

بسم الله السميع العليم

استغلال ال RSS في برامجنا

بما انك تقرأ كلماتي هذه فأنت بالطبع مبرمج (مطور برامج او مواقع ويب) ، و لذا فلا تساورني اي شكوك في أن كلمة مثل RSS قد مرت عليك اثناء تصفك للإنترنت و لشحن ذاكرتك قليلا انظر الى هذه الايقونة () أظنك تتذكر الان، انظر الى هذه ايضا () ...

Really Simple Syndication

إذا تصفحت النت و المنتديات فستجد الكثيرين قد تحدثوا عنها بإسهاب و لا اخفيك انني لا اريد ان اكون واحدا منهم اي انني لا انوي الإسهاب لانهم قد غطوا الموضوع بجوانبه و الحمد لله

ما اريده هنا هو إعطائك فكرة سريعة عنها؟، أمثله؟، متى استغلها كمستخدم؟ متى استغلها كمزود؟

مقدمة

دعنا نتوكل على الله و نبدأ ... ال RSS عبارة بكل اختصار عن نصوص XML و ال XML ما هي إلا لغة لوصف البيانات فلو اردتني ان اصف نفسي بال XML فسأقول:

```
<Person>
  <Name>Ragheed Al-Tayeb</Name>
  <Age>27 Years</Age>
  <Job>IT Engineer</Job>
</Person>
```

لو تمعنت فيما سبق ستجد انني وضعت معلوماتي الشخصية في عنصر باسم Person ثم وضعت فيه عناصر فرعية باسم Name, Age, Job وهي تصف اسمي و عمري و عملي و اذا ارسلت لك ملف يحتوي على هذه البيانات فأنت بسهولة سوف تقرأها على الرغم من عدم وجود اتفاق مسبق بيننا و بنظرة خاطفة عليها ستفهم ما اريد ان اصفه لك لأنها ببساطة لغة ال XML و التي تعتمد على البساطة في الوصف و الذي يعتمد بدوره على مدى بساطة الكائن المراد وصفه و الذي هو في مثالنا السابق (رغيد) ولأنني كائن بسيط (غير معقد) و الحمد لله فوصفي جاء بسيط.

لاحظ:

ان هذه العناصر يمكن ان تختلف ! اي ان تزيد او تنقص اعتمادا على الاشياء التي يريد المصدر وصفها فمثلا لو سألت زيدا ان يصف نفسه (ليس بالضروري) ان تجده قد استخدم نفس العناصر التي استخدمتها انا (Person, Name, Age & Job) ولربما استبدل <Man> بدلا عن <Person> و لربما اضاف عنصر مثل Mobile او ربما e-Mail و استغنى عن وصف عمله Job وبالتالي يكون الملف الذي كونه قريب من التالي :

```
<Man>
  <Name>Zaid Sameh</Name>
  <Age>20 Years</Age>
  <Mobile>123456789</Mobile>
  <e-mail>zaid@somewhere.com</e-mail>
</Man>
```

إضافة بسيطة و هي انه يمكن ان تحتوي هذه العناصر على خصائص معينة فمثلا اذا كان هناك خصائص خاصة بعنصر معين فقد قمت في مثالي الاول بذكر عنصر باسم Job (العمل) وهو عنصر يصفني فماذا اذا اردت ان اذكر شيئا يخص هذا ال Job اي يصف هذا العمل و لا يصف رغيد بحد ذاته (مثل نوع العمل: خاص او حكومي) ... فهنا تظهر الحاجة الى الخصائص Attributes وهي عبارة صفات خاصة بالعنصر ذاته ... خذ المثال التالي :

```
<Person>

  <Name>Ragheed Al-Tayeb</Name>

  <Age>27 Years</Age>

  <Job JobType="Private">IT Engineer</Job>

</Person>
```

وفيه ترى بوضوح انني اضفت الخاصية JobType و اسندت القيمة Private التي تدل على كون العمل هو عمل خاص.

و الان اذا كان رأسك قد ألمك من قبل من العبارات التي يكتبها امثالي عن ال XML (لغة ال XML لغة وحدث تبادل البيانات و سهلت تناقله بين المصادر المختلفة) يعني قاعدة بيانات من نوع معين تنقل بياناتها الى قاعدة اخرى مختلفة عنها عن طريق ال XML لغة برمجة تفاهم مع اخرى عن طريقها ايضا و الكثير الكثير و ... و ... و !!!

