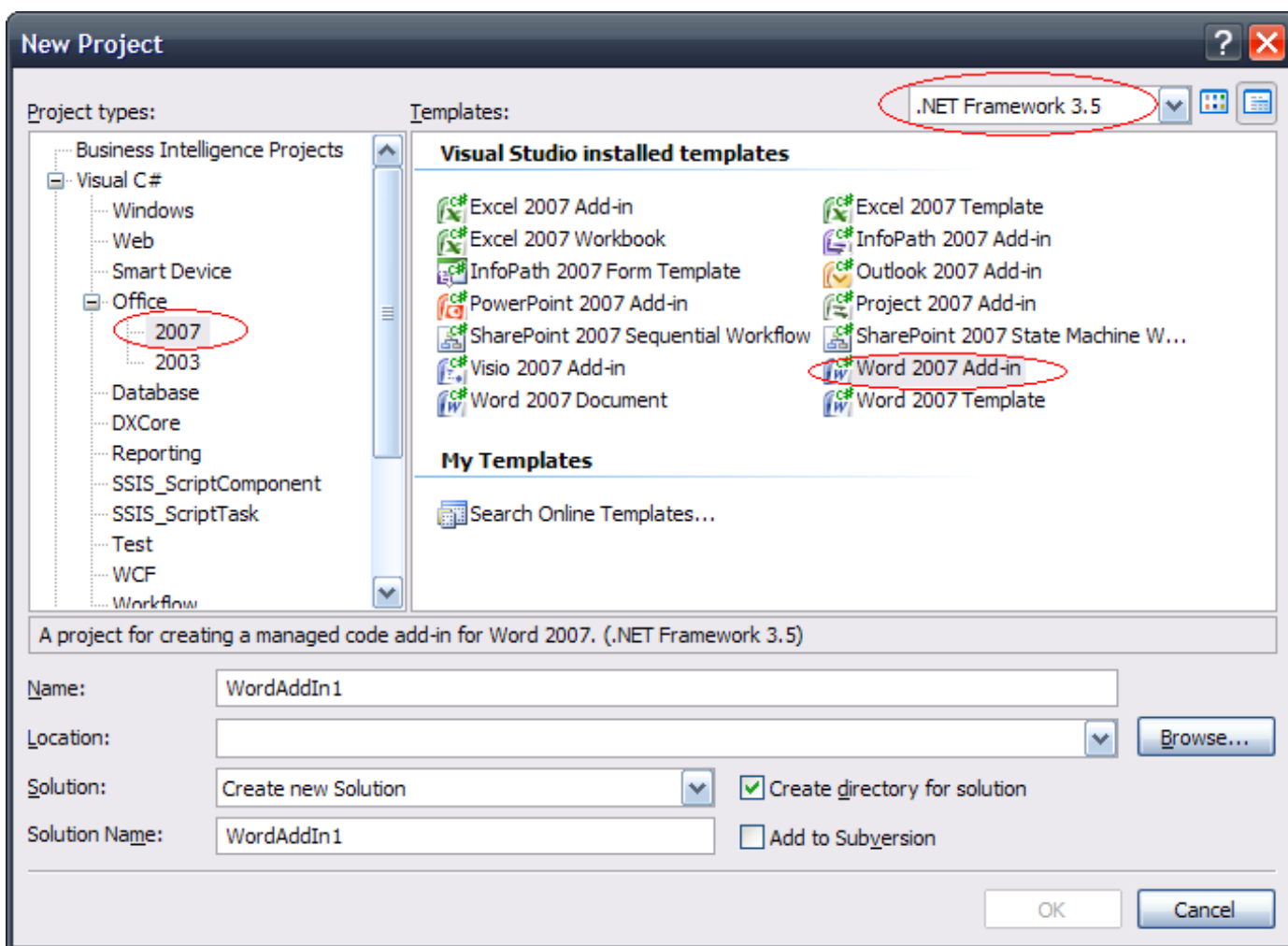


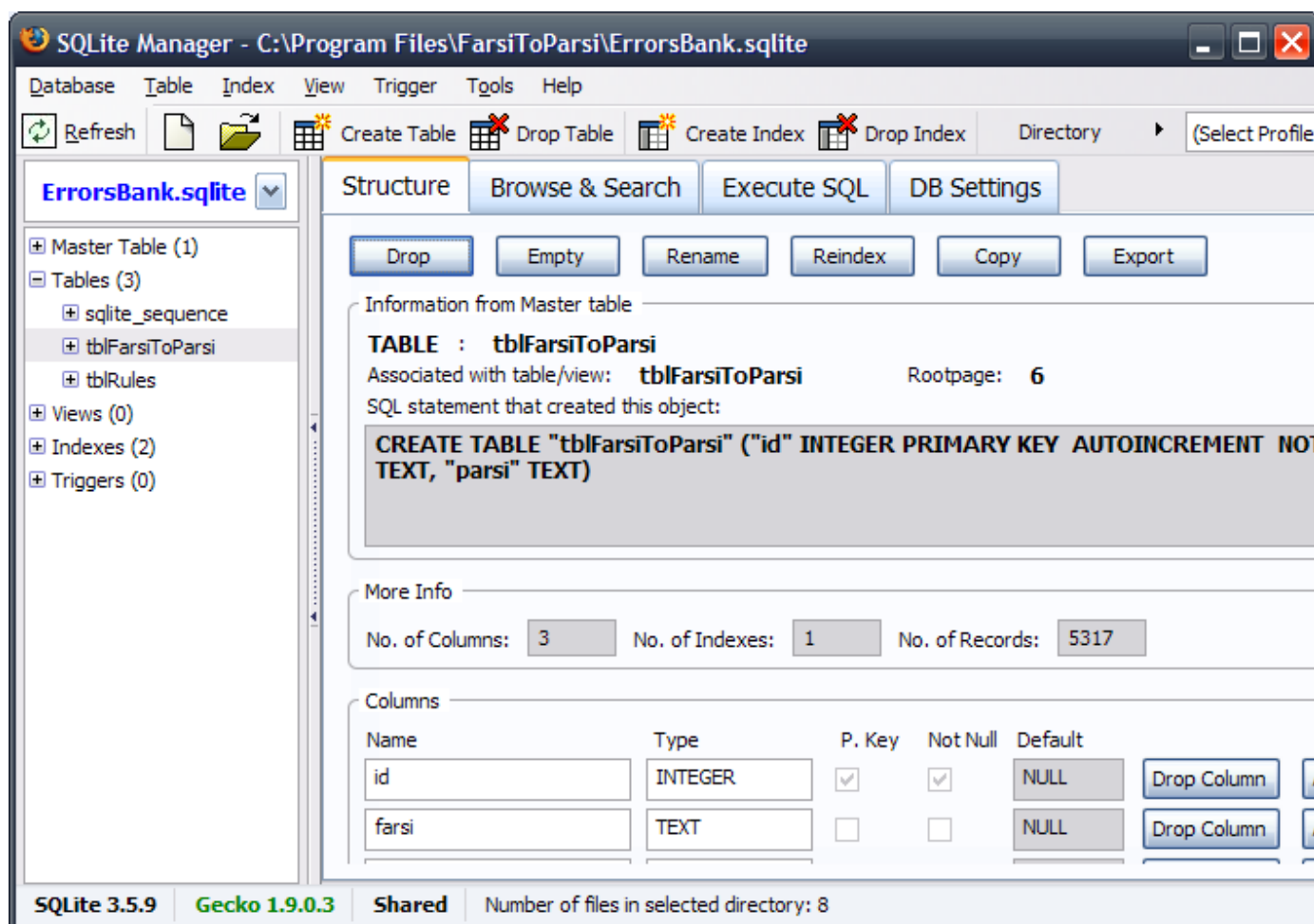
این افزونه با استفاده از ابزار Visual Studio Tools for Office که به VSTO مشهور شده است، تهیه شد. در بسته به روز رسانی سیستم که در ذیل (معرفی افزونه) نیز معرفی شد نگارش sp1 vsto3.0 آن به صورت خودکار نصب خواهد شد. برای ایجاد این پروژه در VS.Net 2008، تنها کافی است یک پروژه جدید Word add-in را آغاز نمائیم. (شکل زیر)



قبل از ادامه بحث، بهتر است در مورد بانک اطلاعاتی مورد استفاده نیز توضیح داده شود. در اینجا از SQLite استفاده شد. (بسیار سبک، کم حجم و سریع است و اساساً یک کاربر نهایی برای تنظیمات آن نیازی نیست اطلاعاتی داشته باشد). بسته به روز رسانی سیستم (در مطلب قبلی)، این مورد را نیز به صورت خودکار نصب خواهد کرد (در GAC باید نصب شود وگرنه افزونه قادر به یافتن آن نخواهد شد).

