

Symbols

#import 150
#pragma managed 144
#pragma unmanaged 161, 189
% 26
& 26–27
* 12, 27
+= 59
-= 60
^ 13, 24, 27, 69

A

abstract functions 106
abstract methods 21
abstract modifier 20–21
abstract sealed classes 22
accessor functions 47, 49, 67
accessors, compiler-generated code 49
ActivateDesktop 200
Activator::CreateInstance 115, 321
adapter pattern 118
AddRef 152
AddServiceEndpoint 337
afxdll.h 312
AfxOleInit 316
afxwinforms.h 236, 240, 251, 265
afxwin.h 312

Ansi strings, converting to and from 181
AppDomains 382
Application class 234, 290, 293, 299
 OnStartup 296
 OnStartup method 293
application domains. *See* AppDomains
Application.LoadComponent 285
args 16
array covariance 80
ArrayList 11
arrays
 basic syntax 69
 direct access 83
 for each optimization 78
 function arguments 74
 marshalling 184
 of non-CLI objects 81–83
 returning from functions 75
 simulating a static array 185
 using with a native pointer 83
 using with for loop 78
arrays of native types, avoiding memory leaks 82
assembly 372
assignment operators 36–38
 defining 36
AsyncCallback 63
auto_groot 165–166

Avalon. *See* Windows Presentation Foundation

B

BAML 284
Base Class Library 5–6, 377
BasicHttpBinding 336, 364
BCL. *See* Base Class Library
BEGIN_DELEGATE_MAP 237, 244, 255, 265, 319
BEGIN_INTERFACE_MAP 357
BeginInvoke 62–63
bindings 336
box instruction 41
 _box keyword 39
boxed value type, assigning null 43
boxing 11, 30, 38
 cost 39, 43
 differences between MC++ and C++/CLI 39
 IL implementation 41
 lower priority in overloads 44
 overuse in loops 44
 type-safety 40
Bray, Brandon 8, 185
Brushes 301
BSTR 182
BuildWindowCore 327

C

-
- C# 7–8
 - unsafe mode 376
 - C# 2.0 12
 - C++ interop 154
 - accessing a native DLL 159
 - comparison with CCW 155
 - comparison with P/Invoke 161
 - C++, managed extensions 8
 - C++/CLI
 - accessing the .NET library 7
 - comparison with old syntax 10
 - Hello World 13
 - instantiating classes 28
 - interop features 7
 - interop mechanisms 134
 - leveraging managed frameworks 7
 - no default copy constructor 33
 - rational behind the syntax 8
 - reasons for using 6
 - reusing existing native code 6
 - role of 4
 - writing managed applications 8
 - C++/CLI, class modifiers 20
 - C++/CLI, constructors 31
 - C3675 50
 - callback function 173, 205
 - CallbackContract 340
 - Canvas 282, 293, 319, 330
 - Background property 282
 - castclass 42
 - CAutoNativePtr 168–169
 - Detach, Attach and Destroy methods 170
 - example usage 171, 209
 - CAvalonView 320, 324
 - CCmdTarget 349
 - CComPtr 351
 - CComSafeArray 153
 - .cctor 31
 - CCW, preserving method signature 153
 - See also* COM Callable Wrappers
 - CDocument 243
 - CEditView 249–250
 - ChannelFactory 338
 - CharSet::Auto 204
 - Cider 280, 297
 - class wrappers, derived 223
 - classes, visibility rules 18
 - CLI 11
 - arrays 69
 - types 22, 380
 - cli namespace 69
 - CLI wrapper 180, 190, 193
 - using from C# 197
 - Click event 264
 - CloseDesktop 195, 199
 - /clr compiler switch 15
 - CLR heap 90, 135, 137–138, 141, 143, 146
 - CLR. *See* Common Language Runtime
 - CLR types, declaring 18
 - /clr:oldSyntax 13, 15
 - /clr:pure 15, 161
 - /clr:safe 13, 15
 - __clrcall calling convention 188
 - CLS compliance 372
 - CLS. *See* Common Language Specification
 - CLS-compliant language 379
 - CoCreateInstance 351
 - CodeDOM 287
 - COM Callable Wrappers 147
 - accessing a CLI library 149
 - COM strings, converting to and from 182
 - Common Language Infrastructure (CLI) 4–5
 - Common Language Runtime 5, 369, 371
 - components 373
 - Common Language Specification 6, 372
 - Common Type System 6, 372
 - compilation modes 15
 - compiler error
 - C3265 163
 - C3628 23
 - C3673 33
 - C3675 50
 - C4368 166
 - C4693 22
 - ComputeHash 223
 - ComVisible attribute 150
 - Console class 14
 - Console::WriteLine 77
 - constraint mechanism 113, 116
 - Control class 234
 - copy constructors 33–34
 - overloading 35
 - copy-assignment operator 36
 - CreateChannel 338
 - CreateDesktop 195, 199
 - CreateManagedControl 237, 252
 - CreateProcess 200–201
 - CSecureStdioFile 208
 - CStdioFile 207
 - CString 183
 - CTime 269
 - .ctor 31
 - CTP 280
 - CTS. *See* Common Type System
 - Current property, enumerating an array in MC++ 11
 - custom RCW 212, 218
 - CustomBinding 336
 - CWinApp::LoadIcon 254
 - CWinFormsControl 235–236, 249, 251, 266
 - CWinFormsDialog 261, 265
 - CWinFormsView 240–242, 323
 - CWnd 268
-
- D**

 - DataGridView 240, 242–243
 - bind XML to the 245
 - CellValueChanged event 244
 - Dock property 242
 - UserAddedRow event 244
 - UserDeletedRow event 244

DataSet 246
 DataSource property 246
 DCOM 333, 362
 callback interface 353
 migrating to WCF 344
 server 346, 352
 DDX_ManagedControl 236
 deep copy 34
 default index properties 57
 keyword 57
 Delegate class
 Combine method 61
 Remove method 61
 delegate_proxy_factory 237
 delegates 58–59, 173
 asynchronous invocation
 61–62
 invocation list, order 59–60
 invoking synchronously 61
 setting up a delegate 59
 use in Windows Forms
 classes 58
 delete 89, 96, 98
 derived class wrappers 223
 DesignMode property 271–
 272
 desktop API, CLI wrapper for
 the 191
 DestroyWindowCore 328
 destructor 88
 semantics 88
 destructors 101
 IL for 94
 deterministic destruction 8,
 82, 87
 support for 12
 DiLascia, Paul 218
 Directory class 341
 DirectX 278
 Dispose 95, 101
 Dispose pattern 87, 95, 381
 DllImport 157, 159, 161
 DllMain 312
 DoDataExchange 236
 double thinking 186
 duplex communication 338
 DuplexChannelFactory 344

E

ECMA 5, 8, 372
 background 9
 Ellipse class 302
 EnableVisualStyles 234
 END_DELEGATE_MAP 237
 EndInvoke 62–63
 EndPoint property 301
 EnumDesktops 196
 enumerations 380
 EnumWindows 173, 175
 EnumWindowTitles 205
 event handlers, hooking to
 objects 58
 event keyword 64
 event, invoking 58
 EventArgs 65, 68
 derived classes 66
 EVENT_DELEGATE_ENTRY
 237
 events 58, 64
 invoking 58
 nontrivial example 66
 use in Windows Forms
 classes 58
 explicit overriding 103

F

fault handler 98
 fault-block 98
 Fibonacci numbers 75
 Fill property 302, 304
 Finalization 94
 Finalize method 95
 finalizer 88, 168, 381
 semantics 88
 finalizers 101
 IL for 94
 finally handler 98
 for each 11, 78
 Forms designer, issue with
 MFC 267
 forwardref 61
 FrameworkElement 304, 324,
 327–328
 F-Reachable queue 94
 FreeBSTR 182

FreeHGlobal 181
 _FUNC_SIG__ macro 188
 function adaptors 58
 function pointers 173
 functions
 abstract 107
 overriding 102
 sealed 107

G

garbage collection 90–94, 144
 cycle 91, 93, 102, 137, 145
 generations 93
 pNextObject 90
 reachable objects 91
 root 91
 Garbage Collector 6, 135, 141,
 381
 GC hole 147
 __gc pointer 24
 GCHandle 163, 204
 gnew 13, 20, 28, 30
 constraint 115
 gcroot 163–164, 219, 223, 313,
 324, 354
 -> operator 163
 gcroot.h 163
 generic 20
 generic keyword 110
 generics 108
 adapter pattern with 118
 comparison with templates
 120–124
 global functions, and 112
 mixing with templates 127
 nongeneric types with 112
 stack example 109
 syntax 110–113
 Get method 72
 GetCallbackChannel 341
 GetCurrentThreadId 202
 GetDelegateForFunction-
 Pointer 173, 175
 trick to use assembler using
 177
 GetFunctionPointerFor-
 Delegate 173, 196, 202,
 206

GetThreadDesktop 195
 GetType method 383
 GetUserObjectInformation
 195
 GetWindowText 329

H

Handle 234
 handles 13, 24
 behavioral differences from
 pointers 25
 syntax 24
 HDESK 193
 _hDesktop 198
 heap fragmentation 84, 144
 Hello World 13
 HwndHost 326, 328
 HwndSource 314, 322, 325–
 326, 330–331
 HwndSource::Handle 314
 HwndSourceHook 329

I

IChannel 338
 ICollection 78
 IComponentConnector 285
 IDisposable 88, 95
 IEnumerable 78
 IEnumerator 78
 IL Disassembler 31
 ildasm.exe 31
 implicit boxing 38
 indexed properties 47, 55–56
 indexers. *See* default index
 properties
 Indigo. *See* WCF
 inheritance 22–23
 InitializeComponent 284
 initobj 30
 instance constructors 31
 InstanceContext 344
 interface methods 19
 interior pointers 135–140
 passing by reference 137
 pointer arithmetic with 138
 interior_ptr 136, 186
 interop 147, 155, 232

interop mechanisms 190
 InteropServices namespace
 181
 IntPtr 185
 intptr_t 220, 313
 InvalidCastException 40
 InvalidOperationException
 335
 Invoke method 61
 IPersistFile 213
 IPrimeNumber 346
 IPrimeNumberCallback 346–
 347
 IPrimeNumberManCallback
 356
 IsExplicitlyDereferenced 139
 IShellLink 213
 IsOneWay 340
 IUnknown 349

J

jagged arrays 71–72
 doubly 73
 typedef with 73
 JIT compiler 6, 372, 375

K

KillTimer 270

L

large object heap 93
 ldelem 70
 LinearGradientBrush 301,
 312
 linker settings, SUBSYSTEM
 233
 LoadBaml 286
 LoadedBehavior 312
 LocalSystem account 361
 l-value 26

M

main method 16
 MAKE_DELEGATE 238, 244,
 252–253, 259, 266, 319

MAKE_DELEGATE macro
 326
 _MANAGED 220
 managed arrays 68
 System::Array 69
 Managed C++ 5, 11
 confusing new usage 12
 lack of acceptance 8
 poor integration with .NET
 12
 problems with 9–13
 twisted syntax and grammar
 in 9
 underscored keywords 10
 managed extensions 8
 managed library, accessing
 from native code 148
 managed objects, instantiating
 13
 managed templates 8, 108,
 124
 ManualResetEvent 343
 Marshal class 173–174, 176
 Marshal::Copy 184–185
 Marshal::GetException-
 ForHR 217
 Mask property 235
 MaskedTextBox 235
 MaskInputRejected event
 238–239
 MaskInputRejectedE-
 ventHandler 238
 MD5CryptoServiceProvider
 223
 MediaElement 311–312, 315
 MediaEnded 317
 memory leaks 81
 MenuStrip 249, 253, 257
 metadata 374, 383
 MethodImplAttribute
 attribute 153, 156
 MFC control, using in a Win-
 dows Forms form 267
 MFC DLL wrapper 208
 MFC extension DLL 206
 MFC strings, converting to
 and from 183
 Microsoft Intermediate Lan-
 guage 6

Miller, Shaun 185
 mixed mode DLL, exporting
 managed and native
 interfaces 218
 mixed types 8, 162
 managed types with native
 members 166
 native types with managed
 members 162
 mixed-mode assembly 376
 mixed-mode programming
 134
 mix-in classes 123
 mix-ins 126
 modopt 139
 Mono 370
 MoveNext enumerating an
 array in MC++ 11
 msclr::com::ptr 215–217
 mscore.dll 374
 MSDN Magazine 218
 MSIL 287, 372
 MSIL verification 375
 MulticastDelegate 59, 61
 multidimensional arrays 71
 multiple inheritance 24
 multiple overriding 105

N

native API, overview 191
 native library, accessing from
 managed code 156
 native pointers 135
 native strings, marshalling
 181
 .NET 4–5
 decompilers 373
 executing a program 374
 Visual C++ and 375
 .NET Framework 369
 hierarchy of layers 370
 Microsoft products that use
 the 371
 .NET Framework 3.0 279
 NetNamedPipeBinding 336
 NetPeerTcpBinding 336
 NetTcpBinding 336
 new 103

New Project Wizard 16
 newarr 70
 newobj 12, 29, 72
 nonboxing cast 44
 nonrectangular arrays 72
 nontrivial event 67
 nontype parameters 122, 126
 nullptr 20, 30, 43, 68

O

object resurrection 94
 Office 2003 style UI 249
 ON_CBN_SELCHANGE
 316
 OnClose 247, 270
 OnCreate 250
 OnHandleCreated 271
 OnInitDialog 265–266, 317
 OnMediaEnded 313
 OnPaint 269
 OnStop 361
 ON_UPDATE_COMMAND_
 UI 259
 OpenDesktop 199
 OperationContext::Current
 341
 OperationContract 335, 338,
 340
 OutOfMemoryException 28
 overloads 44
 override 53, 103–104

P

Page 282
 ParamArrayAttribute 76
 parameter arrays 76
 parameterized types 108
 partial classes 284, 286
 Petzold, Charles 277
 pinning 204
 pinning pointers 83, 135, 141–
 147
 passing to native code 143
 recommendations for usage
 146
 pin_ptr 83, 142, 184
 P/Invoke 157, 161

pointers 83, 135
 behavioral differences from
 handles 25
 confusing usage 12
 polymorphic 34
 Portable Executable 373
 PostMessage 256
 PreCreateWindow 251
 PreserveSig 156
 PreSubclassWindow 270
 PreTranslateMessage 248, 260
 printf 175
 private inheritance 23
 properties 47
 analogy 48
 inheritance and 51
 virtual 54
 protected inheritance 23
 PtrToStringChars 139, 144,
 199, 210
 public fields, break encapsula-
 tion 51
 public inheritance 23
 punctuator 24

Q

QueryInterface 152

R

RadialFill 304
 RadialGradientBrush 304
 RAI1 87, 101
 RaiseWindowFound 205
 reachable objects 91
 ref types 105
 reference types 32
 Reflection 383
 Regasm.exe 150
 renamed overriding 104
 RenderTransform property
 304
 RepositionBars 258
 RootVisual property 314
 RotateTransform 305–306

S

SAFEARRAY 153
 SafeArrayDestroy 153
 safe_cast 40, 43, 53
 scalar properties 47–48
 ScaleTransform 306
 sealed classes 21
 sealed functions 106
 sealed modifier 20–21
 self-assignment 36
 SendMessage 257
 ServiceBase 361
 ServiceContract 335
 ServiceHost 361
 ServiceHost class 337
 Set method 72
 SetItemCount 350
 SetThreadDesktop 195, 202
 SetWindowText 330
 shallow copy 34
 Shared Source CLI 370
 single-dimensional arrays 70
 sizeof 220
 SizeToContent 322
 SkewTransform 308
 smart pointer 167, 217
 SolidColorBrush 301
 spaced keywords 20
 stack semantics 8, 27, 34, 82, 87, 96
 guidelines for using 101
 handling member objects 99
 implementation 98
 support for 12
 unsupported types 96
 StartPoint property 301
 STAThread attribute 290
 static constructors 31
 static properties 55
 stelem 70
 STL strings, converting to and from 183
 STL.NET 8, 129
 StreamReader 97
 StreamWriter 97
 string interning 140
 String::IsInterned 141

StringBuilder 159
 StringToBSTR 182
 StringToHGlobalAnsi 181, 183
 StringToHGlobalUni 182
 Stroke property 303
 struct vs. class 18
 StructLayout 186
 structs, visibility rules 18
 SuppressFinalize 95, 102
 SVC file 362–364
 svcutil.exe 365
 System namespace 14
 System::Array 69–70
 methods of 77
 System::Console 14
 System::Object 24, 116, 221
 wrapping 220
 System::OutOfMemoryException 29
 System::String 140
 BSTR conversion 182
 converting from and to char* 181
 converting from and to wchar_t* 182
 CString conversion 183
 System::ValueType 23
 System::Windows::Controls namespace 292
 SystemArrayLength property 73
 SystemArrayRank property 73
 System.Object 380, 383
 System.Reflection.Emit namespace 384
 System.ServiceModel 334, 339, 352, 356
 System.ValueType 380

T

template specialization 125
 templates, support for 12
 TextBlock 282
 TextBox 235
 this pointer 139
 tlbimp.exe 212
 ToolStrip 249, 254–255, 257

ToString 225
 tracking 25
 tracking reference 26, 74
 transformations 304
 combining multiple transformations 308
 TransformGroup 306
 TranslateMessage 248
 trivial events 65
 trivial properties 50
 compiler generated code 50
 inlined by JIT 51
 _try_cast 40
 try-finally 98
 type conversions 181
 type library importer. *See* tlbimp.exe
 typeid keyword 120
 typename keyword 110

U

UIElement 302
 unary % operator 27
 unboxing 38, 43
 cost 39, 43
 Unicode strings, converting to and from 181
 Universal Resource Identifier 334
 UnmanagedFunctionPointer attribute 204
 UnsafeValueType 186
 unverifiable code 13
 UserControl 241, 297

V

value types 23, 32
 default constructor and 32
 VB.NET 7–8
 VC++ 2002 5
 VC++ 2003 5
 vector 349
 verifiable code 13
 Virtual Execution System (VES) 6, 372
 virtual property 51
 get and set methods 53

Visual C++ 2005 4
 creating a CLR project 16
 New Project Wizard 17
 setting compiler options 18
 support for C++/CLI standard 9
 Visual Studio Orcas 280

W

WaitTillSafe 343–344
 WCF 333
 adding support to an MFC client 356
 callback 358
 client 342
 comparison of migration techniques 362
 IIS hosting 362
 proxy 344, 351
 service 334, 356, 358–359
 creating 340
 writing a simple application 334
 web.config 364
 Window class 291
 WindowInteropHelper 329, 331
 Windows Communication Foundation 8
See also WCF

Windows Forms 8, 232
 hosting as an MFC view 239
 hosting in an MFC dialog 235
 simple app 233
 using as an MFC dialog 261
 Windows Presentation Foundation 7, 277–278
 Windows Vista 7
 Windows Vista SDK 281
 WinForms. *See* Windows Forms
 WinFX. *See* .NET Framework 3.0
 WM_CLOSE 247
 WM_COMMAND 256–257
 WM_COPYDATA 326, 330
 WM_ERASEBKGD 323
 WM_SIZE 258
 WM_SYSKEYDOWN 260
 WPF
 anatomy of a simple application 283
 control library 311
 creating a C++/CLI project 289
 hosting a native control 326
 Image control 324
 native interop 310
 Similarity with MFC 290
 using a mixed mode app 319

using brushes and shapes 300
 using procedural code 289
See also Windows Presentation Foundation
 WPF with C++
 comparison of various techniques 300
 deriving a class from a C# project 297
 wrapper class, implementation 198
 WSDL 335
 WSDualHttpBinding 336, 342, 344
 WSHttpBinding 336

X

x:Class attribute 284
 x:FieldModifier attribute 298
 x:Subclass attribute 287
 XAML 280, 299
 C++ and 287
 compilation process 284
 dynamically loading 294
 use of partial classes 284
 XAMLPad 280
 .XamlReader 296