فمن أين جاء هذا التوحيد الذي يتكلمون عنه طالما أن رغيد وصف نفسه بطريقة و زيد وصف نفسه بأخرى؟! ..

هذا في الحقيقة يأتي من وجود اتفاق بينهم جميعا على عناصر وخصائص موحدة و بالتالي فعلى كل مصدر ان يوصف نفسه باستخدام نفس العناصر المتفق عليها فمثلا جدول في قاعدة بيانات SQL Server او Oracle او Access او MySQL كلها تخزن و تتعامل مع الجداول و العلاقات بينها بطرق مختلفة و لكن اذا ارادت هذه اللغات ان تصف نفسها بال XML فما الجدول إلا عبارة عن حقول و سجلات و بالتالي اذا تم الاتفاق على مجموعة من العناصر و خصائصها بينهم فمن السهل ان تصف قاعدة بيانات كاملة باستخدام ال XML و بالمثال يتضح المقال كما يقول اخي العزيز الغملاسي دائما، لهذا سأذكر مثلا بسيطا لقاعدة بيانات موصوفة بال XML ... كما يلي :

```
<Database Name="SchoolData">

  <Table Name="Students" FieldsCount=3>

    <Row>

      <Name Type="Text" Length=20>Ragheed</Name>

      <Age Type="Number" Length=2>27</Age>

      <Male Type="Boolean">True</Male>

    </Row>

    <Row>

      <Name Type="Text" Length=20>Zaid</Name>

      <Age Type="Number" Length=2>20</Age>

      <Male Type="Boolean">True</Male>

    </Row>

  </Table>

</Database>
```

يألفها من سهولة فيبساطة تستطيع ان تعرف ان قاعدة البيانات السابقة هي باسم SchoolData و انها تحتوي على جدول واحد باسم Students وهذا الجدول مكون من 3=FieldsCount ثلاثة حقول هي Name و Age و Male و بالنظر الى هذه العناصر تجد ان الحقل Name نوعه Text اي نص بطول 20 بينما الحقل Age هو من النوع Number و اما الحقل Male فهو عبارة Boolean اي True او False.

و ان الجدول Students يحتوي على حقلين Row (ويمكن بالطبع ان يحتوي على اكثر) بل يمكن لقاعدة البيانات تلك ان تحتوي على اكثر من جدول.

طبعاً لا اريدك ان تفهم ان هذه بالضبط هي الطريقة التي يتم وصف قاعدة البيانات فيها بال XML وانما هذا مثال للتوضيح.

من وجهة نظر المستخدم

اذا تسألت ما العلاقة بين عنوان الموضوع و المقدمة الطويلة التي بدأت بها ؟! .. فاني ببساطة اقول ... هذه هي ال RSS غير انها عبارة عن XML تم الاتفاق عليها مسبقاً ومنها اصدارات كثيرة لن اخوض فيها لانه ليس هدفي و سأكتفي بالاصدار 2.0 .

لم تعد الانترنت كما كانت مسبقاً فملايين الصفحات منتشرة حول العالم و المئات منها يلفت انتباهك و يجذبك فمثلاً يستهويني آخر مقالات مجلة ال MSDN من موقع مايكروسوفت كما يروقني الاطلاع على حالة الطقس في اماكن تهمني و غيرها الكثير ... منا من يهتم بأخبار الرياضة او السياسة او الاخبار الرياضية و بالتالي اذا جمعت كل اهتماماتك ستجد انه ينبغي عليك في كل يوم ان تزور خمسة , عشرة او حتى عشرون موقع يومياً بل و ربما بشكل متكرر في اليوم!

هنا تكمن الحاجة الى ال RSS فما هي إلا عبارة عن مقتطفات سريعة عن المواضيع المختلفة مع روابط الى المواضيع كاملة على الانترنت.