برای ایجاد این بانک اطلاعاتی، از افزونه [SQLite manager](#) برای فایرفاکس استفاده شد. (این افزونه رایگان شما را از هر ابزار جانبی برای مدیریت یک بانک اطلاعاتی SQLite بی‌نیاز می‌کند) برای مثال فایل ErrorsBank.sqlite برنامه افزونه فارسی به پارسی را توسط افزونه SQLite manager فایرفاکس باز کنید (این

فایل را در محل نصب افزونه می‌توانید پیدا کنید). در اینجا می‌توان جداول جدید را ایجاد کرد، کوئری‌های دلخواه را اجرا نمود و یا اطلاعات را مرور کرده، حذف یا ویرایش کرد (شکل زیر).



و خوشبختانه این بانک اطلاعاتی و [محصول کننده‌های آن](#) با اطلاعات یونیکد فارسی هیچ مشکلی ندارند و برای کارهایی با وسعت کم و تعداد رکورد پائین یکی از بهترین انتخاب‌ها به‌شمار می‌روند. نحوه استفاده از SQLite نیز در دات نت [بسیار ساده است](#). اگر با ADO.Net کار کرده باشید، پس از افزودن ارجاعی از اسمبلی System.Data.SQLite.DLL به پروژه و معرفی فضای نام آن به پروژه، تنها کافی است در کدهای قبلی خود برای مثال SqlConnection را به SQLiteConnection تغییر دهید و امثال آن. یعنی دانش ADO.Net شما در اینجا نیز کاملاً قابل استفاده خواهد بود و نیازی نیست مدتی را صرف آشنا شدن با کلاس‌ها و مفاهیم جدید نمائید (البته این تنها زمانی معنا خواهد داشت که به ویزاردها عادت نکرده باشید و کارهای خود را با کد نویسی انجام داده باشید). تنها یک نکته را باید به‌خاطر داشت و آن هم مربوط است به ساز و کار درونی SQLite. هنگام انجام عملیات insert یا update حتماً از transaction استفاده کنید تا سرعت کوئری‌های شما در SQLite به نحو شگفت‌انگیزی افزایش یابد. مثالی در این مورد را در فایل chm راهنمای SQLite.NET می‌توانید پیدا کنید.

مطلب دیگری که پیش از پرداختن به کد نویسی افزونه باید با آن آشنا شویم، مفهوم smart tags در مجموعه آفیس است که در این پروژه از آن استفاده گردید.

smart tags در مجموعه آفیس برچسب‌هایی هستند که به صورت خودکار توسط یکی از محصولات آفیس مثلاً ورد یا اکسل و امثال آن، پس از تشخیص یک کلمه خاص ایجاد می‌شوند و می‌توان اعمالی را به این برچسب ایجاد شده انتساب داد. برای مثال در

اینجا امکان جایگزین کردن کلمه فارسی با معادل پارسی در نظر گرفته شد.

[ویدیویی](#) در مورد نحوه ایجاد اسمارت تگ‌ها در VS.Net و یا [مثالی پیشرفته‌تر](#) در مورد تشخیص دمای فارنهایت در یک متن و ایجاد smart tag مخصوص به آن برای تبدیل به سلسیوس. (از regular expressions جهت یافتن یک الگو در متن استفاده شده است)

در این پروژه، حدود 3800 واژه فارسی به یک smart tag انتساب داده می‌شود (در روال استاندارد ThisAddIn_Startup). سپس در هنگام نمایش آن، معادل پارسی کلمه نیز به منوی باز شده افزوده گشته و در روال رخداد کلیک آن، تعویض کلمه تشخیص داده شده با واژه پیدا شده صورت خواهد گرفت.

