Database access with ODBC and ADO.NET

Lecture 12

CS 638 Web Programming

Lecture overview

- Introduction to ODBC
  - Important classes
- Short introduction to ADO.NET

What is ODBC?

- Stands for Open Database Connectivity
- An API (application programmer's interface) for accessing relational databases from programs
  - Proposed by Microsoft in 1992, now standard
  - Widely supported by databases and programming environments on various operating systems
- Uses database-specific ODBC driver on client computer between application and database
  - Converts method calls by the applications to SQL commands the database understands
  - Converts database replies to native data for application

Typical application structure

- Define connection – the connection string
- Open the connection
  - May throw exceptions
- Issue commands, receive / transmit data
  - May throw exceptions
  - Can have multiple commands, but not at the same time
- Close the connection

Connection string

- One long string with everything ODBC needs to know to initiate a connection to the database
  - DRIVER={MySQL ODBC 3.51 Driver};
  - SERVER=oberon.cs.wisc.edu;
  - PORT=3400;
  - DATABASE=databaseName;
  - USER=userName;
  - PASSWORD=myPassword;
  - OPTION=3;"
- Switch the user name, database name, and password to what you received from the TA

The driver

- {MySQL ODBC 3.51 Driver}
- Must be installed on your system
  - Is installed on the machines from the lab
  - Using wrong version can cause trouble
  - You can download it from

    http://dev.mysql.com/downloads/connector/odbc/3.51.html

Connection object

- Instance of System.Data.Odbc.OdbcConnection
  - Constructor expects connection string
- Important methods
  - Open – open the connection
  - CreateCommand – create command objects
  - Close – close the connection
- Important attribute
  - State
- Remember to close an open connection
  - Nice use of "finally"
- Use open / close judiciously as operations have high overhead

Command object

- Instance of System.Data.Odbc.OdbcCommand
- Can use constructor or the connection's CreateCommand method
- If you use the constructor, you need to assign to the Connection attribute the connection object to the
- Assign to CommandText attribute the text of the SQL command you want to execute
  - With parameterized queries can specify the various parameters of the command separately using Command.Parameters.AddWithValue()

Parameterized queries

- SQL is picky about quoting text literals, parameterized queries help
- Important for preventing SQL injection attacks if parts of command can come from untrusted source (user)
  - Example:
    - "select c2 from t where c1 = "+name+";"
    - What happens if name is "1; delete from t where 1" ?

Executing commands

- ExecuteNonQuery()
  - Used for executing SQL commands which do not return data such as insert, delete, and update
- ExecuteScalar()
  - Used for executing queries that return exactly one column and one row
    - For example getting a count of some set of rows
  - Return type is Object
    - Return value must be converted to the appropriate type for example using System.Convert.ToInt32()
- ExecuteReader()
  - Used for executing queries that return multiple rows

ExecuteReader()

- Returns System.Data.Odbc.OdbcDataReader
  - Provides access to result one row at a time
  - Read() method moves reader to next row
  - Columns within the current row accessed using []
    - Can use either position or column name as index
    - Return type is Object, must cast results explicitly using methods from System.Convert
  - Close() method ends reader (but does not close the connection)
- No other command associated with the connection can be executed while reader active

ADO.NET

- Microsoft's latest interface to databases
  - Part of the .NET framework
  - Works with wide range of databases
- Offers DataReader that is similar to ODBC Reader
- Offers DataSet – memory resident object-oriented representation of database
  - Supports fast random reads and writes
  - Large memory overhead
  - Uses DataAdapter to read database and to commit changes by sending a batch of updates
    - Disconnected operation model: no open connection while application works with the DataSet

Databases

Lectures 10-12

CS 638 Web Programming

Specific technologies covered

- SQL
  - SELECT command enables powerful queries
- ODBC

Important concepts

- Three-tiered structure for applications
- Representing entities and relationships using the tables of relational databases
  - Use of joins to connect tables
- Using a key to identify a row within a table
  - "Foreign key" is key for another table
- Declarative languages (as opposed to imperative programming languages)
- Conceptual evaluation of query vs. actual execution