

# Introducing Npgsql

A .Net Data Provider for Postgresql

PGCON 2008 – Ottawa – Canada

Francisco Figueiredo Jr.  
francisco@npgsql.org

Hiroshi Saito  
hiroshi@npgsql.org

Josh Cooley  
jbcooley@npgsql.org

# Topics

---

- ▤ History
- ▤ Features
- ▤ Internals
- ▤ About Npgsql in Japan
- ▤ Deployment of Npgsql
- ▤ Gnu/Linux <-> Windows

# Topics

---

## ▣ Samples

- ▣ Console
- ▣ Imagemake Large Object
- ▣ System.Transactions Demo
- ▣ Entity Framework

## ▣ Community

## ▣ Future work

# History

---

- ▄ Work began on Npgsql in May 5, 2002
  - ▄ This was when I first started coding
- ▄ At that time, I was already a big fan of OpenSource philosophy
- ▄ I was only an spectator.
  - I wanted to contribute
- ▄ I wanted a project to work on a network protocol

# History

---

- ▄ I had a look at Postgresql and MySql
- ▄ I decided by Postgresql because of the excellent protocol documentation!
- ▄ Where Npgsql name came from?
  - Miguel de Icaza once asked me that

# History

---

- ▣ By the time Npgsql started, it was common to prefix projects with N (for .Net)
  - Nant
  - Nhibernate
  - NUnit
- ▣ So I thought about: N(.Net)  
pgsql (Postgresql)

# History

---

- ▣ Beginning of 2003, Daniel Morgan from Mono project told me they choose Npgsql as official Postgresql Data Provider
- ▣ That made us be sure we were in the right track! :)

# Features

---

- ▄ Npgsql is done with 100% C#
- ▄ There is no dependency on external libraries
  - This was by design, so it would be easier to install and deploy
- ▄ Works on Mono and MS .Net Runtimes. This means it works on Gnu/Linux, Windows, MacOS and any other supported platform



# Features

---

- ▄ Supports Postgresql server from 7.3 and above
- ▄ Implements protocol versions 2 and 3
- ▄ You can use Npgsql to send usual queries: select, insert, update and delete

# Features

---

## ▄ Has support for Large Object API

- Hiroshi Saito will show an example which uses it

## ▄ Has support for clear and md5 passwords

- Starting with Npgsql2 beta3, we added support for SSPI authentication on windows. Thanks Brar Piening patch

# Internals

---

- ▄ Today we have two versions of Npgsql:
- ▄ We have Npgsql 1.0 which is now only maintained with critical bug fixes
- ▄ And there is Npgsql 2.0 which is being actively developed and targets .Net 2.0+ runtime

# Internals

---

- ▄ We have project files for Nant, Monodevelop, VS.Net 2005 and VS.Net 2008
- ▄ The easiest way to compile Npgsql is grab cvs or source code release, enter Npgsql/src folder and type nant
- ▄ Starting with Beta3, thanks to Jon Hanna's work, Npgsql now returns data on demand to

client

# Internals

---

- ▣ Previous versions got all data from server before making it available to client
- ▣ This behavior has a big impact when dealing with large resultsets

# Examples

---

## ▄ This is a simple console example:

```
public static void Main(String[] args)
{
    NpgsqlConnection conn = new
NpgsqlConnection("Server=127.0.0.1;Port=5432;User
Id=joe;Password=secret;Database=joedata;");
    conn.Open();

    NpgsqlCommand command = new NpgsqlCommand("select
version()", conn);
    String serverversion;

    serverversion = (String)command.ExecuteScalar();
    Console.WriteLine("PostgreSQL server version: {0}",
serverversion);
}
```



# About Npgsql in Japan





# Npgsql Document

**Npgsql (.NET Data Provider)**

非表示 戻る 印刷 オプション(O)

目次(C) キーワード(N) sync toc

- Npgsqlホーム
- ユーザマニュアル
- APIドキュメント
- ダウンロード
- Npgsqlを利用したプロジェクト
- 参考

(global)