فمثلاً اذا كنت تستخدم الويندوز Vista كما في حالتي فأنت ستلاحظ قالب صغير في الشريط الجانبي للفيستا يعرض لك آخر الاخبار من مواقع تقدم خدمة ال RSS انظر الى الصورة التالية :



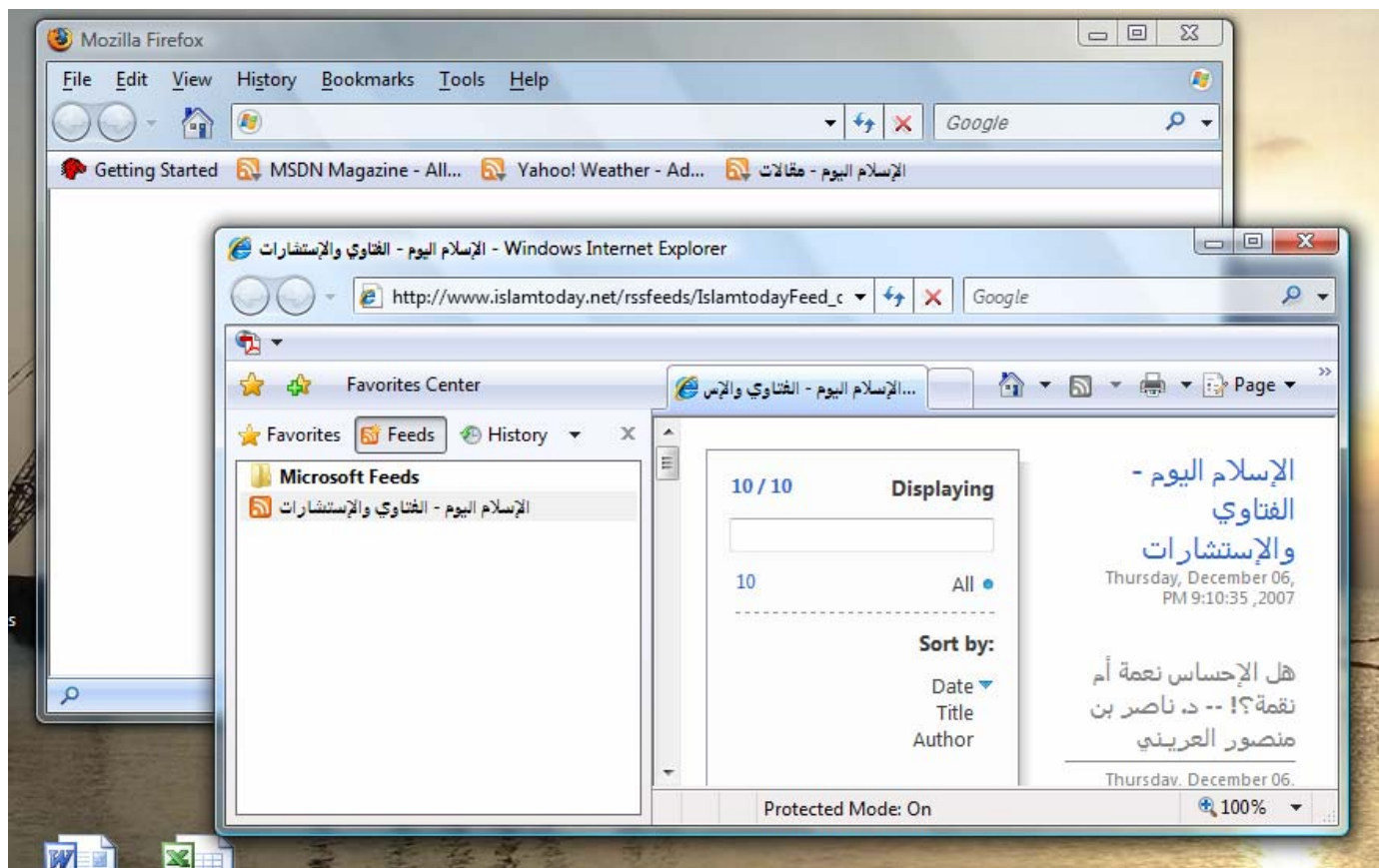
ففي المربع الاسود في الشاشة تراني استعرض الاخبار من MSNBC و بالتالي اذا لفت انتباهي خبر معين فيكفي ان انقر عليه مثلا لفت انتباهي خبر قتل 8 أشخاص في مجمع تجاري على يد شاب، و بالتالي عند النقر عليه يظهر :



