



# Developing and optimizing applications with Oracle and .NET

Tonči Sviličić  
Solution Architect, .NET Technologies

# ||| Agenda

- Oracle and .NET Intro
- Oracle Developer Tools for Visual Studio .NET
  - Demo
- Oracle Data Provider for .NET
- .NET Stored Procedures
  - Demo
- Oracle Data Provider for .NET Performance
  - Best Practices - Demo

# Oracle .NET Development

Oracle Developer Tools for VS.NET

Oracle Data Provider for .NET

Oracle Providers for ASP.NET

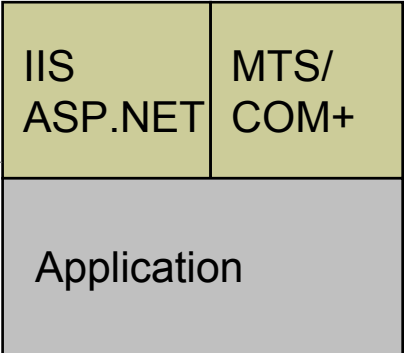


Visual Studio Environment

Develop

C++, C#, VB .NET Application

Deploy



Deploy

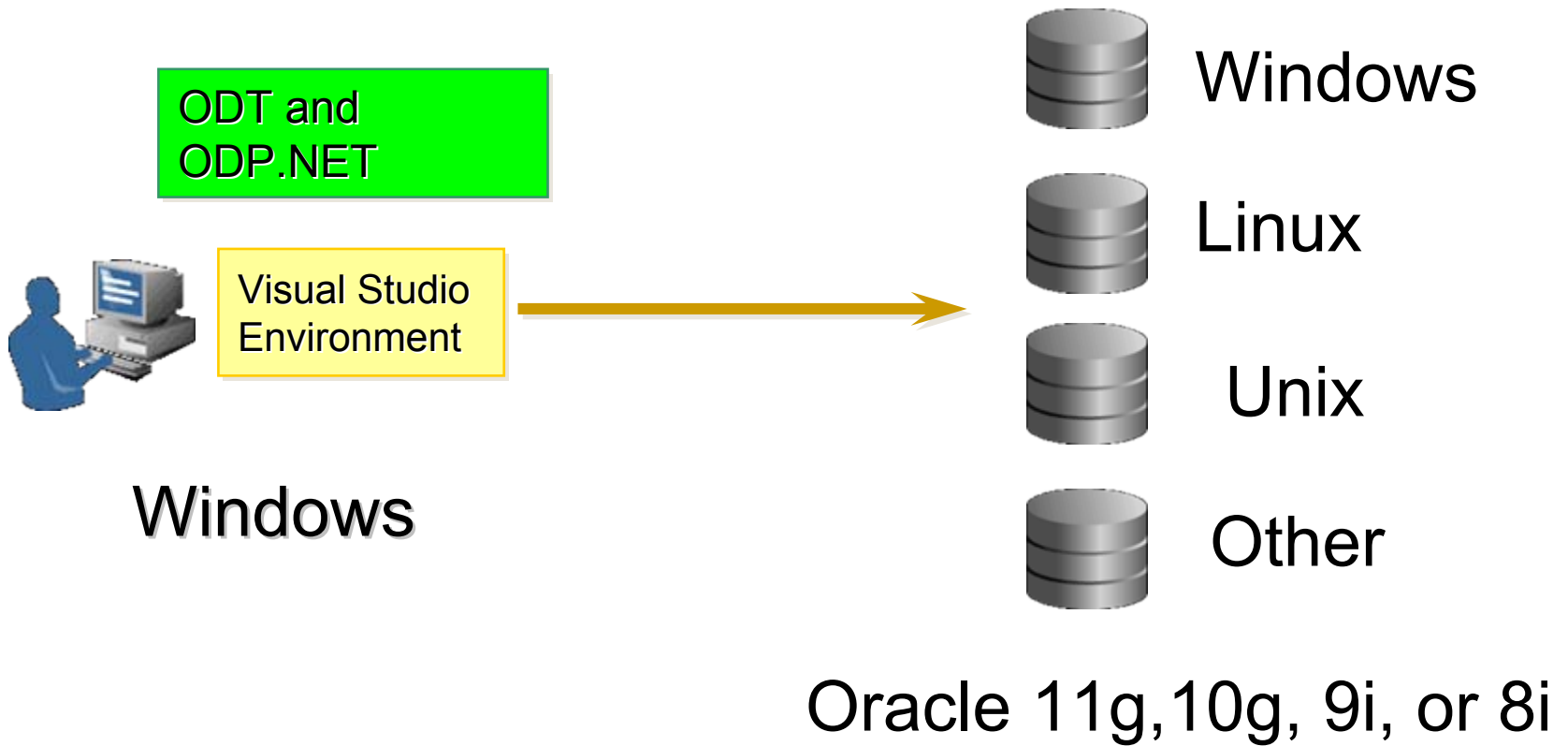
Database Development



Oracle Database Extensions for .NET



# Develop on Windows, Database on any Platform



# Oracle Developer Tools for Visual Studio .NET

- Fully integrated with Visual Studio 2005 and 2008
  - Server Explorer
  - Data Sources Window
  - Data Source Configuration Wizard
  - Dataset Designer
  - Table Adapter Configuration Wizard
  - Query Builder/Query Designer
  - More
- Automatic code generation – Winform and ASP.NET
  - Data Sources Window
  - Integration with SQLDataSource control
  - Makes ASP.NET web application development very easy