- Npgsql
  - LogLevel 列挙
  - NoticeEventHandler デリゲート
  - NotificationEventHandler デリゲート
- NpgsqlCommand クラス
- NpgsqlCommandBuilder クラス
- NpgsqlConnection クラス
- NpgsqlDataAdapter クラス**
  - NpgsqlDataAdapter メンバー
- NpgsqlDataAdapter コンストラクタ
- Properties
- Events

- NpgsqlDataReader クラス
- NpgsqlError クラス
- NpgsqlEventLog クラス
- NpgsqlException クラス
- NpgsqlNoticeEventArgs クラス
- NpgsqlNotificationEventArgs クラス
- NpgsqlParameter クラス
- NpgsqlParameterCollection クラス
  - NpgsqlRowUpdatedEventHandler デ
  - NpgsqlRowUpdatingEventHandler ラ
- NpgsqlTransaction クラス
  - ProtocolVersion 列挙
- ServerVersion クラス

Npgsql.Design

## Npgsql Api ドキュメント

### NpgsqlDataAdapter クラス

このクラスは [Datasets](#) を埋めるselect, update, insert, および delete の数多くのコマンドからのアダプタを表します。

この型の全てのメンバーの一覧は、[NpgsqlDataAdapter メンバー](#) を参照ください。

[System.Object](#)  
[System.MarshalByRefObject](#)  
[System.ComponentModel.Component](#)  
[System.Data.Common.DataAdapter](#)  
[System.Data.Common.DbDataAdapter](#)  
**Npgsql.NpgsqlDataAdapter**

```
[Visual Basic]
NotInheritable Public Class NpgsqlDataAdapter
    Inherits DbDataAdapter
    Implements IDbDataAdapter
```

```
[C#]
public sealed class NpgsqlDataAdapter :
    DbDataAdapter, IDbDataAdapter
```

#### スレッドセーフ

この型の Public static (Visual Basic では **Shared**) のすべてのメンバーは、マルチスレッド操作で安全に使用できます。インスタンスのメンバーの場合は、スレッドセーフであるとは限りません。

#### 必要条件

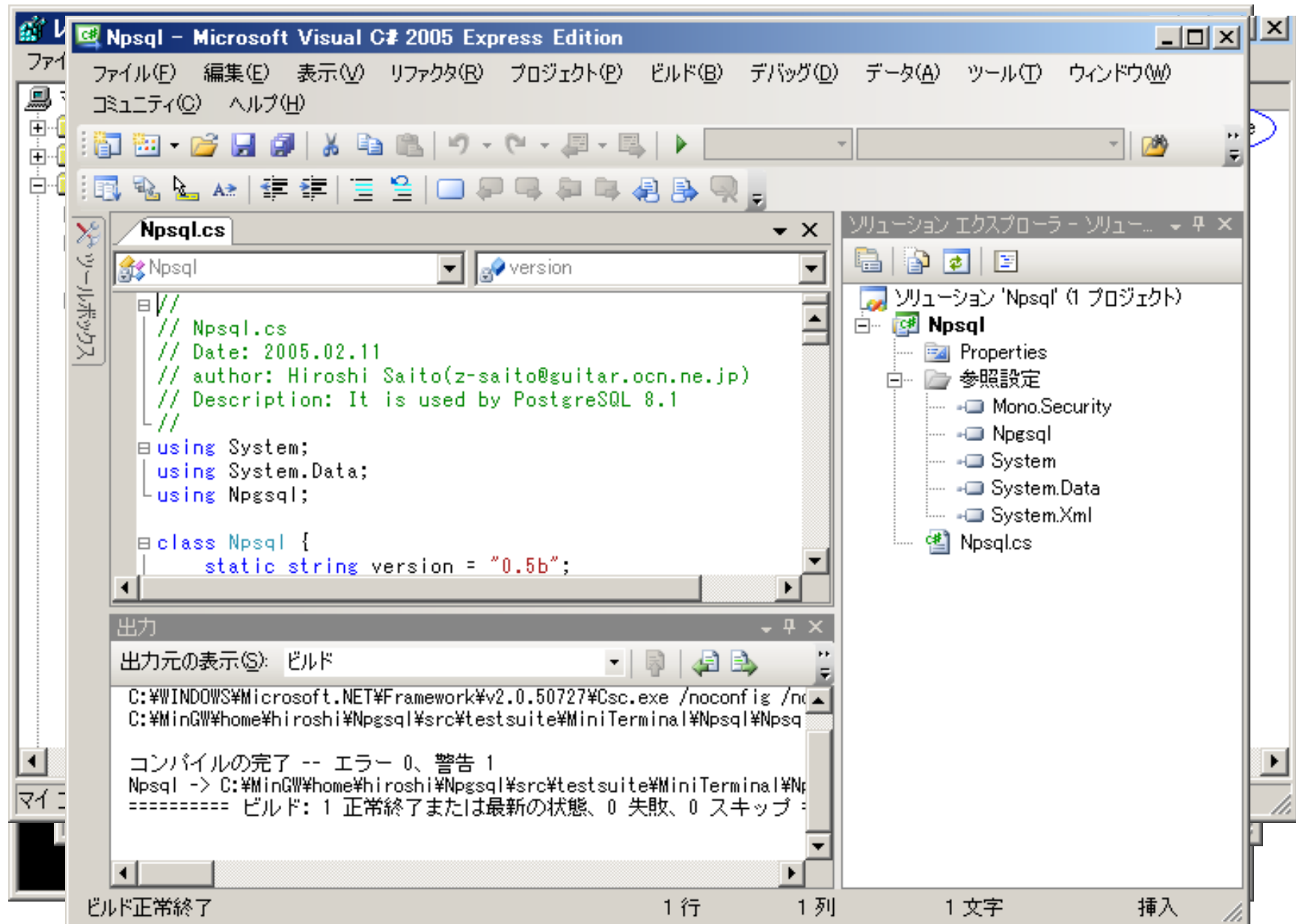
名前空間: [Npgsql](#)  
アセンブリ: Npgsql (Npgsql.dll 内)

#### 参照

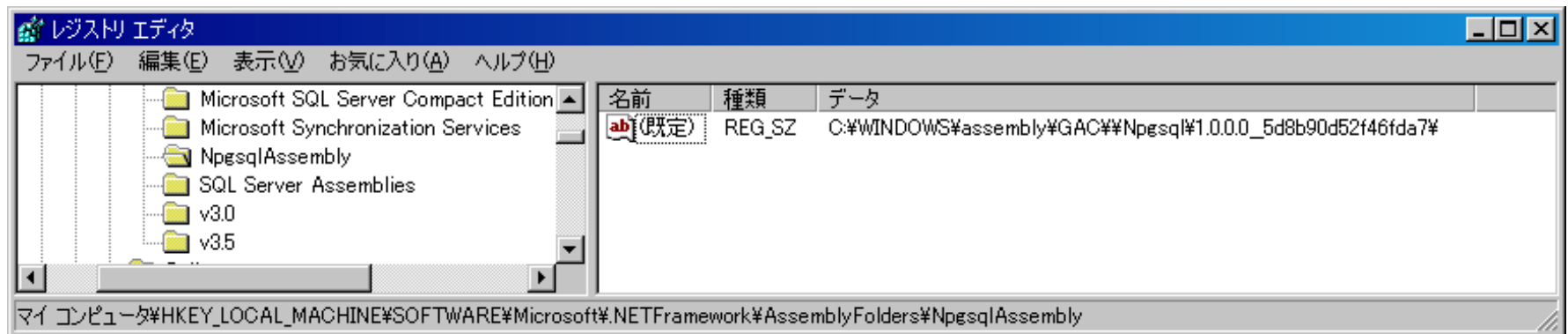
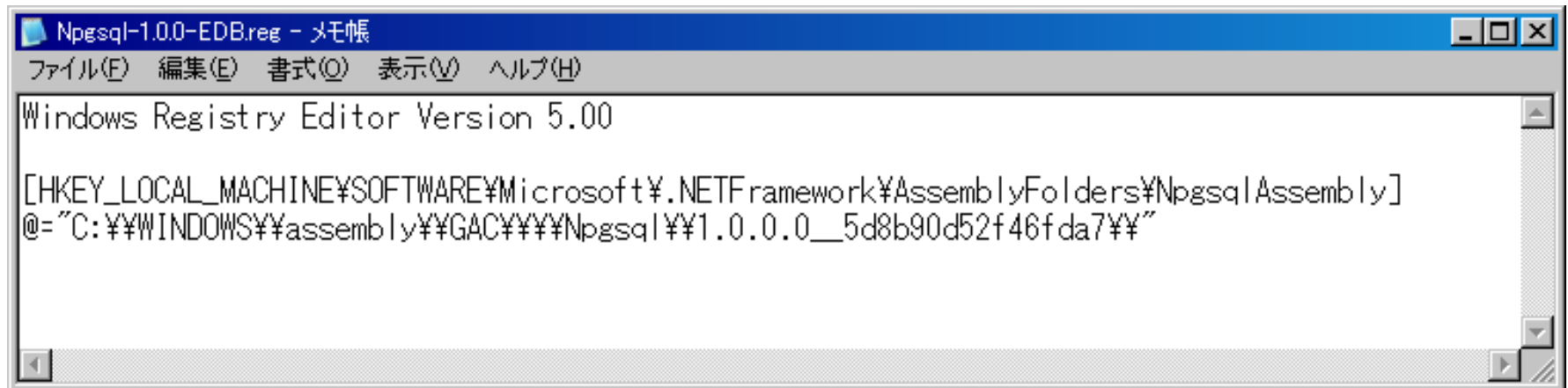
[NpgsqlDataAdapter メンバー](#) | [Npgsql 名前空間](#)