در ادامه فرض بر این است که یک پروژه جدید word add-in را در VS.Net ایجاد کرده‌اید و همچنین ارجاعی را به فایل System.Data.SQLite.DLL افزوده‌اید.

```
using System;
using System.Diagnostics;
using Microsoft.Office.Tools.Word;
using Action = Microsoft.Office.Tools.Word.Action;

private SmartTag _st;
private void init()
{
    try
    {
        //Enable Smart Tags in Word
        if (!Application.Options.LabelSmartTags)
        {
            //ممکن است اسمارت تگ‌ها در ورد غیرفعال باشند. به این صورت می‌شود آنها را فعال کرد
            Application.Options.LabelSmartTags = true;
        }

        _st = new SmartTag(@"www.microsoft.com/Demo#FarsiSmartTag", @"فارسی به پارسی");

        //دریافت واژه‌های فارسی از دیتابیس و افزودن خودکار آنها به اسمارت تگ‌ها
        if (!DBhelper.AddSmartTagItems(_st, "select distinct farsi from tblFarsiToParsi"))
            return;

        Action stActions = new Action("تبدیل"; //تعریف یک اکشن جدید
        stActions.Click += stActions_Click; //انتساب روال‌های رخداد گردان
        stActions.BeforeCaptionShow += stActions_BeforeCaptionShow;
        _st.Actions = new[] { stActions };
        VstoSmartTags.Add(_st); //افزودن اسمارت تگ به مجموعه
    }
    catch (Exception ex)
    {
        EventLog.WriteEntry("FarsiToParsi", ex.ToString(), EventLogEntryType.Error, 7);
    }
}

private void ThisAddIn_Startup(object sender, EventArgs e)
{
    init();
}
```

دو روال رخداد گردان زیر نیز جهت تغییر عنوان پیش فرض به واژه یافته شده در لحظه نمایش منو و روال کلیک نیز ایجاد خواهد شد:

```
static void stActions_BeforeCaptionShow(object sender, ActionEventArgs e)
{
    try
    {
        Action clickedAction = sender as Action;
        if (clickedAction != null)
        {
            string parsi = DBhelper.FindParsi(e.Text); //معادل پارسی از دیتابیس دریافت می‌شود
            clickedAction.Caption = (parsi == string.Empty ? e.Text : parsi);
        }
    }
    catch (Exception ex)
```

```

        {
            EventLog.WriteEntry("FarsiToParsi", ex.ToString(), EventLogEntryType.Error, 7);
        }
    }

    static void stActions_Click(object sender, EventArgs e)
    {
        try
        {
            Action clickedAction = sender as Action;
            if (clickedAction != null)
            {
                e.Range.Text = clickedAction.Caption; // جایگزینی متن موجود با عنوانی که پیشتر پارسی
                شده است
            }
        }
        catch (Exception ex)
        {
            EventLog.WriteEntry("FarsiToParsi", ex.ToString(), EventLogEntryType.Error, 7);
        }
    }
}

```

نکته‌ای را که در اینجا باید حتما رعایت کرد بحث exception handling است. خصوصا در روال استاندارد ThisAddIn_Startup. اگر در این روال خطایی مدیریت نشده رخ دهد، word افزودنی شما را به صورت غیرفعال به مجموعه اضافه خواهد کرد و فعال سازی بعدی آن پس از اصلاح کد واقعا مشکل خواهد بود. همانطور که ملاحظه می‌کنید تمامی خطاها در event log ویندوز نوشته می‌شوند.

همچنین باید دقت داشت که اگر متغیری در سطح کلاس تعریف نشود به احتمال زیاد تا دقایقی بعد توسط garbage collector دیار باقی خواهد شتافت (تعریف st_ در اینجا). اینجاست که شاید ساعت‌ها وقت صرف کنید که چرا روال‌های رخداد گردان دیگر اجرا نمی‌شوند. چرا افزونه دیگر کار نمی‌کند.

همین! کل سورس این add-in منهای بحث دریافت اطلاعات از دیتابیس همین بود! وظیفه‌ی تشخیص کلمات معرفی شده به ms-word به‌عهده‌ی خود آن است و این کار را نیز به‌خوبی انجام می‌دهد. در گذشته‌های نچندان دور ایجاد یک افزونه برای word واقعا مشکل بود که با این روش بسیاری از موانع برطرف شده است.

