



# Index

---

## A

- AbsoluteTime expiration policies, 194
- Abstraction, in custom business entities, 156
- Access control list (ACL)-based authorization, 420, 495–496
- ActiveDirectoryRolesProvider, 478–479
- ActivityIDs
  - adding tracing with, 370–375, 393–394
  - Categories and, 375
  - tracing and logging with, 375–376
- ADCustomAuthenticationProvider-Data, 409–412
- Add method
  - adding items to cache, 229–230
  - adding items to cache with RefreshAction, 231
  - IContainer interface, 89
- AddActivityIdToLogEntry method, 347–348
- AdditionalInfo property, ExceptionFormatter, 262
- ADO.NET, 657
- AdRolesProvider, 445
- Advanced customized exception handling, 300–301
- Advanced Encryption Standard, 511
- AppConfigFileStorageProvider, 15–16, 68–70
- Application blocks, building, 525–577
  - adding
    - ConfigurationDesignManager attribute, 573–574
  - core functionality, 529–530
  - creating configuration nodes, 557–567
  - design-time configuration, 556–557
  - factories, 540–543
  - implementing
    - ConfigurationDesignManager, 567–573
  - pluggable providers, 530–540
  - Quick Start application, 576
  - runtime configuration, 543–556
  - unit tests, 574–576
  - vision and design goals, 526–528
- Application groups, Authorization Manager, 434
- ApplicationException class hierachy, 245–246
- AppMessageExceptionHandler, 277–278, 298–300
- Architecture for .NET: Designing Applications and Services*, 113
- ASP.NET cache, 180–181, 184

## 662 ■ Index

- AssemblyInfo file
    - adding ConfigurationDesign-Manager attribute, 573–574
    - configuring new BackingStore, 218–219
    - design-time example configuration, 105–106
  - Asymmetric algorithms, 490–491
  - Asynchronous pull loading, cache items, 235
  - Attributes property
    - custom event handlers, 298–300
    - custom exception handlers, 277
    - custom StorageProviders, 74–76
    - custom Transformers, 43
  - Audit identifier, 436
  - Authenticate method, 467
  - Authentication, 402–412
    - AuthenticationProviderFactory and IAuthenticationProvider, 403–405
    - caching security credentials. *See* Security Cache
    - configuring, 464–467
    - custom AuthenticationProvider, 406–412
    - DbAuthenticationProvider, 405–406
    - digital certificates and, 488
    - NamePasswordCredentials, 406
    - overview of, 402–403
  - AuthenticationFactory, 465
  - AuthenticationProvider, 460, 464–467
  - AuthenticationProviderFactory, 403–405, 465
  - Authorization, 420–442
    - Authorization Manager (AsMan), 429–434
    - AuthorizationProviderFactory and IAuthorizationProvider, 423–425
    - AuthorizationRuleProvider, 425–426
    - configuring, 474–477
    - developing against Authorization Manager, 435–437
    - extending AuthorizationProvider, 437–442
    - overview of, 420–423
    - rule expressions, 426–429
  - Authorization and Profile
    - Application Block, 399–400
  - Authorization function name, defined, 436
  - Authorization Manager. *See* AzMan (Authorization Manager)
  - AuthorizationProviderFactory, 423–425
  - AuthorizationProviderPermission, 439–441
  - AuthorizationProviders
    - configuring, 474–477
    - extending, 437–442
  - AuthorizationRuleProvider
    - Authorization Manager vs., 429–430
    - configuring, 474–477
    - finite state machine and, 429
    - imperative security checks using, 437–442
    - overview of, 425–426
    - rule expressions and, 426–429
  - Authorize method
    - calling, 476–477
    - defined, 425
    - imperative security checks with, 438
  - AzManAuthorizationManager
    - configuring, 474–477
    - developing against, 435–437
    - overview of, 430–432
    - terminology used in, 432–433
  - AzManAuthorizationProvider, 435–442
  - AzManProvider, 474–477
  - AzManRolesProvider, 446–448
- B**
- BackgroundScheduler
    - design, 185
    - expiration process, 191–192

- overview of, 190–191
  - scavenging process, 239–240
  - BackingStores
    - adding items to cache, 229
    - cache storage classes for, 201–203
    - as Caching Application Block feature, 181
    - configuring, 223–225
    - custom, configuring, 225–226
    - custom, creating, 208–219
    - encrypting, 219, 226–228
    - overview of, 188
  - BaseApplicationException, 246–247
  - BaseBackingStore
    - creating custom CacheStorage, 208–219
    - defined, 190
    - implementing CacheStorage, 201–203
  - Binary data, 40–41
  - BinaryTransformer, 40–44, 78–79
  - Block ciphers, 490
  - BooleanExpression, 426–429
  - BuildContext method, 98–99
  - Byte arrays, 522, 523
- C**
- Cache objects, 189–191
  - Cache stores, 223–226
  - CacheCallback (CacheRefreshAction), 591
  - CacheExpiration, 591
  - CacheExpirationPolicies, 591
  - CacheFactory, 188–189, 229
  - CacheItemExpiration, 595, 604
  - CacheItemPriority, 595
  - CacheItemRefreshAction
    - adding items to cache, 230
    - caching in Data Mapping Application Block, 591–592
    - DataSetMappingProvider and, 595
    - enabling caching, 604, 607
  - CacheItemRemovedCallback, 198–200, 231
  - CacheManager
    - configuring, 221–223
    - data cache storage and, 206–208
    - forwarding requests to Cache object, 189
    - obtaining, 229
    - overview of, 183–189
    - setting, 605
    - storing credentials in, 416
  - CacheManagerFactory, 185–189, 229
  - CacheRefreshAction (CacheCallback), 591
  - CacheService objects, 190
  - CacheSettings
    - for Data Mapping Application Block, 591
    - DataSetMappingProvider and, 595
    - enabling caching with, 605
  - CacheStorage, 190, 201–203
  - Caching Application Block
    - defining, 178–179
    - previous version of, 182
    - using ASP.NET cache vs., 179–181
  - Caching Application Block, design, 183–220
    - BackgroundScheduler, 190–191
    - Cache objects, 189–191
    - CacheFactory, 183–189
    - CacheItemRemovedCallback, 198–200
    - CacheManager, 183–189
    - CacheManagerFactory, 183–189
    - CacheService objects, 190
    - CacheStorage, 201–203
    - custom cache storage, 208–219
    - data cache storage, 206–208
    - disk-resident cache, 204–205
    - encrypting cached data, 219–220
    - expiration policies, 191–198
    - ExpirationPollTimer, 191
    - ICacheItemRefreshAction interface, 198–200
    - isolated storage, 205–206
    - memory-resident cache, 204
    - overview of, 183