# Feature Overview

- Oracle Wizards and Designers
  - Table/View Designer
  - Stored Procedure/Function/package Wizards
  - Grant/Revoke Privileges Wizard
  - UDT Designers
  - UDT Custom Class Code Generation Wizard
  - Table/View Migration Wizard
- Oracle Data Window
- Oracle Database Project
  - Edit and Run SQL scripts
  - SQL\*Plus is built in
  - Source control integration
- Integrated PL/SQL Editor and Debugger

# ■ Feature Overview

- Oracle Query Window
  - Ad Hoc SQL
  - Explain Plan
- .NET Stored Procedure Deployment
- Integrated help system – SQL, PL/SQL keywords
- Easy to use and intuitive
- Stay in Visual Studio for project lifecycle
- Free




# DEMO

Oracle Developer Tools for Visual Studio .NET






# ODP.NET Features

- Full PL/SQL support
    - Packaged, non-packaged, anonymous, autonomous
    - Batch SQL available with anonymous PL/SQL
  - Native Oracle data types
    - REF Cursors, LOBs, XMLType, TimeStamp, etc.
    - Safe type mapping to .NET data types
  - Connection pooling
    - Min, Max, Timeout, Lifetime, Increment, Decrement
- 



# ODP.NET Features

- Transactions
    - Enterprise Services via Oracle Services for MTS
    - Local (Implicit and Explicit)
    - System.Transactions
    - Savepoints
  - Parameters
    - PL/SQL Associative Array data type
    - LOBs
      - Ability to pre-fetch or defer LOB retrieval
    - User-Defined Types (UDTs)
- 



# ODP.NET Features

- XML
  - XMLType data type
  - Query and Save XML from the DB
    - Both relational and object-relational
  - Interoperate with MS XML APIs (e.g. XmlReader)
  - Supports XSLT, XPATH, and XML Schema
- RAC
  - Automatic load balancing
  - Automatic connection cleanup



# ADO.NET 2.0 Support


- Change Notification
- Factory and DbCommon Classes
- Systems.Transactions
- DataAdapter Batch Update
- Schema Discovery APIs
- Connection String Builder
- Server Enumeration

# Top Unique ODP.NET Features

Features	Oracle	SQL 2005
Result Set Retrieval Tuning Options		Yes
No		
Configurable LOB Retrieval	Yes	No
Array Parameters	Yes	No
Proxy Auth. and Secure App Context	Yes	No
Real Application Clusters Support	Yes	No
More Robust Change Notification		Yes
No		