سيقول قائل لماذا يقدم هذا الموقع بتقديم خدمة كهذه ؟ ... أليس من الأفضل ان يقدمها على الموقع و بالتالي يمكن للزوار ان يتصفحوه عوضا عن هذه المقطوعات ! ... في الحقيقة قولا كهذا سيجعل القائمون على الموقع يخسرون الكثير من زوارهم فليس الجميع يمتلكون دائما الوقت الكافي ومن يمتلك الوقت الان لتصفح موقعك لن يمتلكه غدا .. تخيل معي ان معظم المواقع تقدم هذه الخدمة ! ... هل تعرف ماذا سيحصل سيكون بإمكانك ان تعرف عناوين المشاركات الجديدة في المنتديات التي تحبها وبالتالي فتح المواضيع التي تحب المشاركة فيها، ستعرف آخر الاخبار من مواقع الاخبار المفضلة (كالجزيرة مثلا) ستري نتائج المباريات الرياضية ستعرف اسعار العملات ستتابع عدد الزائرون اليوم في موقعك الذي تديره و الاضافات التي اضافوها او ربما ماذا كتب اليوم من جديد في سجل الزوار بموقعك، ستعرف آخر المقالات التقنية للـ MSDN عن لغة البرمجة التي تفضلها، ستأخذ فكرة عن حالة اليوم الطقس في المنطقة التي تسكن فيها عائلتك (اذا كنت مثلي تعمل ببلد بعيد عنهم) و سيسرك مثلا ان تجري اتصلا هاتفيا تسألهم فيه الامطار التي تهطل عليهم الان !! .. لترى الدهشة عليهم وهم يسئلونك من اخبرك عن الامطار التي عندنا؟! ... (لايعرفون ان برنامجا بسيطا نقل الخبر الى سطح مكتبي).

حقيقة لايوجد حصر لما يمكن ان تجنيه كمستخدم من كون الدعم يزداد لخدمة الـ RSS فما ان تزور موقع يروق لك ثم ترى علامة الخدمة (RSS) او (📡) فقل حينها الحمد لله

لكي تشترك في خدمة ماعليك سوى زيارة الرابط الذي سيقترحه عليك الموقع وسيقوم المتصفح لديك بتسجيلها لديك للإطلاع عليها لاحقا .. هذه الصورة توضح شكلها في الـ Microsoft Internet Explorer IE7 و في الـ Mozilla



في الواقع توجد العديد من البرامج الخاصة بقراءة الـ RSS من الموقع التي تقدمها ... مثل :

<http://www.rssreader.com/>

<http://www.feedreader.com/>

<http://www.deskshare.com/awr.aspx>

او يمكنك الاعتماد على متصفح الانترنت الذي تستخدمه كما سبق اعلاه.

من وجه نظر المطور

الى هنا ينتهي حديثي معك كمستخدم ليبدأ حديثي معك كمبرمج و مطور مواقع، فلربما كانت المقدمة قد اسالت لعابك عن ال XML و اعقبنا ان خدمة ال RSS هي عبارة عن وصف باستخدام ال XML و متفق عليها فيا ترى كيف يبدو ملف بسيط من ملفات ال RSS الاصدار 2.0 :

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
  <channel>
    <title>عنوان القناة</title>
    <description>وصفها ويمكن ان تكون قناة رغيد الاخبارية</description>
    <link>http://msdn2.microsoft.com </link>
    <lastBuildDate>Mon, 24 Sep 2007 07:33:37 -0700</lastBuildDate>
    <pubDate>Wed, 21 Nov 2007 14:12:12 -0700 </pubDate>
    <item>
      <title>الخبر الاول</title>
      <description>مجرد خبر ساذج لغرض التجربة</description>
      <link>http://www.a.b.c.com</link>
      <pubDate>Wed, 21 Nov 2007 14:12:12 -0700</pubDate>
    </item>
    <item>
      <title>هذا موضوع ثاني</title>
      <description>موضوع جميل ايضا لافائدة منه</description>
      <link>http://www.anywhere.com</link>
      <pubDate>Wed, 31 Oct 2007 17:34:15 -0700</pubDate>
    </item>
  </channel>
</rss>
```

في الحقيقة قبل ان اختار وضع مثال فكرت مليا هل من المفيد ان اخوض في جميع قيم و احتمالات ملف ال RSS (اي جميع العناصر و خصائصها) ام اكتفي بلمحة بسيطة صغيرة !!؟

ارتأيت أخيرا ان يكون المثال و ما يتبعه مقتصرًا على القيم الشائعة و التي ستجدها اينما ذهبت و يمكنك منها ان تنطلق الى ابعد من ذلك بكثير.

من مثالنا يمكن القول بأن ملف الـ RSS يجب ان يحتوي على او عنصر وهو **<rss>** ويجب ان يحتوي هذا العنصر على الخاصية version التي تحدد الاصدار بالشكل **<rss version="2.0">** هذا لإننا نستخدم الاصدار الثاني (وهو ما سنستخدمه في أمثلتنا).

ويجب ان يحتوي عنصر الـ **<rss>** على العنصر الاساسي **<channel>** وهو العنصر الذي سيحتوي على جميع بيانات الخدمة .. بالشكل التالي :

