


Transient Fault Handling With ADO.NET

Herve Roggero
<http://www.herveroggero.com>
hroggero@bluesyntax.net



pluralsight 
hardcore developer training

Topics Covered

- **Need for a Retry Framework**
 - Why use a retry framework in the first place?
- **Transient Fault Handling Application Block**
 - Introduction to Microsoft's implementation of a retry framework
-  ■ **Configure and Extend**
 - Demo on how to configure and extend the Transient Fault Handling Application Block

Why?

- **Cloud Computing Needs**

- Expect failures

- **Transient Errors**


- A little retry can go a long way!



- **Code your own?**

- Lessons learned incorporated in a retry framework, but needs change over time

Transient Fault Handling Application Block

- **Focuses on Transient Errors and Throttling Conditions**
 - This module focuses on Transient Errors only
- **Detection Strategy**
 - Do we care about the error?
- **Retry Strategy**
 - A model that controls the retry behavior (fixed, incremental, exponential)
-  **Policy**
 - A policy implements a specific **detection strategy**
 - A policy defines the **retry strategy** to implement

<http://tinyurl.com/lmhrj6o>

A Few Classes...

Class	Purpose
RetryPolicy	A retry policy that includes the RetryStrategy and ErrorDetectionStrategy properties
ReliableSqlConnection	A SqlConnection wrapper that implements retry operations
EnterpriseLibraryContainer	A reference to the enterprise configuration
FixedInterval	One of the retry strategies available
RetryManager	Accesses the default retry polies and strategies

A few Methods...

Method	Purpose
SqlConnection.OpenWithRetry()	Extension method on SqlConnection that opens a database connection using a retry policy
ReliableSqlConnection.Open()	Opens a database connection using a retry policy
RetryPolicy.ExecuteAction()	Executes a method within the context of a retry policy
ITransientErrorDetectionStrategy. IsTransient()	Determines whether or not an exception is transient given a specific error detection strategy
SqlCommand. ExecuteNonQueryWithRetry()	Extension method on SqlCommand that executes a query using a retry policy

Summary

- Why implement a retry strategy?
- Detection strategy and retry policy
- Install and configure the Transient Fault Handling Application Block
- Implement retry logic with ADO.NET

