Managing Concurrency and Transactions

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

hardcore developer training

Topics Covered

- **Autocommit vs. Implicit vs. Explicit Transactions**
  - Overview of autocommit, implicit and explicit transactions
- **Database, Local and Distributed Transactions**
  - Introduction to the various types of transactions
- **Managing Optimistic Concurrency**
  - Options for managing optimistic concurrency
- **Implementing Local Transactions with SqlConnection**
  - Demo on how to implement the SqlTransaction class in .NET code

Autocommit vs. Implicit vs. Explicit Transactions

- **Autocommit = Default**
  - By default, the database automatically commits statements
- **Implicit requires COMMIT or ROLLBACK**
  - In implicit mode, a transaction starts automatically
- **Use BeginTransaction to declare an Explicit transaction**
  - In explicit mode, a transaction requires a COMMIT

Database, Local and Distributed Transactions

- **Database Transactions**
  - BEGIN TRAN and COMMIT TRAN in SQL Server/SQL Database
  - Server-side transactions offer highest performance
- **Local Transactions**
  - Within .NET using TransactionScope or BeginTransaction
- **Distributed Transactions**
  - Involves the Distributed Transactions Coordinator (DTC)
  - Applies when multiple databases/systems are involved
  - Use TransactionScope to promote to full distributed transactions

A transaction that does not involve DTC is called a lightweight transaction

TransactionScope is declared in System.Transactions

Managing Optimistic Concurrency

- **Concurrency Models**
  - No concurrency checks: Last update wins
  - Pessimistic Concurrency: Lock established while records are being changed
  - Optimistic Concurrency: Lock established during the commit phase
- **Methods for Optimistic Concurrency**
  - Timestamp
  - WHERE clause

Summary

- Autocommit, Explicit and Implicit transactions
- Database, Local and Distributed transactions
- Options to manage concurrency checks (timestamp, WHERE)
- BeginTransaction method on SqlConnection
- TransactionScope class