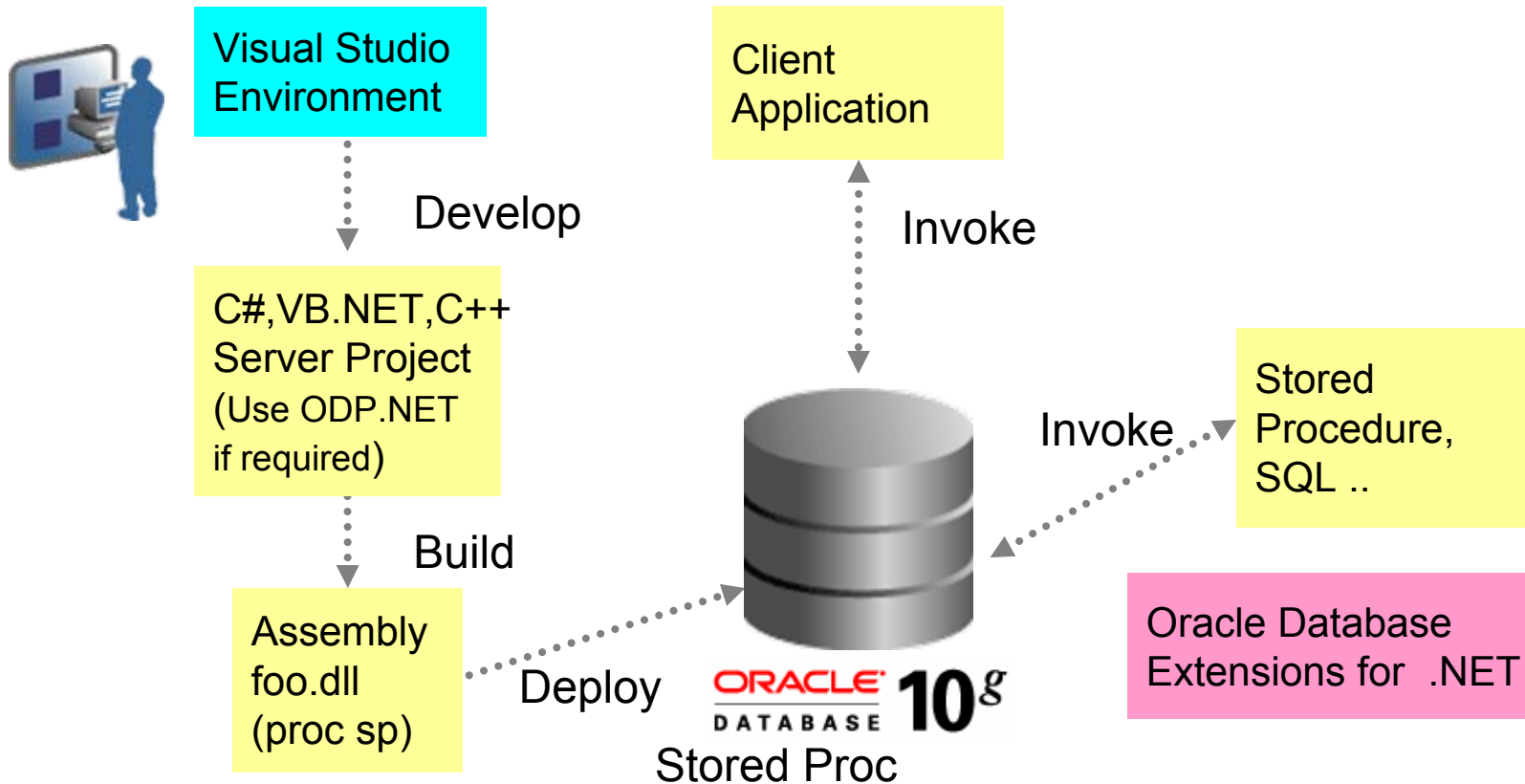


# Oracle Database Extensions for .NET

- Database option
  - DB server on Windows only
    - Both 32-bit and 64-bit supported
  - Oracle Database 10g Release 2 or later
- 

# Oracle Database Extensions for .NET

Oracle Developer Tools for Visual Studio .NET



# ■ When to use

- Ease of development
- Computationally intensive code
- Leverage .NET programming expertise
- Leverage existing .NET code libraries
- Call out to OS or Applications
- When Windows platform only is OK
- When no commits or rollbacks required in SP
- When no distributed transactions are required



# Creating a .NET Stored Procedure

1. Use Visual Studio .NET to build a .NET procedure or function in the usual way
2. If data access is required, use ODP.NET classes
3. Result is a .NET assembly
4. Run .NET Stored Procedure Deployment Wizard (included with Oracle Developer Tools)
5. Call .NET stored procedure as you would any other Oracle stored procedure



# DEMO

.NET Stored Procedures

# ODP.NET Performance Best Practices

- Connection pooling
- Statement caching
- Using FetchSize and RowSize
- Passing array parameters
- Batch SQL
- Ref cursors
- Multiple Active Results Set (MARS)

A vertical bar on the left side of the slide, composed of several colored segments: a small black rectangle at the top, followed by a thin white line, a thin grey line, a thin olive green line, and a thick red line at the bottom.

# DEMO

Optimizing Dana Access Performance with ODP.NET

# Connections

- Keep number of connections at steady state
  - Should never destroy or create large numbers of connections
- Close/Dispose connections explicitly
  - Do not necessarily rely on the garbage collector
  - Recommendation applies to all ODP.NET objects
- Use connection pooling (CP)
  - Creating many connections quickly is resource intensive
  - Make sure to have enough connections in CP
    - Better to have too many connections than too few
  - OS-authenticated CP available with ODP.NET 11g version

# CP Settings

- Min Pool Size = # connections at steady state or average load
- Max Pool Size = # connections at maximum capacity
  - Min and Max Pool Size always obeyed ahead of other CP parameters
- Incr Pool Size = connection growth rate from steady state
- Decr Pool Size = connection decrease rate from steady state
  - Pool size checked every 3 minutes
- Validate Connection = should be false in most cases
  - Extra round trip
  - Writing your own exception handler is likely faster



# ODP.NET Connection Pool Monitoring

- Monitor CP via ODP.NET performance counters and/or tracing
  - Counters available with ODP.NET 11.1.0.6.20 or higher





# Multiple Users – One Connection Pool to Unite Them All

- Use proxy authentication
  - Effective for uniquely-identified users or groups of users, while maintaining CP benefits
  - Has numerous security benefits
  - Creates two sessions, but uses one connection
- Use application context
  - Distinguishes individuals in connection pool with a string identifier
  - Set `OracleConnection.ClientId`, `ActionName`, and/or `ModuleName`
  - Lighter weight than proxy authentication, but does not provide as fine grained security features





# More Information

- .NET Technology Center - .NET Developer Center
    - <http://otn.oracle.com/dotnet>
    - Free downloads of Visual Studio tools, ODP.NET and Oracle Database XE.
    - Whitepapers
    - Sample Code
    - Demo Videos
    - Help Forums
    - “How to” step by step tutorials
    - Latest Oracle on .NET News
  - .NET Blog
    - <http://cshay.blogspot.com/>
- 