```
<channel>

    <title> </title>

    <description> </description>

    <link> </link>

    <lastBuildDate> </lastBuildDate>

    <pubDate> </pubDate>

    <item>

        . . . . .

    </item>

</channel>
```

ومن هنا نرى ان عناصر فرعية يمكن اضافتها الى العنصر **<channel>** حتى نصفه وهي Title وفيه عنوان القناة و Description وهو وصفها و Link وهو رابط الى موقع القناة الاخبارية و LastBuildDate وهو تاريخ التحديث و PubDate وهو تاريخ إنشاء القناة فإذا كنت تنشئ القناة بشكل يومي فينبغي عليك تغييره عند بداية كل يوم وإذا حصل وقمت بتحديث شيء ما خلال اليوم غير العنصر LastBuildDate وهو العنصر الخاص بالتحديث.

وأخيرا العنصر الأكثر أهمية **<item>** وهو العنصر الذي يحتوي على الخبر بالشكل التالي :

```
<item>

    <title>العنوان</title>

    <description>الوصف</description>

    <link>الرابط</link>

    <pubDate>التاريخ</pubDate>

</item>
```

ستجد أنه يحتوي على عناصر (العنوان و الوصف و الرابط و تاريخ الإنشاء) و يمكن ان يحتوي العنصر <channel> على أكثر من عنصر من النوع <item> و ذلك بحسب الاخبار التي تريدها في القناة.

إضافة جميلة الى الخدمة RSS و هي إمكانية احتوائها على روابط صور ... اذا عدت الى الصورة الثانية في الموضوع ستجد انه عندما لفت انباهي موضوع قتل شاب لثمانية اشخاص في مجمع تجاري نقرت عليه فظهرت صورة ذلك الشاب وهذا نتيجة لإستخدام العنصر الخاص بالصورة وهو كالتالي:

```
<image>

    <url>رابط الصورة</url>

    <title>العنوان</title>

    <link>رابط الموقع</link>

    <width>عرض الصور</width>

    <height>ارتفاعها</height>

    <description>وصفها</description>

</image>
```

ملاحظة: لاحظ ان العناصر باللون الاحمر هي عناصر اختيارية يمكن الاستغناء عنها.

وبهذا يمكن أن يضاف العنصر <image> إلى أي من العنصرين <channel> أو <item> لكي تضاف صورة إليهما.

للعلم فقط:

يقوم الكثير من مزودي خدمة الـ RSS بتجاهل العنصر image و القيام بوضع الصور داخل الوصف Description الخاص بكل Item بحيث يكون الوصف عبارة عن HTML فإذا قمت بعرضه يظهر على شكل نصوص و صور وهو أمر شائع نسبياً.

و الآن لننتقل سوياً أنك صاحب موقع إنترنت و تريد تزويد زبائنك (سأسر كثيراً بكوني أول زبون) بخدمة RSS بحيث تعرض لهم مثلاً آخر الأخبار في موقعك أو إمكانية الإشتراك في موضوع معين بحيث يبقون مطلعين على آخر تحديثات الموضوع او ربما تريد عمل خدمة لتزويدك بمعلومات عن موقع مثل عدد المتصفحين الآن! .. إجمالي الزوار ... آخر التعليقات الجديدة حتى تبقى على اطلاع دائم بما يجري هناك على صفحات موقع دون الاضطرار إلى فتحه في كل مرة فقد تصلك الأمور التي تجري فيه بينما تقوم بتصفح جريدتك الإخبارية اليومية!

وبغض النظر عن نواياك فما تحتاج إليه هو إتصال بقاعدة بيانات تحتوي على معلومات ولنقل جدول فيه مجموعة من البيانات و الحقول مثل (العنوان و النص و التاريخ و الرابط) ... و بالتالي ستحتاج إلى Repeater وبضعة سطور برمجية مثلاً قاعدة بيانات بإسم data.mdb و جدول بإسم test فنقوم بتكوين إتصال OleDbConnection ثم امر SQL يقوم بإسترجاع البيانات إلى داخل OleDbDataReader نجعله أخيراً كـ مصدر لبيانات الـ Repeater.

حقيقة إذا لم تكن مبرمج ASP.NET فلن تستطيع معرفة ما أتحدث عنه من Repeater و ما شابه و لكن لا بأس فيمكن بسهولة إنشاء خدمة RSS بأي لغة تطوير و يمكن ان اضع امثلة لاحقاً بإذن الله تعالى بالـ C# أو لمبرمجي الـ PHP أو الـ ASP الكلاسيكية أيضاً ... و سأكتفي في هذا المقام بالـ ASP.NET :

```

Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)

    Dim CnnStr As String = "Provider=Microsoft.Jet.OleDB.4.0; Data Source="
    Dim Cnn As New OleDbConnection( CnnStr & Server.MapPath("data.mdb"))
    Dim Cmd As New OleDbCommand("SELECT * FROM test", Cnn)

    Try
        Cnn.Open()
        Dim rad As OleDbDataReader

        rad = Cmd.ExecuteReader(Data.CommandBehavior.CloseConnection)

        Me.Repeater1.DataSource = rad
        Me.Repeater1.DataBind()

        rad.Close()
    Catch ex As Exception
    Finally
        Cmd.Dispose()
        If Cnn.State <> Data.ConnectionState.Closed Then Cnn.Close()
    End Try

End Sub

```

ونرى منه بوضوح ان الأمر بسيط جدا فالتتو قمت بتكوين خدمة RSS في موقعك ، مبارك !! ...
 صحيح لا تنسى ان تضع ال Repeater على الصفحة بشكل مشابه :

```

<rss version="2.0">
    <channel>
        <title>قناة رغيد الاخبارية</title>
        <description>قناة لا تحتوي على شيء مفيد</description>
        <link>http://msdn2.microsoft.com </link>
        <lastBuildDate>Mon, 24 Sep 2007 07:33:37 -0700</lastBuildDate>
        <pubDate>Wed, 21 Nov 2007 14:12:12 -0700 </pubDate>
    <asp:repeater id="Repeater1" runat="server">
        <ItemTemplate>
            <item>
                <title><%#DataBinder.Eval(Container.DataItem, "title")%></title>
                <description><%#DataBinder.Eval(Container.DataItem, "desc")%></description>
                <link><%#DataBinder.Eval(Container.DataItem, "desc")%></link>
                <pubDate><%#DataBinder.Eval(Container.DataItem, "posttime")%></pubDate>
            </item>
        </ItemTemplate>
    </asp:repeater>
</channel>
</rss>

```

من وجهة نظر أمثالي (مبرمجون مستهلكون !)

لن أطيل الحديث مع المبرمجون الذين يريدون أن يقوموا بتزويد خدمة RSS و سأتجه إلى المبرمجين المستهلكين (أمثالي) و سأساعدكم على تنمية هذا التوجه الإستهلاكي !!

كثيرة هي المواقع التي تزود خدمات الـ RSS و إن كانت قليلة على المستوى العربي للإسف الشديد غير أنني أكيد بإذن الله تعالى من كون هذا التوجه يزداد عربيا بحمد الله تعالى و نسأل الله أن نكون مساهمون في كل خير بإذنه سبحانه.

تخيل أنك تريد أن تقوم باستغلال خدمة الـ RSS في أحد برامجك، فماذا يلزمك حينها ؟

يلزمك حينها أن تقوم بكتابة كود خاص بتحميل بيانات الخدمة من عنوانها فمثلا لنفرض أنك تريد الاشتراك بخدمة آخر المقالات من موقع www.CodeProject.com فما يلزمك حينها أن تقوم بتحميل البيانات من عنوان الخدمة و الذي هو:

<http://www.codeproject.com/webservices/articlerss.aspx?cat=1>

ويمكن ذلك بعدة طرق عن طريق الـ Visual Basic.NET مثلا باستخدام الفئة WebClient وتجدها في المسار System.Net.WebClient وباستخدام الدالة DownloadString() يمكنك أن تقوم بتحميل كافة بيانات صفحة الخدمة إلى متغير نصي بشكل مشابه:

```
Dim webSite As New System.Net.WebClient()  
Dim rssString, URL As String  
  
"http://www.codeproject.com/webservices/articlerss.aspx?cat=1" URL = "  
  
rssString = webSite.DownloadString(URL)
```

في السطرين اعلاه ستكون لديك البيانات كلها من الموقع مباشرة مخزنة في المتغير rssString و يمكنك الان التعامل معها كما تشاء.

ملاحظة:

يمكن بالطبع استخدام xmlTextReader بحيث تقم بالقراءة مباشرة من الموقع بشكل متواصل عنصر بعد عنصر حتى النهاية، حيث أن الـ WebClient تقوم بتحميل مرة واحدة من البداية إلى النهاية و يعني ذلك الإنتظار حتى تنتهي من عملها و بالتالي القراءة المتواصلة عن طريق الـ xmlTextReader ستكون اسرع قليلا.

و سأستخدم أنا في مثالي هنا الطريقة الاولى لسهولة الفهم ... يعني و بعد أن يكون لدينا كل ما نريد في المتغير rssString نستطيع أن نقوم بتحليله على مهل لإستخراج البيانات منه (البيانات اعني عناصر الـ RSS و التي ذكرناها في بداية المقال – لاتنسى تركيب ملف الـ RSS) .

ويمكن أيضا لتسهيل العمل تحميل النص كاملا إلى ملف XML بحيث يسهل التحرك خلاله ... بالشكل التالي :

```
Dim webSite As New System.Net.WebClient()
Dim rssString, URL As String

"http://www.codeproject.com/webservices/articlerrss.aspx?cat=1" URL = ""

rssString = webSite.DownloadString(URL)

Dim doc As New XmlDocument()
doc.LoadXml(rssString)
```

والان فالتفرح ... فقد حملنا كل البيانات في ملف XmlDocument فيه كل بياناتنا وما تبقى علينا هو دوران بسيط خلال هذا الملف وال Nodes الموجودة فيه و إستخراج عناصر ال RSS السابقة الذكر مثل (rss,Channel,Item,Image) بشكل مشابه للتالي :

```
Dim webSite As New System.Net.WebClient()
Dim rssString, URL As String

"http://www.codeproject.com/webservices/articlerrss.aspx?cat=1" URL = ""

rssString = webSite.DownloadString(URL)

Dim doc As New XmlDocument()
doc.LoadXml(mXMLSource)

Dim Channel As XmlNode = Nothing

' Look for the Channel Element.
For Each node As XmlNode In doc.ChildNodes
    If node.Name.ToLower = "rss" Then
        ...
    ElseIf node.Name.ToLower = "channel" Then
        ...
Next
....
```

الامر سيكون بعدها سهل عليك بإذن الله تعالى .. و لكي لا اجعل رأسك يصاب بالصداع ... يمكنك استغلال الفئات التي سنناقشها سريعا في الفقرة التالية لتصميم برامجك و تطويرها كما تشاء.

نظرة سريعة على فئات الـ rtaRSS

اما الاحرف الثلاثة الاولى في إسم الفئات فلا تسألني عنها فلا اعرف من اي جاءت ؟!

خاصة اذا عرفت أن اسمي هو (رعيد الطيب أحمد أو Ragheed Al-Tayeb Ahmed) و اسم الفئات هو rtaRSS و اما الاحرف البقية فلا اظنها تخفى عليك RSS (سوف اصاب بإحباط شديد إذا بدأت بالتساؤل عن معناها ؟!)

على العموم هن ثلاث فئات الاولى rtaRSSChanel و الثانية rtaRSSItem و الثالثة rtaRSSImage و يمكن استخدامهن لإعطائك وصول سريع لخدمات الـ RSS دون عنا كتابة الكود الكثير فلكني تقوم بتحميل كل بيانات خدمة المقالات الاخيرة من موقع الـ CodeProject فيمكنك ان تقوم بالتالي :