- Caching Application Block,
  - developing with
  - cache stores, 223–226
  - CacheManagers, 221–223
  - encrypting cached data, 226–228
  - overview of, 220–221
- Caching Application Block, using
  - API, 229–240
  - adding items to cache, 229–230
  - expiration policies, 230–231
  - flushing cache, 238–240
  - loading items in cache, 231–236
  - overview of, 229
  - removing items from cache, 237–238
- Caching Architecture Guide for .NET Framework Applications*, 178
- Caching data. *See also* Security Cache
  - clearing cached configuration data, 57
  - for data access logic component, 604–607
  - in Data Mapping Application Block, 590–593
- CachingStoreProvider
  - configuring Security Cache, 468
  - creating custom
    - SecurityCacheProvider, 418–419
  - overview of, 416
- Categories
  - configuring, 387–389
  - default, 389
  - filtering LogEntries based on, 329–331
  - tracing with, 375
- Chaining exceptions, 244, 301
- Change Watcher, 13–14
- Child nodes, dynamically creating, 566–567
- ClearSectionCache method, 49
- ClearSingletonSectionCache method, 57
- CollectAllUserRoles method, 446–448
- Collections, and custom business entities, 157
- CommandBehavior.CloseConnection method, 148–149
- CommandMapping
  - creating new, 612
  - DataSetMappingProvider and, 595
  - DataTableMapping configuration, 608–612
  - insert, update and delete commands, 611–612
  - lazy loading, 609–611
  - runtime configuration, 550–556
- CommandParameterMappings
  - Data Mapping Application Block configuration, 612–617
  - DataSetMappingProvider and, 595
  - runtime configuration, 550–556
- CommandRegistrars, 657
- Commands, configuration design-time, 93–96
- CommandStatementType, runtime, 550–556
- CommandText property,
  - SelectCommandMapping, 601
- CommandTimeout property,
  - SelectCommandMapping, 601
- CompareHash method, 503–504, 523
- Compile type checking, 152–153
- ComPlusInformationProvider, 316
- Configuration Application Block
  - Core of, 651–652
  - defining, 2–3
  - ProviderFactory class of, 540–543
- Configuration Application Block, design-time
  - configuration example, 100–107
  - configuration hierarchies, 93
  - configuration menu items and commands, 93–96
- ConfigurationDesignManager and, 96–100
- IComponent, 86–89
- IContainer, 89–90
- IServiceContainer, 91–93

- IServiceProvider, 91–93
- ISite, 90–91
  - overview of, 84–86
- System.ComponentModel
  - namespace and, 86
  - using Configuration Tool. *See*
    - Enterprise Library
    - Configuration Tool,
    - configuring
- Configuration Application Block,
  - runtime
    - accessing data with
      - ConfigurationContext, 57–59
    - change notification, 55–56
    - clearing cached data, 57
    - configuration, 545
    - configuration sections, 44–46
    - ConfigurationBuilder object, 49
    - ConfigurationContext object, 49–52
    - ConfigurationManager object, 52
    - ConfigurationSettings class, 47–49
    - creating custom distribution
      - strategy, 338–344
    - creating custom transformer, 40–44
    - DatabaseStorageProvider, 17–35
    - decoupling physical storage
      - location, 8–10
    - defining Configuration Application Block, 2–3
    - design of, 3–4
    - Enterprise Library transformers, 36–40
    - IConfigurationProvider/ConfigurationProvider, 12–13
    - IStorageProviderReader/IConfigurationChangeWatcherFactory/Storage Provider, 13–14
    - IStorageProviderWriter, 14–15
    - overview of, 10–12, 52–53
    - reading and writing configuration data, 4–7
    - reading data, 53–54
    - StorageProviders, 15–17
    - transformers, 35–36
    - writing data, 54–55
  - Configuration design-time API, 657
  - Configuration hierarchies, 93
  - Configuration Management
    - Application Block, 2
  - Configuration menu items and
    - commands, 93–96
  - Configuration metadata, 8–10
  - Configuration nodes
    - creating, 557–567
    - from IComponent, 86–89
    - IServiceContainer and
      - IServiceProvider, 93
    - ISite, 90–91
  - Configuration sections
    - configuring for custom
      - Transformers, 78–79
    - configuring for
      - MvpSerializerTransformer, 78–79
    - configuring for
      - XmlSerializerTransformer, 77–78
    - creating, modifying and removing, 63–65
    - encrypting configuration data, 80–84
    - overview of, 44–46
  - ConfigurationBuilder object, 49, 50
  - ConfigurationChangeDatabaseWatcher, 26–30
  - ConfigurationChangedEvent, 55–56
  - ConfigurationChangedEventArgs, 22–26, 27
  - ConfigurationChangeWatchers, 46
  - ConfigurationChangingEventArgs, 22–26
  - ConfigurationContext object, 49–52, 57–59, 653
  - ConfigurationDatabaseChangedEventArgs, 25–26, 27
  - ConfigurationDesignHost
    - creating configuration nodes, 565
    - implementing IContainer, 89–90
    - implementing ISite, 90

## 666 ■ Index

- ConfigurationDesignHost (*continued*)
  - services initialized and added by, 93–94
- ConfigurationDesignManager
  - attribute, 573–574
  - configuration design-time and, 96–100, 104–105, 657
  - implementing, 567–573
- ConfigurationErrorService, 94
- ConfigurationManager
  - clearing cached configuration data, 57
  - detecting configuration changes, 55–56
  - reading configuration data, 52–54
  - saving and retrieving configuration data, 52
  - using ConfigurationContext, 49–50, 57–59
  - writing configuration data, 52, 54–55
- ConfigurationName property, 12
- ConfigurationNodeCommand
  - hierarchy, 96–97
- ConfigurationNodeSite, 90
- ConfigurationProtection, 47
- ConfigurationProvider, 12–13, 537
- ConfigurationSections property, 47–49
- ConfigurationSettings, 47–49
- ConfigurationSource, 653
- ConfigurationView, 12–13
- Connected access, 148–149
- Connection strings, 72, 141–143
- Container property, ISite, 91
- ContextItems, LogEntry, 327–328
- ContextualWrapData, 282–284
- ContextualWrapHandler
  - creating, 284–287
  - creating design-time classes, 287–290
  - InnerException from, 299–300
- Core functionality
  - building application block, 529–530
  - decoupling from, 532–537
  - Enterprise Library configuration, 651–652
- CreateCacheManager method, 185–187
- CreateConfigurationChangeWatcher method, 13–14
- CreateDatabase method, 130
- CreateDefaultDatabase method, 130–131
- CreateDynamicMenuItems method, 560, 567
- CreateDynamicNodes method, 566–567
- CreateHash method
  - Cryptographer class, 493–494
  - IHashProvider interface, 503–504
  - overview of, 522
- CreateParameter method, 133–135
- CreateSite method, 89
- Credentials, caching. *See* Security Cache
- CRUD (Create, Retrieve, Update and Delete) operations
  - data access logic components and, 169–174, 527
  - managing with IMapper class, 587–590
- Cryptographer, 492–494
- Cryptography Application Block, 487–524
  - adding, 509
  - configuring Database
    - AuthenticationProvider, 465
  - creating and comparing hashes, 519–524
  - Cryptographer class, 492–494
  - defining, 488–489
  - encrypting cached data with, 182
  - encrypting/decrypting data, 509–519
  - hash providers, 501–508
  - .NET cryptography and, 489–492
  - overview of, 487–488
  - SymmetricProviderFactory and HashProviderFactory, 495
  - SymmetricProviders, 496–501
- CryptographyUtility, 504–505

- CryptProtectData method, 495–496, 497
  - CryptUnprotectData method, 495–497
  - CurrentSectionName method, 36
  - Custom business entities, 155–158
  - Custom cache storage, 208–219
  - Custom data transfer objects (DTOs), 148
  - Custom exceptions, 245–247
  - Custom StorageProviders. *See* StorageProviders, custom
  - Custom Transformers. *See* Transformers, custom
  - CustomAuthenticationProviderData, 407–412
  - CustomersMapper, 602–604
  - CustomStorageProviderData, 75
- D**
- DALCs. *See* Data access logic components (DALCs)
  - Data Access Application Block
    - creating data cache storage, 206–208
    - Database AuthenticationProvider
      - configuration, 465
    - database command wrappers, 123–124, 133–135
    - database providers, customized, 131–138
    - database providers, design, 113–123
    - DatabaseFactory and
      - DatabaseProviderFactory
        - classes, 128–131
    - defining, 110–111
    - earlier versions of, 111–112
    - Enterprise Library configuration, 657–658
    - parameters and ParameterCache, 124–128
  - Data Access Application Block,
    - developing API, 138–174
    - configuring database provider, 138–146
    - data access logic components (DALCs), 169–174
    - retrieving data, connected access, 148–149
    - retrieving data, custom business entity components, 155–158
    - retrieving data, disconnected access, 149–155
    - retrieving data, overview of, 146–148
    - transaction support, 166–169
    - updating data, 158–166
  - Data access logic components (DALCs)
    - configuring Data Access Application Block, 169–174
    - configuring Data Mapping Application Block, 602–604
    - creating custom ProfileProvider, 453–455
    - designing applications with, 527
  - Data cache storage, 206–208
  - Data integrity, 488
  - Data Mapping Application Block,
    - building
      - adding
        - ConfigurationDesignManager
          - attribute, 573–574
        - Core functionality, 529–530
        - creating configuration nodes, 557–567
        - goals of, 527–528
        - implementing
          - ConfigurationDesignManager, 567–573
        - overview of, 579–580
        - pluggable providers, 530–540
        - runtime configuration, 547–556
        - Unit tests, 574–576
  - Data Mapping Application Block,
    - design, 580–595
    - caching data, 590–593
    - DataBaseWrapper class, 581–587
    - DataMapper class, 587–590
    - DataMappingProviderFactory
      - class, 593–594
    - DataSetMappingProvider, 594–595

## 668 ■ Index

- Data Mapping Application Block,
  - design (*continued*)
  - IDataMappingProvider interface, 593–594
  - transaction support, 585–587
  - two-sweeps update procedure, 584–585
- Data Mapping Application Block,
  - developing with, 595–620
  - adding Application Block, 598–599
  - caching data, 604–607
  - CommandMappings, 608–612
  - CommandParameterMappings, 612–617
  - creating data access logic
    - component, 602–604
  - creating DataSets and stored
    - procedures, 597–598
  - DataSetMapping, 599–600
  - DataTableMappings, 607–608
  - overview of, 595–597
  - SelectCommandMapping, 600–602
  - transactions, 617–619
- Data privacy, 487–488
- Data Protection API. *See* DPAPI (Data Protection API)
- Data Protection Configuration dialog, 510–511
- Data transfer, 18–22
- DataBackingStore
  - cache storage class, 201–203
  - configuring, 224–225
  - permitting encrypted cache data, 219
- Database class database provider, 111–112, 114–117
- Database command wrappers, 123–124, 133–135
- Database providers
  - configuring, 138–146
  - customizing, 131–138
  - Database class, 114–117
  - DB2Database class, 122–123
  - design, 113–115
  - online resources for, 112
  - OracleDatabase class, 120–122
  - SqlDatabase class, 117–120
- Database, setting up, 18
- Database types, 139–141
- DatabaseDependencyExpiration
  - enabling caching, 606
  - HasExpired method of, 196–198
  - setting for CacheItem, 198
- DatabaseDistributionStrategy, 339–340
- DatabaseDistributionStrategyData, 338
- DatabaseDistributorSettings, 342–343
- DatabaseFactory, 112–113, 131, 146–155
- DatabaseInstance
  - DataBackingStore configuration, 224–225
  - database provider configuration, 142–146
  - DataSetMapping configuration, 600
  - DataSetMappingProvider
    - configuration, 595
  - defined, 131
  - design-time example configuration, 101
  - runtime configuration, 550–556
  - Security Database Administration
    - Console and, 459–460
- DatabaseProfileProvider, 480–484
- DatabaseProviderFactory, 128–131
- DatabaseRolesProvider, 477–478
- DatabaseSink, 348–349, 384–385
- DatabaseStorageProvider, 17–35
  - creating ConfigurationChange-DatabaseWatcher, 26–30
  - creating DatabaseStorageProvider, 30–33
  - creating data transfer object, 18–22
  - creating specific classes, 22–26
  - design-time example configuration, 106
  - overview of, 17–18
  - post-creation (changing metaconfiguration), 33–35
  - setting up database, 18

- storage locations, 8–9
  - using ConfigurationContext, 58–59
- DatabaseStorageProviderData, 18–22, 102
- DataBaseStorageProviderNode, 100–106
- DatabaseWrapper
  - building application block, 532
  - building Data Access Application block, 529–530
  - DataMappingFactory and, 593
  - decoupling from Core
    - functionality, 535–537
  - overview of, 581–584
  - transaction support, 585–587
  - two-sweeps update procedure, 584–585
  - unit tests, 576
- DataCacheSettings, 529–530
- dataConfiguration.config, 139
- DataMapper, 529–530, 587–590
- DataMappingCollection, 535–537, 593–594
- DataMappingConfigurationDesign-Manager, 567–573
- DataMappingFactory, 593–594
- DataMappingProvider
  - building application block, 532
  - overview of, 593–594
  - runtime configuration, 547–556
- DataMappingProviderData property, 561–564
- DataMappingSettings, 548–549
- DataMappingsProviderFactory, 541–543, 593–594
- DataReaders, 148–149
- DataSetMapping
  - Data Mapping Application Block
    - configuration, 599–600
  - DataSetMappingProvider and, 595
  - decoupling from Core
    - functionality, 535–537
  - enabling caching, 604–605
  - SelectCommandMapping
    - configuration, 601–602
- DataSetMappingDataCollection, 595, 596
- DataSetMappingProvider
  - adding menu items for, 559
  - configuration data for, 538–540
  - configuring DataSetMapping, 600
  - Data Mapping Application Block
    - and, 594–595
  - DataMappingProviderData
    - property, 561–564
  - decoupling from Core
    - functionality, 535–537
  - OnSited method of, 565–566
  - runtime configuration, 549
- DataSetRefreshAction, 607
- DataSets
  - caching data and, 590–591
  - creating, 597–598
  - data access logic component
    - returning, 171–174
  - disconnected access with, 148–152
  - typed, 152–155
  - updating data with UpdateDataSet, 159–166
- DataSetType, 602
- DataTableMapping
  - configuring CommandMappings, 608–612
  - creating, 607–608
  - DataSetMappingProvider and, 595
  - runtime configuration, 550–556
- DB2Database database provider, 122–123
- DbAuthenticationProvider
  - configuring authentication, 464–467
  - overview of, 405–406
  - Security Database Administration
    - Console and, 459–460
- DbCommandWrapper
  - creating, 133–135
  - handling parameters and
    - ParameterCache, 124–128
  - methods and properties of, 126
  - overview of, 123–124
- DbProfileProvider, 451, 452

## 670 ■ Index

- DbRolesProvider, 445
  - DebugInformationProvider, 316–318
  - DebugSink, 351–356, 362
  - DebugSinkData method, 355–356
  - Declarative security, 438–442
  - Decoupling physical storage location, 8–10
  - Decrypt method, 495
  - Decrypting data, 509–510
  - DecryptSymmetric method, 493–494, 519
  - DefaultCacheManager property, 222–223
  - DefaultSelectCommand, SelectCommandMapping, 601
  - Delete command, DataTableMapping, 611–612
  - Demand method, 439–441
  - Deployment issues, 152–153, 157
  - Deserialize method, 36, 38
  - Design goals, building application block, 526–528
  - Design-time configuration classes. *See also* Configuration Application Block, design-time; Enterprise Library Configuration Tool, configuring
    - building application block, 556–574
    - custom AuthenticationProvider, 411–412
    - custom BackingStore, 214–219
    - custom distribution strategy, 340
    - custom DistributorService, 343
    - custom Sinks, 355–356
    - exception handlers, 287–290
  - Destinations
    - components of, 334–335, 344
    - configuring, 387–389
    - filtering LogEntries based on, 329–331
    - log distribution and. *See* Log distribution
  - Dictionary attacks, 492
  - DictionaryToken, 361–366
  - Digital certificates, 488
  - Disconnected access
    - configuring DataSets, 149–152
    - configuring typed DataSets, 152–155
    - with custom business entities, 155–158
  - Discretionary access control lists (DACLS), 420
  - Disk-resident cache, 204–208
  - Distributor, 334, 377–381
  - DistributorService
    - customizing, 341–344
    - MsmqListener and, 336
    - MsmqLogDistributionStrategy and, 335–336
  - DPAPI (Data Protection API)
    - DpapiSymmetricCryptographyProvider, 495, 497
    - enabling, 80–84
    - overview of, 495–497
    - protecting plain text files, 47
    - SymmetricAlgorithmProvider, 497–498
    - SymmetricCryptographer, 497–498
  - DpapiSymmetricCryptographyProvider
    - configuring, 510–511
    - defined, 495
    - overview of, 497
  - DTOs (data transfer objects), custom, 148
  - Dynamic parameter discovery, 124–125, 165–166
- E**
- EHAB. *See* Exception Handling Application Block (EHAB)
  - EIF (Enterprise Instrumentation Framework), 309
  - ELConfigurationContextInformationProvider, 320–324
  - EMAB (Exception Management Application Block), 243–244, 246–247
  - EmailSink, 349, 384–386

- Encapsulation, 156, 534–535
- Encryption. *See also* Cryptography
  - Application Block
    - cached data, 219–220, 226–228
  - Caching Application Block and, 182
  - configuration data, 79–84
  - Encrypt method, 495
  - EncryptSymmetric method, 493–494, 516–518
- Enterprise Instrumentation
  - Framework (EIF), 309
- Enterprise Library Configuration (SolutionConfigurationNode), 88
- Enterprise Library Configuration Tool, configuring, 62–84
  - AppConfigFileStorageProvider, 68–70
  - Caching Application Block, 219
  - configuration sections, 63–65
  - Cryptography Application Block, 509
  - custom StorageProviders, 73–76
  - custom Transformers, 78–79
  - defining, 3
  - design-time, 3
  - encrypting configuration data, 79–84
  - MvpSerializerTransformer, 78
  - overview of, 62–63
  - RegistryStorageProvider, 70–72
  - runtime, 544–545
  - SqlStorageProvider, 71–73
  - XmlFileStorageProvider, 66–68
  - XmlSerializerTransformer, 77–78
- Enterprise Library, .NET framework 2.0, 651–659
  - Configuration design-time API, 657
  - Core, 651–652
  - Data Access Application Block, 657–658
  - instrumentation, 654–656
  - logging, 658–659
  - ObjectBuilder subsystem, 656–657
  - security, 659
  - StorageProviders. *See* StorageProviders
  - System.Configuration and
    - configuration runtime, 652–654
  - Transformers. *See* Transformers
- Enterprise Library providers. *See* Providers
- Entitlement, Authorization Manager, 432–433
- Entropy, 496–497
- EventLogSink, 349, 385
- Events, ConfigurationManager, 53
- Exception handlers. *See also* ExceptionFormatter
  - chaining exceptions, 301
  - creating design-time classes, 287–290
  - logging exceptions, 295–297
  - LoggingExceptionHandler, 260–262
  - overview of, 255–256
  - ReplaceHandler, 259–260
  - replacing exceptions, 295
  - setting ExceptionFormatter, 297
  - WrapHandler, 256–259
  - wrapping exceptions, 294
- Exception handlers, custom
  - advanced exception handling, 300–301
  - creating ContextualWrapData, 282–284
  - creating ContextualWrapHandler, 284–287
  - creating WrapperException, 280–282
  - overview, 275–280
  - simple exception handling, 297–300
- Exception Handling Application Block (EHAB), 241–305
  - custom exceptions and, 245–247
  - defining, 242–243
  - exception hierarchies and, 245–247
- Exception Management
  - Application Block and, 243–244
  - overview of, 241–242

## 672 ■ Index

- Exception Handling Application
  - Block (EHAB), design, 247–290
  - creating design-time classes, 287–290
  - customizing, creating
    - ContextualWrapData, 282–284
  - customizing, creating
    - ContextualWrapHandler, 284–287
  - customizing, creating
    - WrapperException, 280–282
  - customizing, overview, 275–280
  - exception handlers overview, 255–256
- ExceptionFormatters, custom, 269–275
- ExceptionFormatters, overview of, 262–265
- ExceptionFormatters,
  - TextExceptionFormatter, 265–267
  - XmlExceptionFormatter, 267–269
- ExceptionPolicy class, 248–254
- LoggingExceptionHandler, 260–262
- overview of, 247–248
- ReplaceHandler, 259–260
- WrapHandler, 256–259
- Exception Handling Application
  - Block (EHAB), developing with, 290–305
  - advanced customized exception handling, 300–301
  - chaining exceptions, 301
  - configuring overview, 291–294
  - logging an exception, 295–297
  - overview of, 290–291
  - replacing an exception, 295
  - setting ExceptionFormatter, 297
  - simple customized exception handling, 297–300
  - using API, 301–304
  - wrapping an exception, 294
- Exception handling chain, 258
- Exception hierarchies, 245–247
- Exception Management Application
  - Block (EMAB), 243–244, 246–247
- Exception Management in .NET Architecture Guide*, 255
- Exception policy, 291–294
- Exception type
  - adding to exception policy, 292–294
  - simple customized handlers, 298–299
  - wrapping exceptions, 294
- ExceptionFormatter
  - custom, 269–275
  - defining, 247
  - overview, 262–265
  - setting, 297
  - TextExceptionFormatter, 265–267
  - XmlExceptionFormatter, 267–269
- ExceptionHandler, 248
- ExceptionNotHandledEvent, 254
- ExceptionPolicy, 248–254
- ExceptionPolicyEntry, 252
- ExceptionPolicy.HandleException
  - method, 278
- Exceptions, and Caching Application
  - Block, 189–190
- ExceptionUtility, WrapHandler, 258–259
- ExecuteDataSet method
  - generic DataSets, 151–152
  - LoadDataSet method vs., 154
  - OracleDatabase database provider, 120–122
- ExecuteNonQuery method, 146–147, 158–159
- ExecuteReader method, 120–122
- ExecuteScalar method, 146–147
- ExecuteXmlReader method, 118–120, 135–137
- Expiration policies, 191–198
  - adding items to cache, 230
  - adding items to cache with multiple, 230–231
- Caching Application Block, 182

- Caching Application Block API, 230–231
    - customizing, 196–198
    - notification-based, 195–196
    - overview of, 191–192
    - time-based, 192–195
  - ExpirationPollFrequencyInSeconds
    - setting, 191, 221–222
  - ExpirationPollTimer, 185, 191
  - ExpireIdentity method, 470–471
  - ExpirePrincipal method, 472
  - ExpireProfile method, 474
  - Explicit flushing, of cache, 238–239
  - Explicit parameter handling, 128
  - ExtendedFormatTime expiration
    - policies, 195
  - ExtendedProperties dictionary, 314–315, 390
  - Extensibility
    - custom business entities and, 157
    - decoupling from Core functionality and, 534
    - generic DataSets and, 150
    - Typed DataSets and, 152–153
  - Extensions collection, 225, 227
  - ExtensionsConfigurationDesignManager, 105–106
  - ExtraInformationProvider, 315–324
    - ComPlusInformationProvider, 316
    - custom, 320–324
    - custom TokenFunction and Formatter for, 363–366
    - DebugInformationProvider, 316–317
    - IExtraInformationProvider, 315–316
    - ManagedSecurityContextInformationProvider, 318–319
    - overview of, 315
    - UnmanagedSecurityContextInformationProvider, 319
- F**
- Factories
    - building application block with, 540–543
    - defined, 185
  - File Key Algorithm StorageProvider, 82–83
  - FileDependency expiration, 195–196
  - FileKey Algorithm StorageProvider, 513–514
  - Filters, log. *See* LogFilter
  - FlatFileSink, 349, 383, 385–386
  - Flush method, 239
  - FlushContextItems method, 327–328
  - Flushing cache, 238–240
  - Format method
    - custom ExceptionFormatter, 272–273
    - DictionaryToken class, 363–366
    - ExceptionFormatter class, 262–263
  - Formatters. *See also* ExceptionFormatter
    - configuring, 381–383
    - DictionaryToken class, 361–362
    - KeyValueToken class, 361
    - overview of, 356–357
    - role of, 334–335, 344
    - setting default, 389
    - TimeStampToken class, 360
    - TokenFunction class, 359–360
  - FormatterTypeName property, 297
- G**
- Get connection method
    - Database class database provider, 116
    - OracleDatabase database provider, 121
    - SqlDatabase database provider, 117–120
  - GetAuthenticationProvider method, 403, 465
  - GetAuthorizationProvider method, 423
  - GetBoolean method, 122
  - GetByte method, 122
  - GetConfigStoredProcedure property, 72, 101
  - GetConfiguration method, 52–54

## 674 ■ Index

- GetCurrentLastWriteTime method, 27–30
  - GetDataAdapter method
    - Database class database provider, 116
    - OracleDatabase database provider, 121
    - SqlDatabase database provider, 117–120
    - UpdateDataSet, 160
  - GetDataMappingProvider, 593–594
  - GetGuid method, 122
  - GetIdentity method, 469–470
  - GetPrincipal method, 471
  - GetProfile method
    - configuring Security Cache, 473–474
    - DbProfileProvider and, 451
    - IProfileProvider interface and, 451
    - PortalProfileProvider and, 455–457
    - ProfileProvider and, 483–485
  - GetProfileProvider method, 451
  - GetRoles method, 479
  - GetRolesProvider method, 443
  - GetSecurityCacheProvider method, 415
  - GetSecurityInfo method, 319
  - GetService method, 89
  - GetSqlCommandWrapper method
    - creating database provider, 135–138
    - Database class database provider, 116
    - DbCommandWrapper class, 121
    - OracleDatabase database provider, 121
    - SqlDatabase database provider, 117–120
  - GetStoredProcCommandWrapper method
    - creating database provider, 135–138
    - Database class database provider, 116
    - DbCommandWrapper class, 121, 126
    - OracleDatabase database provider, 121
    - SqlDatabase database provider, 117–120
  - GetStoredProcCommandWrapper-WithSourceColumns method, 165–166
  - GetUserNameEx method, 319
  - GuidToken, 417
- ## H
- HandleException method
    - ContextualWrapHandler class, 285–287
    - Exception Handling Application Block API using, 301–304
    - overview of, 250–251
    - WrapHandler class, 256–258
  - HandlingInstanceID, 258–259
  - HasExpired method, 196–198
  - Hash providers, 501–508
    - configuring, 520–521
    - custom hash algorithms, 505–508
    - DbAuthenticationProvider class, 406
    - HashAlgorithmProvider and HashCryptographer classes, 503–505
    - KeyedHashAlgorithmProvider class, 505–506
    - overview of, 501–503
  - Hash value, defined, 491
  - HashAlgorithmProvider, 503–505
  - HashCryptographer, 503–505
  - Hashes
    - algorithms, 491–492
    - creating and comparing, 519–524
  - HashProvider, 460
  - HashProviderFactory, 494–495
  - Hierarchies
    - configuration, 93, 96–97
    - exception, 245–247
    - Exception Handling Application Block and, 291
    - iUIHierarchy interface, 93–94
  - HttpCachingStoreProvider, 417–420

- I**
- IAuthenticationProvider interface, 403–405, 407–412
- IAuthorizationProvider interface, 423–425
- IBackingStore interface, 182, 201–203
- IBM DB2Database, 122–123
- ICacheItemExpiration interface, 196–198, 591
- ICacheItemExpirationPolicies interface, 591
- ICacheItemRefreshAction interface, 198–200, 231
- ICacheStorage interface, 182
- ICloneable interface, 637–642
- IComponent interface, 86–89
- IConfigurationChangeWatcherFactory interface, 13–14
- IConfigurationDesignManager interface, 96–100, 567
- IConfigurationProvider interface, 12–13, 36–40
- IContainer interface, 89–90
- IDataMappingProvider interface, 532, 535–537, 593–594
- IDataParameter interface, 133–135, 622–625
- IDataParameterCollection interface, 625–629
- IDataReader interface, 122, 629–633
- IDbCommand interface, 633–637
- IDbConnection interface, 637–642
- IDbDataAdapter interface, 642–647
- IDbTransaction interface, 166–169, 586–587, 647–650
- Identity objects, in authentication, 405
- IDictionary interface. *See* ExtraInformationProvider
- IExtraInformationProvider interface, 315–316
- IFormatter interface, 356–357
- IHashProvider interface, 503–504
- ILogDistributionStrategy interface, 333, 338–339
- ILogFormatter interface, 357
- ILogSink interface, 345
- Imperative security checks, 438–442
- IncludeType element, custom StorageProvider, 34
- Inheritance, and Typed DataSets, 152–153
- Initialize method
  - custom BackingStores, 209–214
  - custom exception handlers, 277
  - custom Sinks, 351, 353–356
  - custom StorageProviders, 75–76
- IConfigurationProvider interface and, 12
- provider configuration and, 537
- WrapHandler, 256–258
- InnerException
  - ContextualWrapHandler, 290, 299–300
  - WrapHandler, 256–258, 259
- InProcLogDistributionStrategy interface
  - configuring, 378
  - distributing LogEntries to Destinations, 333–334
  - LogDistributor function, 334–335
- Insert command, DataTableMapping, 611–612
- InsertUpdate, runtime, 550–556
- Instrumentation, Enterprise Library, 654–656
- IntelliSense, and Typed DataSets, 152–153
- Interoperability
  - custom business entities, 156
  - DataSets, 151
- IProfileProvider interface, 449–451
- IRolesProvider interface, 443–445
- ISecurityCacheProvider interface, 413–416, 418, 468
- IServiceContainer interface, 91–93
- IServiceProvider interface, 91–93
- ISite interface, 90–91
- Isolated storage, 205–206

## 676 ■ Index

- IsolatedStorageBackingStore
  - cache storage and, 201–203
  - Caching Application Block and, 182
  - configuring, 224
  - permitting encrypted cache data, 219
- IsolationLevel, 618–620
- IStorageProvider interface, 74
- IStorageProviderReader interface, 9–11, 13–14
- IStorageProviderWriter interface, 9–11, 14–15
- ISymmetricProvider interface, 495, 503
- IToken interface, 416–417
- ITransformer interface, 9
- ITransformer interface, 35–36
- IUIHierarchy interface, 93, 94
  
- K**
- Key management, encryption, 512–515
- KeyAlgorithmPair, 48, 80–84
- Keyed hashes, 491
- KeyedHashAlgorithmProvider, 505–506, 520–521
- KeyValueToken, 361
  
- L**
- LinkNodeService, 94
- ListDictionary, 616
- LoadData method, 30–33
- LoadDataSet method, 120–122, 153–155
- Loading items in cache, 231–236
  - asynchronous pull loading, 235
  - proactive loading method, 232–233
  - reactive loading method, 235–236
  - synchronous pull loading, 236
- Log Alert, configuring, 386
- Log creation, 311–328
  - basic logging, 311–315
  - custom ExtraInformationProvider, 320–324
  - custom LogEntry, 325–328
  - extra information, 315–319
  - LogWriter and LogEntry, 324–325
  - primary classes for, 311–312
- Log distribution, 332–344
  - custom strategy for, 337–344
  - DistributorService, 335–336
  - ILogDistributionStrategy interface, 333
  - InProcLogDistributionStrategy interface, 333
  - Log Distributor, 334–335
  - MsmqListener, 336
  - MsmqLogDistributionStrategy interface, 333–334, 336
  - MsmqLogDistributor, 336–337
- LogDistributor, 334–335, 345
- LogEntries
  - ContextItems and, 327–328
  - custom, 325–327
  - Enterprise Library logging configuration, 659
  - ExtendedProperties and, 315
  - filtering, 328–332
  - intrinsic properties of, 324–325
  - log creation and, 311–314
  - log distribution and. *See* Log distribution
- LogFilter
  - Categories, 329–331
  - configuring Logging and Instrumentation Application Block for, 377
  - Enterprise Library logging configuration, 659
  - generic DataSets, 150
  - LogEntry, 328–329
  - Priorities, 332
- Logger, 311–315, 390–392
- Logging and Instrumentation Application Block, 307–395
  - defining, 308–309
  - logging exceptions, 295–297
  - LoggingExceptionHandler using, 261–262

- previous version of Logging
    - Application Block and EIF, 309–310
  - using API, 389–394
- Logging and Instrumentation
  - Application Block, design filtering, 328–332
  - log creation. *See* Log creation
  - log distribution. *See* Log distribution
  - logging while tracing, 375–376
  - overview of, 310–311
  - routing. *See* Routing
  - tracing, 366–376
- Logging and Instrumentation
  - Application Block, developing with
    - Categories and Destinations, 387–388
    - client settings, 377–379
    - defaults, 388–389
    - Distributor settings, 379–381
    - Formatters, 381–383
    - overview of, 376–377
    - Sinks, 383–387
  - Logging Application Block, 309, 652
  - Logging, .NET framework 2.0, 658–659
  - LoggingExceptionHandler
    - customizing ExceptionFormatter, 274–275
    - defined, 244
    - logging exceptions, 295–297
    - overview of, 260–262
  - LogSink. *See* Sinks
  - LogWriter
    - log creation and, 311
    - log distribution and, 332
    - using ContextItems, 328
- M**
- ManagedSecurityContextInformation-Provider, 318–319
- Marking, in expiration process, 191–192
- Master key, saving, 515
- MaximumElementsinCacheBefore-Scavenging, Cache Manager, 221–222, 239–240
- Memory-resident cache, 204
- Menu items
  - adding in configuration design-time, 93–96, 559
  - dynamically adding, 560–561
- MenuContainerService, 94
- Metaconfiguration
  - configuration sections as part of, 44–46
  - custom StorageProvider, 33–35
  - custom Transformer, 43–44
  - modifying data for an application, 139
  - MvpSerializerTransformer, 39–40
  - storing application configuration data with, 8–10
- Methods
  - BaseBackingStore, 202
  - CacheManager, 188
  - ConfigurationContext, 51
  - ConfigurationDesignHost, 89–90
  - ConfigurationManager, 53
  - ConfigurationNodeSite, 91
  - Cryptographer class, 493
  - Database class database provider, 116
  - DatabaseProviderFactory class, 130
  - DataBaseWrapper, 582
  - DataMapper, 589–590
  - DB2Database class, 123
  - DbCommandWrapper class, 125
  - ExceptionFormatter, 264
  - ExecuteXmlReader method, 118–120
  - IConfigurationDesignManager interface, 98
  - IContainer interface, 89
  - ISecurityCacheProvider, 415
  - ITransformer interface, 36
  - LogSink, 348

## 678 ■ Index

- Methods (*continued*)
    - OracleDatabase database provider, 121
    - ServiceContainer, 92
    - TextExceptionFormatter, 268
    - UserRoleManager, 462
  - MSMQ (Microsoft Message Queueing), 333
  - MsmqListener, 336–337
  - MsmqLogDistributionStrategy
    - configuring, 378–381
    - DistributorService and, 335–336
    - sending messages asynchronously, 333–334
  - MsmqLogDistributor, 336–337
  - MsmqSink, 349, 386
  - MvpSerializerTransformer
    - configuring, 78
    - defined, 36
    - overview of, 38–40
- N**
- NamePasswordCredentials
    - configuring authentication, 466–467
    - creating custom
      - AuthenticationProvider, 407–408
    - DbAuthenticationProvider and, 405
    - defined, 399
    - overview of, 406
    - supporting
      - IAuthenticationProvider with, 405
  - .NET cryptography
    - hash algorithms, 491–492
    - overview of, 489
    - symmetric and asymmetric algorithms, 490–491
  - .NET managed data provider, creating, 621–650
    - ICloneable interface, 637–642
    - IDataParameter interface, 622–625
    - IDataParameterCollection interface, 625–629
    - IDataReader interface, 629–633
    - IDbCommand interface, 633–637
    - IDbConnection interface, 637–642
    - IDbDataAdapter interface, 642–647
    - IDbTransaction interface, 647–650
    - overview of, 621–622
    - NeverExpire setting, caches, 230
    - NodeCreationService, 94, 104–105
    - NodeMapRegistrars, 657
    - NodeNameCreationService, 94
    - Nodes. *See* Configuration nodes
    - Nonrepudiation, 488
    - Notification-based expirations, 191, 195–196
    - Notification, of exceptions, 275–276
    - NullBackingStore
      - cache storage and, 201–203
      - configuring, 223
      - encrypted cache data not allowed for, 219
      - as memory-resident cache, 204
    - NumberToRemovedWhenScavenging, CacheManager, 221–222, 239–240

**O**

    - ObjectBuilder, 656–657
    - OLE DB, 111
    - OleDbHelper, 111
    - OnConfigurationChanged event
      - notification, 55–56
    - Online resources
      - ChangeWatcher, 14
      - customized Sinks, 351
      - Data Mapping Application Block design, 528
      - database providers, 112, 114
      - database providers for MySQL and OleDb, 132
      - DPAPI (Data Protection API), 47
      - SqlDependency Expiration, 196
    - OnSited method, 565–566
    - Open method, 98–99, 570–571
    - Optimistic concurrency, 150, 159–166
    - OracleDatabase database provider, 120–122

- OracleDataReaderWrapper, 122
- OracleHelper, 111
- P**
- ParameterCache, 124–128
- Parameters, 124–128, 133–135
- ParameterToken method
  - Database class database provider, 116
  - OracleDatabase database provider, 121
  - SqlDatabase database provider, 117–120
- Parent-child relationships, configuration nodes, 89
- Partitioned caches, 206–207
- Passwords, 519–524
- Performance, and caches, 179
- Plain text files, protecting, 47
- Pluggable providers, 530–540
  - configuration, 537–540
  - decoupling from Core functionality, 532–537
  - overview of, 530–532
  - provider type implementation, 532
- PopulateDictionary method. *See* ExtraInformationProvider
- Portability, 112
- PortalProfile, 481–483
- PortalProfileMapper, 455–457
- PortalProfileProvider, 455–457
- PostHandlingAction, 253, 293
- PrepareMessage method, 347–348
- Priorities, filtering LogEntries, 332
- Private fields
  - custom business entities and, 156
  - generic DataSet issues, 151
- Proactive loading method, 232–233
- Process Log method, 337
- Profile object, creating, 480–483
- ProfileFactory, 480
- ProfileProviderFactory, 449–451, 480
- ProfileProviders, 448–457
  - configuring, 480–484
  - custom, 452–453
  - data access logic component for, 453–455
- DbProfileProvider, 451
- default settings for, 484
- overview of, 448–449
- PortalProfileProvider, 455–457
- ProfileProviderFactory and IProfileProvider, 449–451
- Properties
  - BaseBackingStore, 202
  - CacheManager, 188
  - configuration node, 89
  - ConfigurationContext, 51
  - ConfigurationDesignHost, 89–90
  - ConfigurationNodeSite, 90–91
  - ConfigurationSettings, 47
  - configuring SqlStorageProvider, 72
  - DataBaseWrapper, 582
  - DataMapper, 589–590
  - DB2Database class, 123
  - DbCommandWrapper class, 125
  - design-time example configuration, 101–102
  - ExceptionFormatter, 264
  - ExecuteXmlReader method, 118–120
  - IContainer interface, 89
  - LogEntry, 324–325
  - OracleDatabase database provider, 121
  - ServiceContainer, 92
  - TextExceptionFormatter, 268
- Provider Model Design Pattern, 10–11
- Provider type, implementing, 532
- ProviderData, 537–538
- ProviderFactory, 540–543
- Providers, 10–36. *See also* StorageProviders
  - DatabaseStorageProvider, 17–35
  - IConfigurationProvider/ConfigurationProvider, 12–13
  - IStorageProviderReader/IConfigurationChangeWatcherFactory/Storage Provider, 13–14
  - IStorageProviderWriter, 14–15
  - overview of, 10–12

## 680 ■ Index

Publish method, 248  
PutDataSet method, 583–584

**Q**

Quick Start  
  customizing exception handlers, 276  
  customizing Sinks, 351–356  
  overview of, 576

**R**

Reactive loading method, 235–236  
Reading configuration data, 4–7, 53–54  
Ref cursors, 121  
RefreshAction, 591–592, 607  
Register method  
  ConfigurationDesignManager, 567–570  
  configuring new BackingStore, 217–218  
  design-time example configuration, 105  
  IConfigurationDesignManager interface, 98–99  
RegistryRoot property, RegistryStorageProvider, 70–71  
RegistryStorageProvider  
  configuring, 70–72  
  defining, 9  
  overview of, 15–16  
RegistrySubkey property, RegistryStorageProvider, 70–71  
Relative paths, and XmlFileStorageProvider, 66  
Remove method  
  IContainer interface, 89  
  removing items from cache, 237–238  
ReplaceHandler  
  chaining notification handler with, 276  
  overview of, 259–260  
  replacing exceptions, 295  
Retrieving items from cache, 237  
RijndaelManaged algorithm, 228, 511  
Role-based authorization, 420

Role management, 442–448  
  Authorization Manager and, 433  
  creating custom RolesProvider, 446–448  
  RolesProviderFactory, IRolesProvider and, 443  
  Security Database Administration Tool and, 460–464  
  SecurityRolesProvider and, 443–445  
RolesFactory, 477–480  
RolesProviderFactory, 443, 477–480  
RolesProviders  
  configuring, 477–480  
  custom, 446–448  
  RolesProviderFactory, IRolesProvider and, 443  
  SecurityRolesProvider and, 443–445  
Routing, 344–366  
  creating custom event Sink and, 351–356  
  Formatters and. *See* Formatters  
  Sinks and, 345–350  
RowUpdating event handler, 164  
Rule expressions, 426–429, 475–476  
Runtime configuration, 543–556. *See also* Configuration Application Block, runtime

**S**

Salted hashes, 491–492, 521  
SaveIdentity method, 469  
Save method, 98–99, 570–573  
SavePrincipal method, 471  
SaveProfile method, 472–473  
Scalability, and caches, 179–180  
Scavenging  
  Caching Application Block and, 182  
  flushing cache with, 238–240  
Scope, Authorization Manager, 433–434  
Searches, and custom business entities, 157  
Security Application Block, 397–400  
Security Application Block, design, 399–457

- authentication, 402–412
- authorization. *See* Authorization
- caching security credentials, 412–420
  - profile management, 448–457
  - role management, 442–448
- Security Database, 401–402
- Security Application Block,
  - developing with, 457–484
  - authentication, 464–467
  - authorization, 474–477
  - overview of, 659
  - ProfileProviders, 480–484
  - RolesProviders, 477–480
  - security cache, 467–474
  - Security Database Administration Console, 458–464
- Security Cache, 412–420
  - CachingStoreProvider, 416
  - creating custom SecurityCacheProvider, 417–420
  - IToken, 416–417
  - overview of, 412–413
  - SecurityCacheFactory and ISecurityCacheProvider, 413–416
- Security Cache, configuring, 467–474
  - CachingStoreProvider, 468
  - ExpireIdentity method, 470–471
  - ExpirePrincipal method, 472
  - ExpireProfile method, 474
  - GetIdentity method, 469–470
  - GetPrincipal method, 471
  - GetProfile method, 473–474
  - ISecurityCacheProvider, 468
  - SaveIdentity method, 469
  - SavePrincipal method, 471
  - SaveProfile method, 472–473
- Security Database
  - Administration Console, 402
  - Administration Console configuration, 458–464
  - overview of, 401–402
- SecurityCacheFactory, 413–416
- SecurityCacheItem, 416
- SecurityCacheProvider, custom, 417–420
- SecurityRolesProvider, 443–445
- SelectCommandMapping
  - configuring Data Mapping Application Block, 600–602
  - creating data access logic component, 602–604
  - DataSetMappingProvider and, 595
  - enabling caching, 606
  - lazy loading with, 609–611
  - runtime configuration, 550–556
- SelectCommands, 550–556
- Semi-static data, 592–593
- SendLog method, 339–340
- SendMessage method, 345, 347–348
- SendMessageCore method
  - custom Sinks, 351–352, 355–356, 362
  - EventLogSink, 349
  - FlatFileSink, 349
  - LogSink, 347–348
  - WMILogSink, 350
- Serializable attribute, custom Transformers, 40
- Serialization
  - custom business entity issues, 157
  - generic DataSets supporting, 150
  - reading and writing configuration data, 4–7
- SerializationInfo object, 246
- Serialize method, 36–38
- Service object, retrieving, 91–92
- ServiceContainer, 91–92
- ServiceHelper utility, 104
- SetConfigStoredProcedure property, 72, 101–102
- SetContextItem method, 327–328, 390–392
- SetProfile method
  - DbProfileProvider, 451
  - IProfileProvider interface, 451
  - PortalProfileProvider, 455–457
  - ProfileProvider, 483
- Shared partitions, 207

## 682 ■ Index

- Sinks
  - configuring, 383–386
  - custom, 351–356, 362
  - DatabaseSink, 348–349
  - Destinations and, 334–335
  - EmailSink, 349
  - FlatFileSink, 349
  - MsmqSink, 349
  - overview of, 345–348
  - routing LogEntries to, 344
  - WMILogSink, 350
- SlidingTime expiration policies, 194–195
- Smart client applications, 181
- SolutionConfiguration node, 88
- Sorting
  - custom business entity issues, 157
  - generic DataSets, 150
- SourceColumn property, 612–613
- SqlConfigurationSource, 653
- SqlDatabase database provider, 117–120, 127–128
- SqlDependency Expiration, 196
- SqlHelper, 111, 114
- SQL Server, data access helper, 111
- SqlStorageProvider
  - configuring, 71–73
  - defining, 9
  - overview of, 15–16
- Storage, decoupling location of
  - physical, 8–10
- StorageProviders, 18–22
  - AppConfigFileStorageProvider, 15–16
  - configuring, 66–76
  - defined, 2
  - Enterprise Library, 10–12, 15–17
  - FileKey Algorithm
    - StorageProvider, 82–83, 513–514
  - IStorageProviderReader, 9, 11, 13–14
  - IStorageProviderWriter, 9, 11, 14–15
  - managing with
    - ConfigurationBuilder, 49
    - RegistryStorageProvider, 9, 15–16
    - responsibilities of, 9
    - SqlStorageProvider, 9, 15–16
    - XmlFileStorageProvider, 9, 15–16
- StorageProviders, custom, 17–35
  - configuring with Configuration
    - Tool, 73–76
  - creating
    - ConfigurationChangeDatabase
      - Watcher, 26–30
    - creating data transfer object, 18–22
    - creating DatabaseStorageProvider, 30–33
    - creating specific classes, 22–26
    - overview of, 17–18
    - post-creation (changing
      - metaconfiguration), 33–35
    - setting up database, 18
    - storage locations, 8–9
- Stored procedures
  - creating, 617–619
  - runtime configuration, 550
- Stores, defined, 205
- Strategies, ObjectBuilder, 656–657
- Stream ciphers, 490
- Strings, and hashes, 522, 523
- Strings, connection, 72, 141–143
- SubjectLineEnder, Sinks, 386
- Subkey paths,
  - RegistryStorageProvider, 70–71
- Sweeping, expiration process, 191–192
- Symmetric algorithms
  - encrypting cached data, 226–228
  - encrypting configuration data, 81
  - overview of, 490–491
- SymmetricAlgorithmProvider
  - configuring, 510–516
  - defined, 495
  - overview of, 497–498
- SymmetricCryptographer, 497–498
- SymmetricCryptoProviderFactory, 494
- SymmetricProviderFactory, 495
- SymmetricProviders, 496–501

- adding for encrypting/decrypting data, 509–510
  - custom symmetric algorithms, 498–501
  - defining DPAPI, 495–496
  - DpapiSymmetricCryptographyProvider, 497
  - overview of, 495
  - SymmetricAlgorithmProvider, 497–498
  - SymmetricCryptographer, 497–498
  - Synchronous pull loading, cache items, 236
  - System.ComponentModel namespace, 86–93
    - configuration design-time and, 84–85
    - deriving interfaces from, 91
    - design-time example configuration, 104
    - IComponent and, 86–89
    - IContainer and, 89–90
    - IServiceContainer and, 91–93
    - IServiceProvider and, 91–93
    - ISite and, 90–91
    - overview of, 86
  - System.Configuration namespace, 652–654
  - System.Diagnostics namespace, 657–658
  - System.Security.Cryptography namespace, 489
- T**
- Tab.config file, 66–68
  - TDD (Test-Driven Development) methods, 575
  - Test-Driven Development (TDD) methods, 575
  - TextExceptionFormatter, 263, 265–267
  - TextFormatter
    - configuring, 381–383
    - customizing, 363–366
    - customizing Sinks, 362
    - defined, 356
    - overview of, 357–359
  - TextFormatterWithEIDictionary, 366
  - Time-based expirations, 191, 192–195
  - TimeStampToken, 360
  - TokenFunction
    - customizing, 363–366
    - DictionaryToken, 361–362
    - IToken interface, 416–417
    - KeyValueToken, 361
    - overview of, 359–360
    - TimeStampToken, 360
  - TraceListeners, 659
  - Tracing, 366–376
    - activity identifiers and, 370–375
    - adding to applications, 393–394
    - Category stack and, 375
    - logging while, 375–376
    - LogSink using, 347–348
    - overview of, 366–370
  - Transactions
    - Data Access Application Block, 166–169
    - Data Mapping Application Block, 585–587
    - DataBaseWrapper and, 584–585
    - programmatically setting level of, 617–620
  - TransformerProvider, 36, 40
  - Transformers
    - adding to applications, 70
    - ConfigurationBuilder managing, 49
    - configuring, 65, 77–79
    - custom, 8–9, 40–44, 76–79
    - defined, 3
    - design, 37
    - ITransformer and
      - TransformerProvider, 35–36
    - MvpSerializerTransformer, 38–40
    - overview of, 35–36
    - serializing and deserializing with, 9–10
    - XmlSerializerTransformer, 36–38

## 684 ■ Index

Type Selector utility  
 configuring custom BackingStores, 225–226  
 configuring custom Transformers, 78–79  
 configuring exception policies, 292–294  
 Typed DataSets, 152–155  
 TypeName property  
 custom exception handlers, 277, 299  
 custom Sinks, 352  
 custom StorageProviders, 74–76  
 database providers, 139–141  
 design-time example configuration, 102

**U**

UIHierarchyService, 94  
 Unit tests, 574–576  
 Update command,  
   DataTableMapping, 611–612  
 UpdateBehavior method, 160–166  
 UpdateDataSet method, 159–166, 583, 585  
 Updating data, Data Access  
   Application Block, 158–166  
   ExecuteNonQuery method, 158–159  
   UpdateDataSet method, 159–166  
 UserRoleManager, 402, 461–464

**V**

Vendors, building application block, 534  
 Visual Studio, 597

**W**

Windows Management  
   Instrumentation (WMI) event, 191–192  
 WMI (Windows Management Instrumentation) event, 191–192  
 WMILogSink, 350, 386

WrapHandler, 256–259, 294  
 WrapperException, 278, 280–282, 288–290  
 Wrapping exceptions  
   HandlingInstanceID, 258–259  
   overview of, 294  
   using WrapHandler, 256–258  
 Write method, 311–314, 390–392  
 WriteConfiguration method, 52–55  
 Writing configuration data,  
   Configuration Application Block, 4–7  
*Writing Secure Code* (Howard and LeBlanc), 491

**X**

XML, generic DataSets supporting, 150  
 XmlExceptionFormatter, 263, 267  
 XmlFileBackingStore  
   configuring, 226–228  
   creating custom BackingStore, 213–219  
   enabling encryption for, 226–228  
 XmlFileCacheStorageData, 212–213  
 XmlFileCacheStorageNode, 215–216  
 XmlFileCommandWrapper, 133–135  
 XmlFileParameter, 133–135  
 XmlFileStorageProvider, 9, 15–16, 66–68  
 XmlIncludeType  
   adding XmlSerializerTransformer to configuration section, 77–78  
   configuring new BackingStore, 217–218  
   design-time example configuration, 104  
   overview of, 47–48  
 XmlIncludeTypeService, 94, 105  
 XmlSerializer  
   adding MvpSerializerTransformer to configuration section, 78  
   MvpSerializerTransformer, 38–40  
   reading and writing configuration data, 4–7

- XmlSerializerTransformer, 36–38
- XmlSerializerTransformer
  - configuring, 77–78
  - defining, 9–10
  - deserializing XML with, 35
- MvpSerializerTransformer
  - compared to, 38–39
  - overview of, 36–38