کلاس DBHelper که کار دریافت اطلاعات واژه‌ها را از دیتابیس SQLite انجام می‌دهد به شرح زیر است:

```

using System;
using System.Data.SQLite;
using System.Diagnostics;
using System.Reflection;
using Microsoft.Office.Tools.Word;

namespace Farsi2Parsi
{
    class DBHelper
    {
        #region Methods (2)

        // Public Methods (2)

        public static bool AddSmartTagItems(SmartTag st, string strSQL)
        {
            SQLiteDataReader myReader = null;
            SQLiteCommand sqlCmd = null;
            bool ret = false;
            try
            {
                SQLiteConnection sqlCon = new SQLiteConnection
                {
                    ConnectionString = "Data Source=" + ConStr.ConnectionString
                };
                sqlCon.Open();
                sqlCmd = new SQLiteCommand(strSQL, sqlCon);
                myReader = sqlCmd.ExecuteReader();

                if (myReader != null)
                    while (myReader.Read())

```

```

        {
            if (myReader.GetValue(0) != DBNull.Value)
                st.Terms.Add(myReader.GetValue(0).ToString());
        }

        ret = true;
    }
    catch (Exception ex)
    {
        EventLog.WriteEntry("FarsiToParsi", ex + "\n" + Environment.CurrentDirectory + "\n" +
            Assembly.GetExecutingAssembly().Location, EventLogEntryType.Error, 7);
    }
    finally
    {
        if (myReader != null)
            myReader.Close();

        if (sqlCmd != null)
            sqlCmd.Connection.Close();
    }
    return ret;
}

public static string FindParsi(string farsi)
{
    SQLiteDataReader myReader = null;
    SQLiteCommand sqlCmd = null;
    string ret = string.Empty;
    string strSQL = "select parsi from tblFarsiToParsi where farsi='" + farsi.Replace("'", "''")
+ "'";
    try
    {
        SQLiteConnection sqlCon = new SQLiteConnection
        {
            ConnectionString = "Data Source=" + ConStr.ConnectionString
        };
        sqlCon.Open();
        sqlCmd = new SQLiteCommand(strSQL, sqlCon);
        myReader = sqlCmd.ExecuteReader();

        if (myReader != null)
        {
            myReader.Read(); //اولین مورد کافی است
            if (myReader.GetValue(0) != DBNull.Value)
                ret = myReader.GetValue(0).ToString();
        }
    }
    catch (Exception ex)
    {
        EventLog.WriteEntry("FarsiToParsi", ex + "\n" + Environment.CurrentDirectory + "\n" +
            Assembly.GetExecutingAssembly().Location, EventLogEntryType.Error, 8);
    }
    finally
    {
        if (myReader != null)
            myReader.Close();

        if (sqlCmd != null)
            sqlCmd.Connection.Close();
    }
    return ret;
}
}
#endregion Methods
}

```

همانطور که پیشتر نیز عنوان شد اگر با ADO.net آشنایی داشته باشید، هیچ نکته‌ی خاص جدیدی را در اینجا مشاهده نخواهید کرد و تنها یک سری امور روزمره کاری با ADO.net مطرح شده است، باز کردن کانکشن، اجرای کوئری، دریافت اطلاعات و پاکسازی نهایی. (قسمت finally را با استفاده از عبارت using می‌شود حذف کرد)

هنگام نصب برنامه، مسیر پوشه نصب در رجیستری ویندوز توسط نصاب نوشته خواهد شد. از همین مورد برای ایجاد رشته اتصالی به دیتابیس استفاده گردید.

```
class ConStr
{
    public static string ConnectionString
    {
        get
        {
            return
Microsoft.Win32.Registry.LocalMachine.OpenSubKey("SOFTWARE\\FarsiToParsi").GetValue("folder") +
"\\ErrorsBank.sqlite";
        }
    }
}
```

سورس کامل این افزونه را به صورت یک پروژه VS.Net 2008 SP1 از [اینجا](#) می‌توانید دریافت کنید.
نصاب برنامه با استفاده از [NSIS](#) ایجاد شده که در روزی دیگر درباره‌ی آن توضیح خواهم داد.
اگر قصد داشته باشید از روش‌های متداول استفاده کنید، مشاهده ویدیوی زیر توصیه می‌شود:

<http://msdn.microsoft.com/en-us/office/bb851702.aspx>

برای توزیع این نوع افزونه‌ها علاوه بر دات نت فریم ورک، به چهار به روز رسانی دیگر نیز نیاز خواهد بود:
به روز رسانی نصاب ویندوز (که احتمالاً نصب هست)

WindowsInstaller-KB893803-v2-x86.exe

: Microsoft Office System Update: Redistributable Primary Interop Assemblies

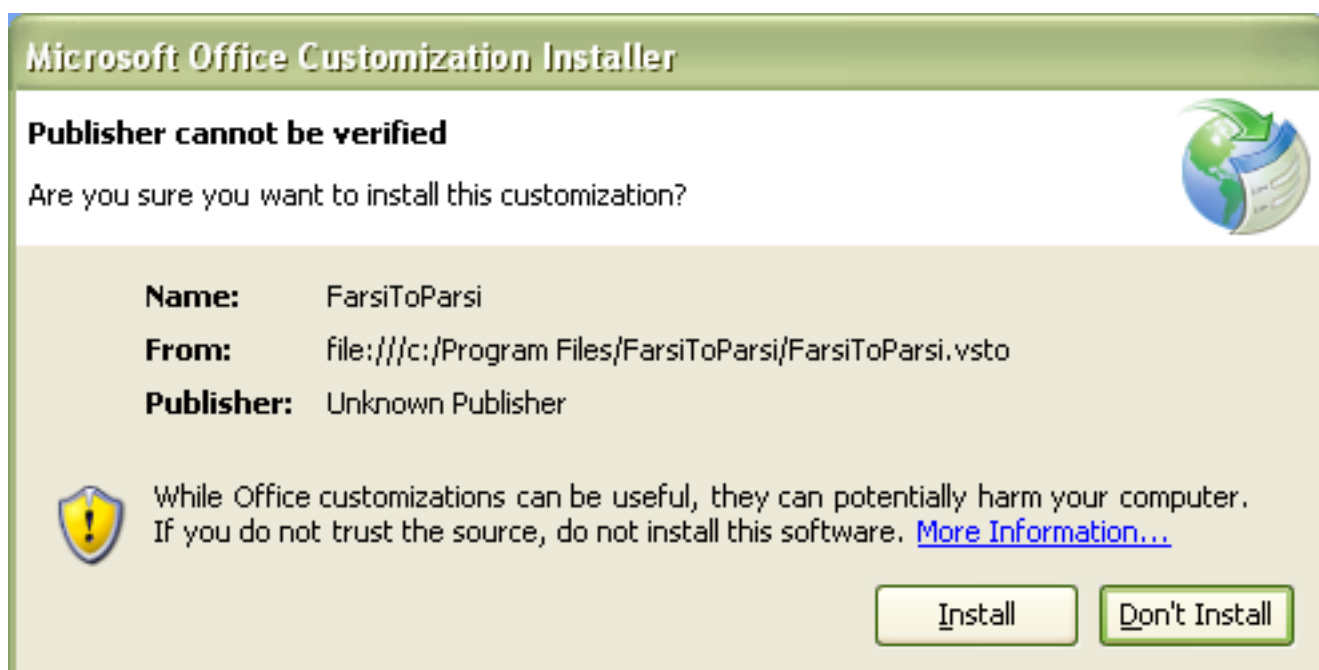
o2007pia.msi

نصب و همچنین sp1 آن

vstor30.exe

vstor30sp1-KB949258-x86.exe

این موارد را من در بسته [به روز رسانی سیستم](#) قرار داده‌ام که به صورت خودکار و یکی پس از دیگری اجرا و نصب خواهند شد.
پس از آن با کلیک بر روی فایل با پسوند vsto که در پوشه build برنامه موجود است، می‌توان افزونه را نصب کرد (click once installation).