Npgsql 開発チーム 2002-2006

# Deployment of Npgsql



# Deployment of Npgsql



# Linux <-> Windows

## Linux(OpenSUSE 10.1)

接続

PostgreSQLデータベースに接続するための項目を設定してください

1. サーバ名または、IPアドレスの指定  
localhost
2. ログイン情報の指定:  
ユーザ名: saito  
パスワード: \*\*\*\*\*
3. サーバのデータベースを選んで下さい:  
saito リフレッシュ

追加パラメータ

接続タイムアウト: 15 Port: 5432

接続チェック

Ok キャンセル

## Windows-XP SP2

データ接続プロパティ

接続

PostgreSQLデータベースに接続するための項目を設定してください

1. サーバ名または、IPアドレスの指定  
localhost
2. ログイン情報の指定:  
ユーザ名: saito  
パスワード: \*\*\*\*\*
3. サーバのデータベースを選んで下さい:  
saito リフレッシュ

追加パラメータ

接続タイムアウト: 15 Port: 5432

接続チェック

Ok キャンセル

# OpenSUSE 10.1 (Linux)

The screenshot displays the OpenSUSE 10.1 desktop environment. A terminal window is open, showing the execution of a PostgreSQL query. The query results are as follows:

key	ndata
SAITO	11111
SSS	123
DAVE	777
Hiroshi	2020

A dialog box titled "接続" (Connection) is overlaid on the terminal. It is a PostgreSQL connection configuration window. The settings are:

- Server name or IP address: localhost
- User name: saito
- Password: \*\*\*\*\*
- Server database: saito
- Connection timeout: 15
- Port: 5432

The dialog box includes a "接続チェック" (Check connection) button and "Ok" and "キャンセル" (Cancel) buttons.

The terminal window also shows the command `psql` and the output of the `select * from npda` query. The desktop background features icons for "saito のホーム" (Home), "Firefox", "フロッピー・ドライブ" (Floppy drive), and "ゴミ箱" (Trash). The taskbar at the bottom shows the application menu, a window manager icon, and the system tray with the date and time: 11月10日 (金) 11:03.

# Npgsql NET2.0

The screenshot shows the Visual Studio IDE with the following components:

- Menu Bar:** ファイル(F), 編集(E), 表示(V), プロジェクト(P), ビルド(B), デバッグ(D), ツール(T), ウィンドウ(W), コミュニティ(C), ヘルプ(H)
- Toolbox:** Standard Windows Forms controls.
- Code Editor:** Shows the code for `Form1.cs`. The code is as follows:

```
namespace DBFactries
{
    enum provider { Npgsql }
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            DbProviderFactory factory = NpgsqlFactory.Instance;
            //DbProviderFactories.GetFactory("Npgsql")
            DbConnection db = factory.CreateConnection();
            db.ConnectionString = "Server=localhost;Database=hirosi";
            db.Open();
            DbCommand cmd2 = db.CreateCommand();
            cmd2.CommandText = "select * from saito";
            cmd2.ExecuteNonQuery();

            DbDataReader hReader = cmd2.ExecuteReader();
            textBox1.Clear();
            while (hReader.Read())
            {
                textBox1.AppendText(hReader[0].ToString() + "%t");
                textBox1.AppendText(hReader[1].ToString() + "%n");
            }
            hReader.Close();
            db.Close();
        }
    }
}
```
- Form Designer:** Shows a window titled `Form1` with a button labeled `button1` and a text box containing the following data:

dave@postgres	Dave Page
z-saito@postgres	Hiroshi Saito
mag@postgres	Magnus Hagander

pgAdmin III Query - hiroshi @ localhost:5432 \*

ファイル(F) 編集(E) ビュー(V) クエリー(Q) お気に入り(F) ヘルプ(H)

pgAdmin III Query - hiroshi @ localhost:5432 \*

MemberShip - Visual Web Developer 2005 Express Edition

ファイル(F) 編集(E) 表示(V) Web サイト(S) ビルド(B) デバッグ(D) 書式(O) レイアウト(Y) ツール(T) ウィンドウ(W)

CREATE ( role appl );

CREATE ( user role appl );

CREATE user\_ CREATE

出力パン

データの出

クエリーは、

スタートページ Default2.aspx Default.aspx.cs Web.Config Default.aspx

ログイン

ユーザー名: \*

パスワード: \*

次回のために保存

ログイン

プロパティ

Login1 System.Web.UI.WebControls.Login

User Name Required Error Message ユーザー名が必要です。

動作

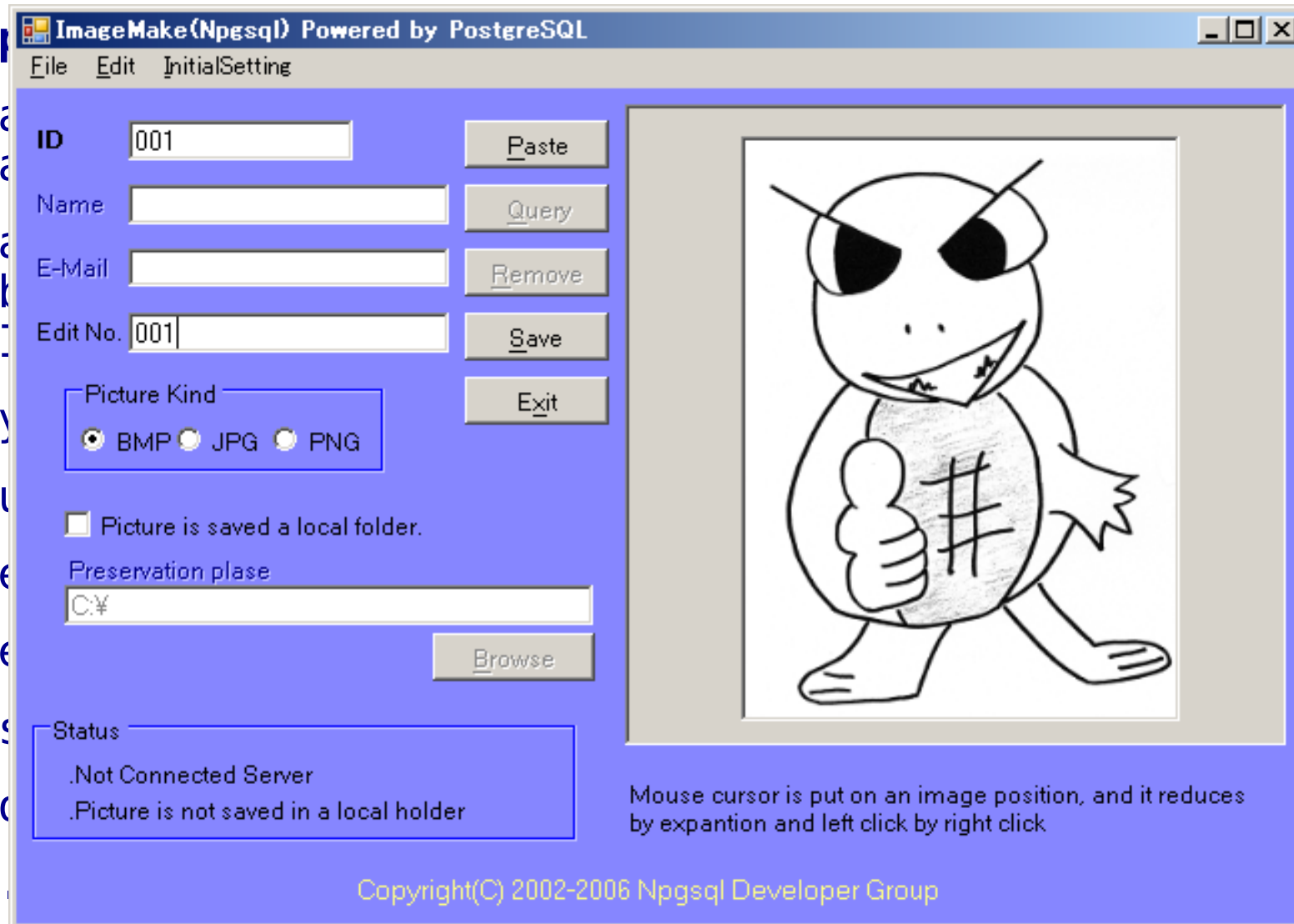
DestinationPageUrl	~/Default2.aspx
DisplayRememberMe	True
Enabled	True
EnableTheming	True
EnableViewState	True
FailureAction	Refresh
RememberMeSet	False
SkinID	
ToolTip	
Visible	True
VisibleWhenLoggedIn	True

その他

OK. 行 17 カラム 1 文字 426 187 ms

OK. 行 2 カラム 2 文字 22 500 ms

# ImageMake-LargeObject







# System.Transactions in Npgsql

---

.Net and Mono

# Using System.Transactions

---

```
using (TransactionScope scope =
    new TransactionScope())
{
    using (NpgsqlConnection connection =
        new NpgsqlConnection(connectionString))
    {
        // auto enlist in transaction
        connection.Open();
        // perform transacted operations
    }
    scope.Complete();
}
```

# Distributed Transactions

---

- Requires Windows and the Microsoft CLR.
- Requires MSDTC to support managing the distributed transaction.
- Does not yet support recovery from failure in the middle of the two phase commit.

# System.Transactions Demo

---

- Demo distributed transaction with two connections
- Demo recovery from failure during two phase commit

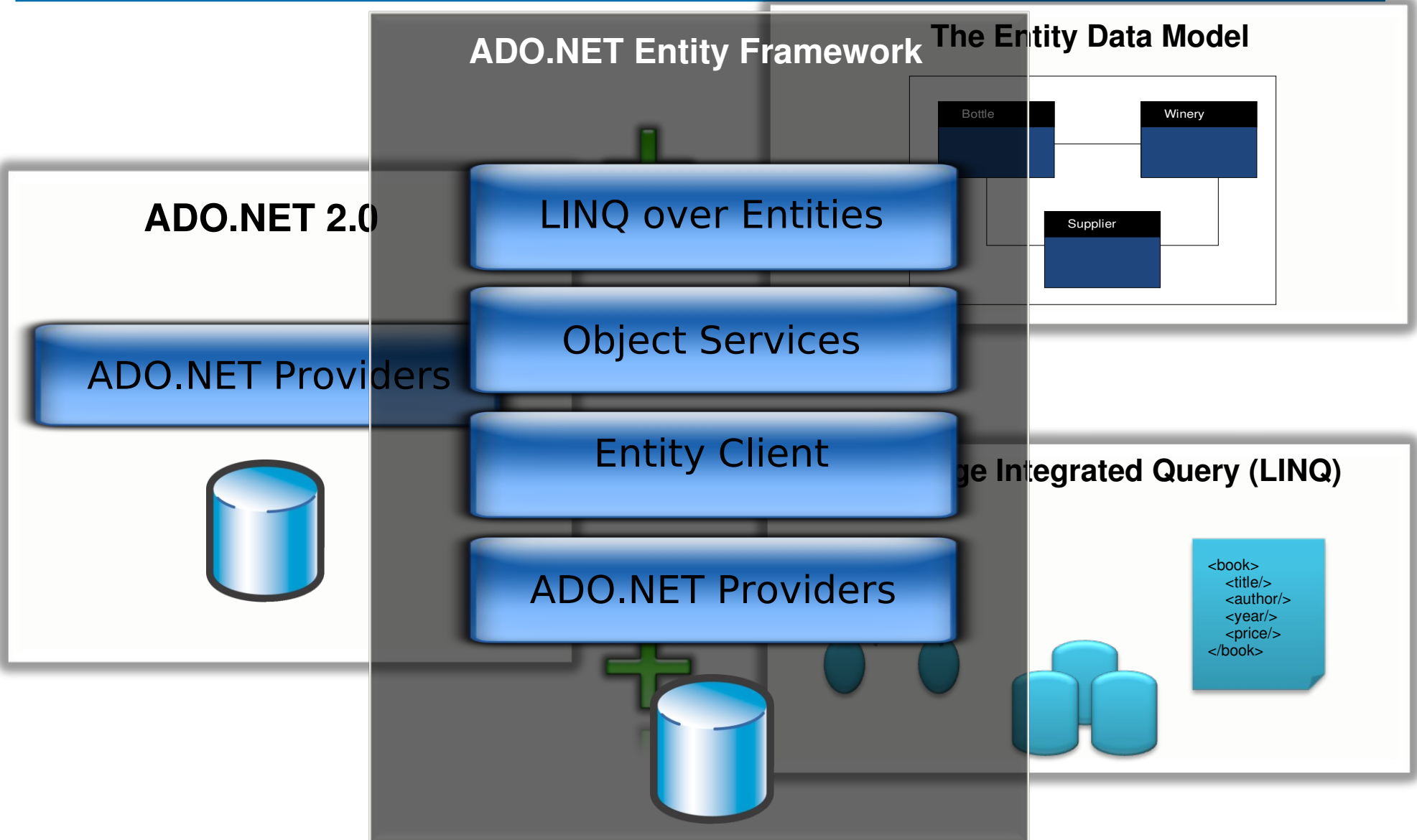
# Entity Framework

---

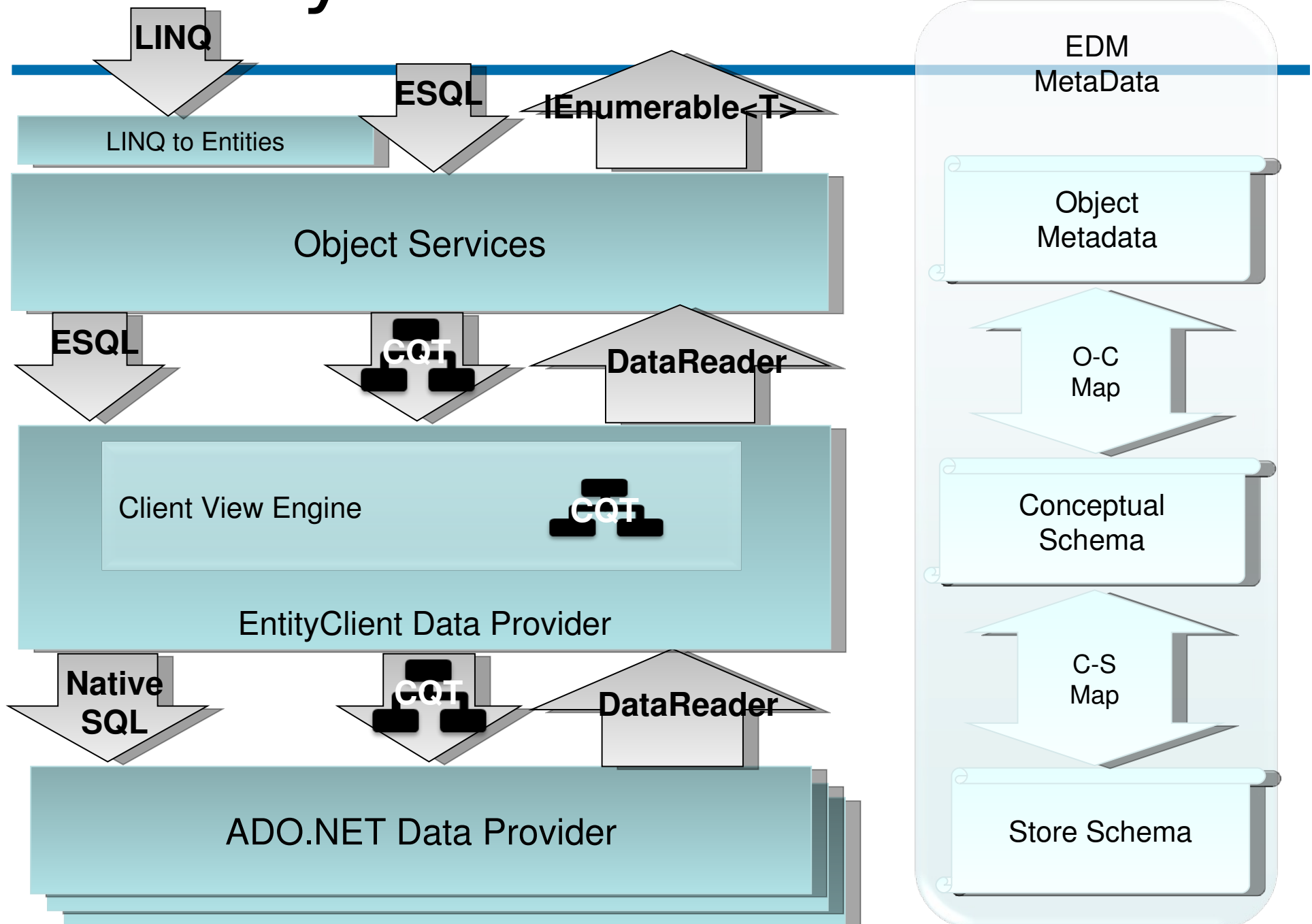
## ADO.NET for .NET 3.5 SP1

\*portions of presentation reused from DevDays 2007  
with permission from Microsoft

# The Entity Framework



# Entity Framework Overview





# EDMGEN.EXE

---

- Generates the model files
- Model.ssd (Store Schema Definition Language)
  - Describes tables and columns which map to Entities and Relationships
- Model.csd (Conceptual Schema Definition Language)
  - Describes the Entity Data Model (incl. EntitySets, EntityTypes, Associations & AssociationsSets)
- Model.msl (Mapping Specification Language)
  - Describes how the Entity Framework maps between the Conceptual Model (CSDL) and the logical Storage Schema (SSDL)
- Model.cs or Model.vb
  - Partial class implementing the model created

# Entity Framework Demo

---

- Generate Model
- ESQL with objects
- LINQ with objects
  - LINQ with relationship restrictions

# Future of Npgsql

---

- Npgsql 2.0 release (see current betas)
  - Includes support for .NET 2.0
  - Includes support for .NET 3.5 SP1
- Custom Type support
- Better Visual Studio integration for 2005 and 2008 (DDEX support).

# Community

---

- ▄ Npgsql has a very active community
- ▄ Project is hosted at [pgfoundry.org](http://pgfoundry.org). Previously it was hosted at [gborg.org](http://gborg.org)
- ▄ In March, we registered [npgsql.org](http://npgsql.org) domain
  - Thanks Magnus for registrar tip! :)

# Community

---

- ▄ pgfoundry.org provided us with a great user interface which were forums. They improved very much user interaction
- ▄ Users are very active at Npgsql forums
- ▄ We get great feedback there
- ▄ Many users provide valuable bug reports and fixes as well!!

# Community

---

## ▄ We have some friendly urls redirections:

- ▄ <http://project.npgsql.org>
- ▄ <http://documentation.npgsql.org>
- ▄ <http://forums.npgsql.org>
- ▄ <http://cvs.npgsql.org>

# Obrigado – Arigato – Thank you

---

## References:

- <http://www.microsoft.com/japan/msdn/>
- [http://www.mono-project.com/Main\\_Page](http://www.mono-project.com/Main_Page)
- <http://www.npgsql.org/>

Francisco Figueiredo Jr.  
francisco@npgsql.org

Hiroshi Saito  
hiroshi@npgsql.org

Josh Cooley  
jbcooley@npgsql.org