg Tool</i>	236
	<i>Conclusion</i>	236
Chapter 16	<i>Database Access with ATL Server</i>	239
	<i>Using OLE DB Consumer Templates</i>	239
	<i>Using the ATL Server Data Source Cache</i>	247
	<i>Using the Data Source Cache Service</i>	251
	<i>Putting It All Together</i>	251
	<i>Conclusion</i>	259
Chapter 17	<i>Advanced SRF</i>	261
	<i>Extending Stencil Syntax</i>	261
	<i>Overriding File Extensions</i>	272
	<i>Conclusion</i>	272
Chapter 18	<i>Advanced Request Handlers</i>	273
	<i>Creating a Thread Pool</i>	273
	<i>Runtime ATL Server Information</i>	275
	<i>Dynamically Sizing a Thread Pool</i>	277
	<i>Conclusion</i>	285
Chapter 19	<i>Advanced Web Services</i>	287
	<i>ATL Server and COM</i>	287
	COM Attributes	287
	User-Defined Types	293
	IDispatch Types	297
	Allocation	305

<i>Determining Whether an Object Is Invoked As COM or As SOAP</i>	305
SOAP Faults and IErrorInfo	309
<i>ATL Server Web Service Internals</i>	312
How an ATL Server Web Service Works	313
<i>ATL Server Implementation</i>	314
Error Checking	336
Structs and Enums	337
<i>Arrays</i>	344
<i>WSDL Generation</i>	345
<i>Conclusion</i>	345

Chapter 20 Performance Tuning ATL Server Web Applications

<i>What Performance Is</i>	347
<i>A Typical Web Request from a Performance Perspective</i>	348
<i>ATL Server Facilities</i>	350
<i>Pitfalls and Programming Idioms to Avoid</i>	350
Excessive Memory Allocation	351
Locks	351
I/O	352
<i>Measuring Performance</i>	353
<i>Conclusion</i>	355

Chapter 21 ATL Server and Passport .NET

<i>Using Passport</i>	357
<i>Initializing Passport Manager</i>	359
<i>Using Passport to Authenticate</i>	365
IsAuthenticated	368
<i>Displaying User-Specific Content</i>	369
<i>Putting It All Together</i>	371
<i>Using CPassportHandlerT</i>	376
<i>Conclusion</i>	380

Chapter 22 Interoperability with ATL Server Web Services	381
<i>What Interoperability Is</i>	<i>381</i>
<i>When and Why You Should Care About Interoperability</i>	<i>383</i>
<i>ATL Server Interoperability Limitations</i>	<i>383</i>
Standard Basic Data Types Not Supported by ATL Server	383
WSDL Problems	386
A Specific Interoperability Problem	389
<i>Conclusion</i>	<i>391</i>
 Chapter 23 Extensibility of ATL Server Web Services	 393
<i>ATL Server SOAP Request Processing Model</i>	<i>393</i>
<i>Using a Different Transport Protocol (Non-HTTP)</i>	<i>395</i>
Overriding the HTTP Transport	396
ReadClient/WriteClient	397
GetServerVariable	398
Mapping an HTTP Request Object to a Physical Object	398
<i>Custom Parsing on the Server Side</i>	<i>401</i>
<i>Custom Data Types Marshaling</i>	<i>411</i>
<i>Conclusion</i>	<i>419</i>
 Chapter 24 SMTP and MIME Support in ATL Server	 421
<i>Connecting to an SMTP Server</i>	<i>421</i>
<i>Managing Connections with CSMTPConnection</i>	<i>422</i>
<i>Sending MIME Messages</i>	<i>423</i>
CMimeMessage	424
Sending Text	427
Adding Attachments	428
<i>Extending ATL Server's SMTP Support:</i>	
<i>Sending HTML Messages</i>	<i>430</i>
<i>Tips for Improving Performance</i>	<i>436</i>
Creating a Delivery Queue	436
Processing the Delivery Queue	438
<i>Conclusion</i>	<i>440</i>

Chapter 25	Developing HTTP Client Applications in ATL Server	441
Classes for Developing HTTP Clients		442
A Simple HTTP Client		442
Transmitting and Receiving Data		447
Handling Query and Form Parameters		447
Sending and Receiving Large Payloads		451
Securing the HTTP Communication		453
Authentication		453
Secure Communication (HTTPS)		454
Conclusion		455
Chapter 26	Securing Your Web Application	457
Security Best Practices		457
Authentication		459
NTLM Authentication		460
Basic Authentication		461
Encryption		464
Conclusion		466
Chapter 27	ATL Server FAQ	467
Generic ATL Server Application FAQ		467
What Does “Assertion in AtIsapi.h, Line 663” Mean?		467
How Can I Access the Physical Directory of an ATL Server Application?		468
How Can I Access the Client’s SSL Certificate in a Secure Web Application?		468
Mail Support FAQ		469
How Do I Send HTML Mail Messages with ATL Server?		470
SOAP FAQ		470
How Do I Make a SOAP Client Point to a Different URL?		470
How Do I Use an HTTP Proxy in a SOAP Client?		471
How Do I Use SOAP over a Custom Transport Protocol?		471
How Do I Use WinINet, ServerXMLHTTP, or a Custom HTTP Stack in a SOAP Client?		471
How Do I Use HTTPS/SSL in a SOAP Client?		472
How Do I Select a Specific Certificate for an SSL Connection?		473
How Do I Access the Client’s Certificate in a SOAP Server?		474

How Do I Use Custom Encryption/Compression with SOAP?	474
How Do I Develop an Asynchronous SOAP Client?	475
How Do I Send XML Fragments/Unescaped Strings As Part of a SOAP Message?	476
How Do I Perform Custom Server-Side Parsing?	477
<i>Conclusion</i>	477
 <i>Index</i>	 479

ATL Server

High Performance C++ on .NET

Kumar, P.; Lee, E.; Grewal, J.S.; Crivat, B.

2003, XXI, 520 p. 21 illus., Softcover

ISBN: 978-1-59059-128-4

A product of Apress