سایر اطلاعات در مورد پروژه‌های VSTO را می‌توان از طریق وبلاگ رسمی آنها دنبال کرد:

[/http://blogs.msdn.com/vsto](http://blogs.msdn.com/vsto)

ایده‌های دیگری را هم در همین رابطه می‌توان پیاده سازی کرد. برای مثال درست کردن یک افزونه برای بررسی آئین نگارش فارسی در متون word. دقیقاً با همین روش قابل پیاده سازی است و یا ایجاد غلط یاب بهتری نسبت به آنچه که هم اکنون برای آفیس 2003 توسط مایکروسافت ارائه شده است (این غلط یاب با صفحه کلید استاندارد تایپ ایران همخوانی ندارد، به همین جهت با استقبال نیز مواجه نشد).

نظرات خوانندگان

نویسنده: Afshar Mohebbi
تاریخ: ۱۳۸۷/۰۸/۱۷ ۱۲:۱۵:۰۰

کار خیلی جالبی انجام داده‌اید. موفق باشید.

نویسنده: Anonymous
تاریخ: ۱۳۸۷/۰۸/۱۷ ۲۲:۲۵:۰۰

واقعا دستت درد نکنه
چند تا سوال داشتم
میشه دیتا بیس رو xml گذاشت؟ (با یونیکد مشکل پیدا نمیکنه!)
اگه امکان داره چند کتاب الکترونیکی خوب رو که در این زمینه ها مطالعه کردی معرفی کنی
مرسی

نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۰۸/۱۸ ۰۰:۴۰:۰۰

- فایل xml با یونیکد مشکلی ندارد و اینجا هم میشد از xml استفاده کرد.
برای نمونه یکی از برنامه نویسان گرامی در سایت
<http://psoft.ir/?p=23>
از xml بعنوان دیتابیس نمایش اشعار، پند و اندرز و امثال آن استفاده کرده.
- حقیقت امر این است که در این سایت من از بحث وارز و امثال آن پرهیز خواهم داشت. ولی محض نمونه این عبارات را به
همراه pdf در انتهای آن در گوگل جستجو کنید کتابهای مربوطه رو می‌تونید به سادگی پیدا کنید:
Wrox.Professional.VSTO.2005.Visual.Studio.2005.Tools.for.Office
Visual.Studio.Tools.for.Office.Using.C.Sharp.with.Excel.Word.Outlook.and.InfoPath

موفق باشید

نویسنده: امین
تاریخ: ۱۳۸۷/۰۸/۱۸ ۰۷:۰۴:۰۰

سلام آقای نصیری
خسته نباشید. امیدوارم که همیشه موفق باشید و سلامت تا ما هم همیشه از وبلاگتون استفاده کنیم.

نویسنده: مهدی پایروند
تاریخ: ۱۳۸۸/۰۳/۰۵ ۱۸:۰۵:۱۶

سلام، لطفا در مورد نحوه کار با تاریخ در دیتا بیس sqlite کمی توضیح و راهنمایی بفرمایید.
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۳/۰۵ ۱۸:۱۷:۱۱

سلام،
تاریخ که احتمالا تاریخ فارسی مد نظر شما است همیشه بین علما در این مورد اختلاف بوده و هست ! (:
یکی تاریخ رو به صورت عددی با فرمت یونیکس ذخیره میکنه (unixepoch). بعد تمام مقایسه‌های لازم هم درست صورت خواهد

گرفت (مثلا از تاریخ تا تاریخ) چون با عدد کار می‌کنید.

یکی مثل قبض برق و آب و تلفن از رشته استفاده می‌کنه ولی با این شرط که حتما روز و تاریخ دو رقمی ثبت شوند و اگر یک رقمی بود قبل از آن یک صفر قرار گیرد تا مقایسه از تاریخ تا تاریخ همیشه کار کند (از زمان فاکس پروی مرحوم تا به امروز)

یکی هم زمان رو از نوع datetime استاندارد در نظر می‌گیرد که در sqlite تقریبا اثری ندارد. اما sqlite یک تابع به نام date معرفی کرده که به خوبی بر روی رشته‌های تاریخی کار می‌کند و بسیار توانمند است:

<http://www.sqlite.org/cvstrac/wiki?p=DateAndTimeFunctions>

نویسنده: مهدی پایروند

تاریخ: ۱۳۸۸/۰۳/۰۵ ۱۸:۵۷:۳۱

ltdn f,n , llkمفید بود و ممنون از راهنماییتون استاد (: