APPENDIX G

ServiceModelEx Catalog

CollectionExtensions

Category

C#

Description

Augments the LINQ collection extensions with additional methods.

See Also

`ArrayExtensions`

ArrayExtensions

Category

C#

Description

Provides all the LINQ extensions that return a collection except when operating on an array, in which case they return an array. Augments the LINQ collection extensions with additional methods for arrays.

See Also

`CollectionExtensions`

InProcFactory<S,I>

Category

Hosting

Description

Instantiates a .NET class over WCF. No config file or host is required.

See Also

`DuplexClientBase<T,C>`
`ServiceThrottleHelper`
`NetNamedPipeContextBinding`
`ServiceHost<T>`
`GenericResolver`

WcfWrapper<S,I>

Category

Hosting

Description

Wraps the `InProcFactory<S,I>` to provide regular .NET syntax for creating and managing the service. The result is indistinguishable from plain .NET programming, yet the object is instantiated and called over WCF.

See Also

`NetNamedPipeContextBinding`

ServiceHost<T>

Category

Hosting

Description

Adds type safety to `ServiceHost`, adds support for string-based base addresses, streamlines and automates metadata publishing, automates handling known types, adds any error handler extensions, adds the `ErrorHandlerBehavior` to the service, including exception details in faults, verifies all queues, purges queues on shutdown in debug mode, enables security audits, and reads throttle values.

See Also

`GenericResolver`
`LogbookManager`
`LogbookManagerClient`
`ErrorHandlerHelper`
`DebugHelper`
`QueuedServiceHelper`

MetadataHelper

Category

Contracts

Description

Performs type-safe contract queries of service metadata.

DataContractSerializer<T>

Category

Data Contracts

Description

Generic type-safe wrapper of `DataContractSerializer`.

GenericResolver, GenericResolverBehaviorAttribute, GenericResolverInstaller

Category

Data Contracts

Description

Automates handling known types. Can add specific known types or implicitly all types used in the application. Installed using host or proxy extension method, or declaratively on the service class.

See Also

```
ServiceHost<T>
InProcFactory<S,I>
```

ServiceThrottleHelper

Category

Instance Management

Description

Sets throttle extensions for `ServiceHost`.

IInstanceStore<ID,T>

Category

Durable Services

Description

Defines an interface for any instance store.

See Also

```
FileInstanceStore<ID,T>
```

`TransactionalMemoryStore<ID,T>`

FileInstanceStore<ID,T>, FilePersistenceProvider, FilePersistenceProviderFactory

Category

Durable Services

Description

File-based store; corresponding provider and provider factory.

See Also

`IInstanceStore<ID,T>`
`TransactionalMemoryProviderFactory`
`TransactionalInstanceProviderFactory`

MemoryProvider, MemoryProviderFactory

Category

Durable Services

Description

Abstract provider and provider factory for in-memory store.

See Also

- `TransactionalMemoryProviderFactory`
- `TransactionalInstanceProviderFactory`
- `TransactionalMemoryProvider`
- `TransactionalInstanceProvider`

TransactionalMemoryStore<ID,T>, TransactionalMemoryProvider, TransactionalMemoryProviderFactory

Category

Durable Services

Description

In-memory transactional store shared across clients; corresponding provider and provider factory.

See Also

- IInstanceStore<ID,T>
- TransactionalInstanceProviderFactory
- FilePersistenceProviderFactory

TransactionalInstanceStore<ID,T>, TransactionalInstanceProvider, TransactionalInstanceProviderFactory

Category

Durable Services

Description

Per client session in-memory transactional store; corresponding provider and provider factory.

See Also

- IInstanceStore<ID,T>
- TransactionalMemoryProviderFactory
- FilePersistenceProviderFactory

InstanceContext<T>, DuplexClientBase<T,C>, DuplexChannelFactory<T,C>

Category

Operations

Description

Provide type safety for the client with callback contracts, via either a proxy class or a channel factory.

See Also

`SecureDuplexClientBase<T,C>`

DebugHelper

Category

Faults

Description

Constant to include exception details in faults in debug mode only, extension for `FaultException<ExceptionDetail>` to extract the CLR exception.

ErrorHandlerHelper

Category

Faults

Description

Used for promoting exceptions and logging errors to `LogbookManager`.

See Also

`ErrorHandlerBehaviorAttribute`
`CallbackErrorHandlerBehaviorAttribute`

ErrorHandlerBehaviorAttribute, CallbackErrorHandlerBehaviorAttribute

Category

Faults

Description

Used for declaratively promoting exceptions and logging errors to `LogbookManager`; attributes for service and callback.

See Also

```
ServiceHost<T>
ErrorHandlerHelper
```

ILogbookManager, LogbookManagerClient, LogbookManager

Category

Faults

Description

Service contract for logging errors; proxy for the contract; the logbook service itself.

See Also

```
ErrorHandlerHelper
```

BindingRequirementAttribute

Category

Transactions, General

Description

Used for declaratively enforcing the Client/Service propagation mode on the service or the callback object, declaratively insisting on an intranet-only binding, and declaratively requiring reliability.

ResourceManager, TransactionalLock

Category

Transactions

Description

Helper classes useful when implementing a transactional resource manager.

See Also

`Transactional<T>`

Transactional<T>

Category

Transactions

Description

A generic volatile resource manager. Performs transactions on any serializable type without compromising on the type programming model.

See Also

`ResourceManager`
`TransactionalLock`

TransactionalCollection<C,T>

Category

Transactions

Description

A generic abstract volatile resource manager collection, useful when implementing a custom transactional collection.

See Also

```
Transactional<T>
```

TransactionalArray<T>, TransactionalDictionary<K,T>, TransactionalLinkedList<T>, TransactionalList<T>, TransactionalQueue<T>, TransactionalSortedDictionary<K,T>, TransactionalSortedList<K,T>, TransactionalStack<T>

Category

Transactions

Description

The array and all the collections in .NET as transactional volatile resource managers. The collections and the array are polymorphic with the nontransactional built-in collections and are completely interchangeable.

See Also

```
TransactionalCollection<C,T>
```

TransactionalBehaviorAttribute

Category

Transactions

Description

Enables transparent transactional programming on a per-session service. Negates the need to use a per-call service just for transactions.

See Also

`TransactionalMemoryStore<ID,T>`
`TransactionalMemoryProvider`
TransactionalMemoryProviderFactory
`Transactional<T>`
`TransactionalDictionary<K,T>`
`NetNamedPipeContextBinding`

OneWayClientBaseAsync<T>

Category

Concurrency management

Description

A proxy helper class that issues true asynchronous fire-and-forget one-way calls. Requires all operations on contract to be one-way.

FormHost<F>

Category

Concurrency and Windows Forms

Description

Turns a Windows Forms form into a service, without affecting the form's implementation. Service calls simply update the UI.

SafeButton, SafeLabel, SafeListBox, SafeProgressBar, SafeStatusBar, SafeTextBox, SafeTrackBar

Category

Concurrency and Windows Forms

Description

Controls that can be accessed by any service or callback call, on any thread, to update the UI.

See Also

`FormHost<F>`

ThreadPoolSynchronizer, ThreadPoolBehaviorAttribute, CallbackThreadPoolBehaviorAttribute, ThreadPoolHelper

Category

Custom Synchronization Context

Description

Custom thread pool synchronization context; attributes to apply it declaratively on the service or a callback object; helper class to close the threads in the pool.

See Also

`AffinitySynchronizer`
`PrioritySynchronizer`
`ThreadAffinityBehaviorAttribute`
`PriorityCallsBehaviorAttribute`

AffinitySynchronizer, ThreadAffinityBehaviorAttribute, CallbackThreadAffinityBehaviorAttribute, HostThreadAffinity

Category

Custom Synchronization Context

Description

Custom synchronization context providing thread affinity; attributes to apply it on the service and the callback; extension methods to apply at the host level.

See Also

`ThreadPoolSynchronizer`
`ThreadPoolBehaviorAttribute`

`CallbackThreadPoolBehaviorAttribute`

PrioritySynchronizer, PriorityClientBase<T>, PriorityContext, PriorityCallsBehaviorAttribute

Category

Custom Synchronization Context

Description

Custom synchronization context executing calls by priority; proxy class to pass priority in headers; helper class to extract priority from headers; attribute to apply priority processing on service.

See Also

`ThreadPoolSynchronizer`

`ThreadPoolBehaviorAttribute`

`GenericContext`<T>

`HeaderClientBase`<T,H>

AsyncContextSynchronizer, AsyncContextBehaviorAttribute, HostAsyncContextExtensions

Category

Custom Synchronization Context

Description

Custom synchronization context that reinstates the flow of all WCF contexts from the currently executing thread to the continuation thread when using the async/await pattern (TAP) within a service operation.

See Also

`ThreadPoolSynchronizer`

`ThreadPoolBehaviorAttribute`

HostThreadAffinity

QueuedServiceHelper

Category

Queued Services

Description

Used for verifying all queues on the client, verifying a queue for an endpoint as an extension, and purging a queue.

See Also

ServiceHost<T>

ClientResponseBase<T>, ServiceResponseBase<T>, ResponseContext

Category

Queued Services

Description

Framework for supporting a response service. Proxy class for the client to pass a response address and method IDs; a proxy class for the service to automate calling the response service proxy; response context for direct manipulation.

See Also

GenericContext&lt;T&gt;
HeaderClientBase&lt;T,H&gt;

CredentialsManager, AspNetSqlProviderService

Category

Security and ASP.NET providers

Description

Smart client application for managing the ASP.NET credentials store; WCF service wrapping the ASP.NET providers.

See Also

```
MetadataHelper
```

SecureClientBase<T>, SecureDuplexClientBase<T,C>, SecurityBehaviorAttribute, SecurityHelper, ServiceSecurity

Category

Declarative Security

Description

Declarative security framework. Declarative security proxy class for regular and duplex calls; attribute for declarative support for the service; declarative security extension methods for regular and duplex type-safe channel factories; declarative security extension methods for the host; extension methods for automatic impersonation; enumeration for declarative security scenarios.

See Also

```
DuplexClientBase&lt;T,C&gt;
DuplexChannelFactory&lt;T,C&gt;
```

HeaderClientBase<T,H>, HeaderChannelFactory<T,H>, GenericContext<T>

Category

Message Headers

Description

Streamlines passing custom out-of-band parameters to the proxy or channel factory; helper class for extracting and setting the information in the headers.

See Also

```
InterceptorClientBase&lt;T&gt;
InterceptorChannelFactory&lt;T&gt;
```

ContextClientBase<T>, ContextManager

Category

Context Bindings

Description

Streamlines passing custom out-of-band parameters to the proxy over the context bindings; helper class for extracting and setting the information in the context and managing standard IDs.

See Also

```
GenericContext&lt;T&gt;
HeaderClientBase&lt;T,H&gt;
```

NetNamedPipeContextBinding, NetNamedPipeContextBindingElement, NetNamedPipeContextBindingCollectionElement

Category

Context Bindings

Description

Adds the context protocol to the IPC binding; helper classes (not required for direct use) for enabling administrative configuration.

See Also

```
ContextClientBase&lt;T&gt;
ContextManager
InProcFactory&lt;S,I&gt;
```

AsyncContext, AsyncContextScope

Category

C#

Description

A context scope and helper class that reinstates the flow of all WCF contexts within a service operation from the currently executing thread to the continuation thread when the scope is closed.

See Also

```
AsyncContextSynchronizer
```

DiscoveryHelper

Category

Discovery

Description

Helper classes for finding available base addresses, enabling discovery on the host, discovering service addresses based on cardinality and scope.

See Also

```
DiscoveryFactory
```

DiscoveryFactory

Category

Discovery

Description

A discovery helper class. It is both a class factory for creating proxies to discovered services and creating discoverable hosts. Can create proxies based on cardinality and scope, or discover the MEX endpoint for address and binding.

See Also

```
DiscoveryHelper
```

AddressesContainer<T>

Category

Discovery

Description

Elaborate container for endpoint addresses, offers multiple indexes and enumerators.

See Also

```
AnnouncementSink<T>
DiscoveredServices<T>
```

AnnouncementSink<T>

Category

Discovery

Description

Encapsulates an announcement endpoint yet specific to contract type. Allows accessing announced services addresses list via the base class of AddressesContainer<T> as well as subscribing to events notifying of services.

See Also

`AddressesContainer<T>`
`DiscoveredServices<T>`
`DiscoveryPublishService<T>`

DiscoveredServices<T>

Category

Discovery

Description

Continuously discovers in the background services using the discovery endpoint. Allows accessing discovered services addresses list via the base class of `AddressesContainer<T>`.

See Also

`AddressesContainer<T>`
`AnnouncementSink<T>`
DiscoveryPublishService<T>

IDiscovery, IDiscoveryCallback, IAnnouncements, DiscoveryService, DiscoveryFactory, DuplexDiscoveryClient, IDiscoverySubscription, IAnnouncementsSubscription, DuplexAnnouncementSink<T>

Category

Discovery

Description

A small framework that adds WCF-like discovery support over TCP. Discoverable services use the helper methods of `DiscoveryFactory` to publish their discoverable endpoints to `DiscoveryService`. The client uses `DuplexDiscoveryClient` to discover the service. The client can use the helper methods of `DiscoveryFactory` to streamline the interaction. Similar support for announcements as with regular discovery.

See Also

- `DiscoveryHelper`
- `DiscoveryFactory`
- `AnnouncementSink<T>`

PersistentSubscriptionServiceClient, IPersistentSubscriptionService, ISubscriptionService, PersistentSubscription, PublishService<T>, SubscriptionManager<T>, PersistentSubscriptionManager

Category

Publish-Subscribe

Description

Framework for implementing publish-subscribe solution. Proxy class for adding persistent subscribers; service contracts for the publishing and subscription services; base classes for the implementation of the publishing and subscription services; demo application for managing persistent subscriptions.

See Also

- `MetadataHelper`
- `DuplexClientBase&lt;T,C&gt;`
- `DuplexChannelFactory&lt;T,C&gt;`

DiscoveryPublishService<T>

Category

Publish-Subscribe and Discovery

Description

Base class for a discovery-enabled publish-subscribe service. DiscoveryPublishService<T> supports a discover endpoint so that publishers can discover it. DiscoveryPublishService<T> provides an announcement endpoint for subscribers, and it continuously discovers subscribers with a discovery endpoint. Also provides for helper methods for hosts and publishers.

See Also

AnnouncementSink<T>
DiscoveredServices<T>

GenericInvoker, OperationInterceptorBehaviorAttribute, ServiceInterceptorBehaviorAttribute

Category

Generic Interceptor

Description

Framework for easy injection of an interceptor to do pre- or post-call processing; attributes for operation- and service-level declarative use.

See Also

InterceptorClientBase&lt;T&gt;
SecurityCallStackInterceptor
OperationSecurityCallStackAttribute
SecurityCallStackBehaviorAttribute

InterceptorClientBase<T>, InterceptorChannelFactory<T>

Category

Generic Interceptor

Description

Proxy base class and channel factory for easy interception of client calls.

See Also

`HeaderClientBase&lt;T,H&gt;`
`OperationInterceptorBehaviorAttribute`
`ServiceInterceptorBehaviorAttribute`

SecurityCallFrame, SecurityCallStack, SecurityCallStackContext, SecurityCallStackClientBase<T>, SecurityCallStackInterceptor, OperationSecurityCallStackAttribute, SecurityCallStackBehaviorAttribute

Category

Security Identities

Description

Framework for propagating the stack of callers' identities in the message headers; definitions of the call stack and supporting types; proxy class enabling the client to push its identity; interceptor for managing the call stack; attributes to inject the interceptor at the operation or service level.

See Also

`GenericInvoker`
`OperationInterceptorBehaviorAttribute`
`ServiceInterceptorBehaviorAttribute`
`GenericContext&lt;T&gt;`
`InterceptorClientBase&lt;T&gt;`

FabricRuntime, ActorRegistration

Category

Service Fabric

Description

Models Azure Service Fabric runtime registration for Reliable Services and Reliable Actors.

See Also

`Actor`
`Actor&lt;S&gt;`
`StatelessServiceBase`
`StatefulServiceBase`
`ApplicationManifestAttribute`

ApplicationManifestAttribute

Category

Service Fabric

Description

Models Azure Service Fabric's `applicationmanifest.xml` file providing Reliable Actor and Reliable Service address validation for modeled instances.

See Also

`Actor`
`Actor&lt;S&gt;`
`StatelessServiceBase`
`StatefulServiceBase`

ServiceTestBase

Category

Service Fabric

Description

Test framework that supports granular testing of Reliable Actors and Reliable Services beyond the unit test level supporting multiple levels of intermediate integration testing.

See Also

```
Actor
Actor&lt;S&gt;
StatelessServiceBase
StatefulServiceBase
ApplicationManifestAttribute
```

IActor, ActorId, ActorBase, Actor, Actor<S>, ActorProxy, ActorGarbageCollectionAttribute, ReadonlyAttribute, CompletesActorInstanceAttribute, VolatileActorStateProviderAttribute, KvsActorStateProviderAttribute, ReliableDictionaryActorStateProviderAttribute

Category

Service Fabric

Description

Framework for modeling Azure Service Fabric's Reliable Actors.

See Also

```
IService
StatelessServiceBase
StatefulServiceBase
```

ServiceProxy

ServicePartitionClient

WcfCommunicationClient<T>

WcfCommunicationClientFactory<T>

ServiceFabricClientBase&lt;T&gt;

"ApplicationManifestAttribute" on page 965

IService, StatelessServiceBase, ServiceProxy, ServicePartitionClient, WcfCommunicationClient<T>, WcfCommunicationClientFactory<T>, ServiceFabricClientBase<T>, ServicePartitionResolver

Category

Service Fabric

Description

Framework for modeling Azure Service Fabric's default and WCF Reliable Services.

See Also

IActor

ActorId

ActorBase

Actor

Actor&lt;S&gt;

ActorProxy

ApplicationManifestAttribute