```
Dim rssChannel As New rtaRSSChannel()  
Dim rssItem As New rtaRSSItem  
  
rssChannel.URL = "http://www.codeproject.com/webservices/articlerrss.aspx?cat=1"  
rssChannel.UpdateNow()  
  
For Each rssItem In rssChannel.Items  
    Dim Title As String = rssItem.Properties("title")  
    ListBox1.Items.Add(Title)  
Next
```

في الحقيقة الكود السابق هو كل ما يلزمك لرؤية كل عناوين المقالات الأخيرة في موقع CodeProject في القائمة ListBox1 التي في برنامجك !

و لبيان الفئات دعني اذكر بسرعة اهم عناصر الفئة الرئيسية في مشروعي و هي **rtaRSSChannel**

تحتوي هذه الفئة على خاصية مهمة هي **URL** عنوان الموقع أو بالأصح عنوان خدمة الـ RSS التي تريد الارتباط بها.

xmlSource الخاصية

و تحتوي على النص كاملا الذي تم تحميله من الموقع (المصدر).

Updating الخاصية

وهي عبارة عن خاصية لفحص حالة الفئة **True** يعني انه جاري التحديث أو التحميل من الانترنت و **False** تعني لاشيء(Ready).

Items الخاصية

وهي الاهم حيث تعيد مصفوفة بكل العناصر التي تم تحميلها وهي من النوع **rtaRSSItem** وسيتم وصفها لاحقا.

Image الخاصية

تعيد الصورة المرتبطة بالقناة **channel** (في حال وجودها) وهي من النوع **rtaRSSImage**.

الخاصية **LastError**

وتعيد وصف آخر خطأ حصل داخل الدالة اثناء التحميل أو خلافة ويمكنك استغلالها لتشخيص المشاكل.

الخاصية **Properties**

وتعيد كل خواص الفئة مثلا اذا اردت عنوان القناة فأستخدم ال title كما جاء في وصف ملف ال RSS وهكذا بالنسبة لبقية الخواص.

والان ننتقل إلى الحديث عن الدوال أو الاوامر Methods تحتوي على اثنتين مهمتين هما :

الأمر **UpdateNow**

وهو المستخدم في المثال السابق و الذي يقوم بتحميل البيانات الخاصة بالخدمة من الانترنت ويعيد True في حالة نجاح العملية.

الأمر **Update**

يقوم بنفس عمل الامر السابق غير انه ليس متزامن اي انه لا ينتظر حتى تنتفذ العملية و إنما يبدأها في Thread مستقل ليسمح لك بإكمال شغرك و في حال نجاح العملية وتحميل البيانات يفجر الحدث **Updated** في حال الفشل يفجر الحدث **UpdateError**.

و بالتالي ففي حال انك اردت استخدامه سيتحول مثالك السابق إلى :

```
Private WithEvents rssChannel As New rtaRSSChannel()  
  
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
    rssChannel.URL = "http://www.codeproject.com/webservices/articlerrss.aspx?cat=1"  
    rssChannel.Update()  
End Sub  
  
Private Sub rssChannel_Updated(ByVal Sender As Object) Handles rssChannel.Updated  
    For Each Item As rtaRSSItem In rssChannel.Items  
        Me.ListBox1.Items.Add(Item.Properties("title").ToString)  
    Next  
End Sub  
  
Private Sub rssChannel_UpdateError(ByVal Sender As rtaRSSChannel, ByVal ErrorString  
As String) Handles rssChannel.UpdateError  
    MsgBox("Error: " & ErrorString)  
End Sub
```

فعند الضغط على الزر Button1 سيتم إستدعاء الأمر Update و الذي بدوره سيقوم بإستدعاء اما الحدث Updated في حال نجاح العملية أو UpdateError في حال العكس.

لا ينبغي ان انبهك بضرورة الحرص على استخدام ال WithEvents في حال كونك تريد استخدام احداث هذه الفئة كما جاء في السطر الاول من الكود اعلاه.

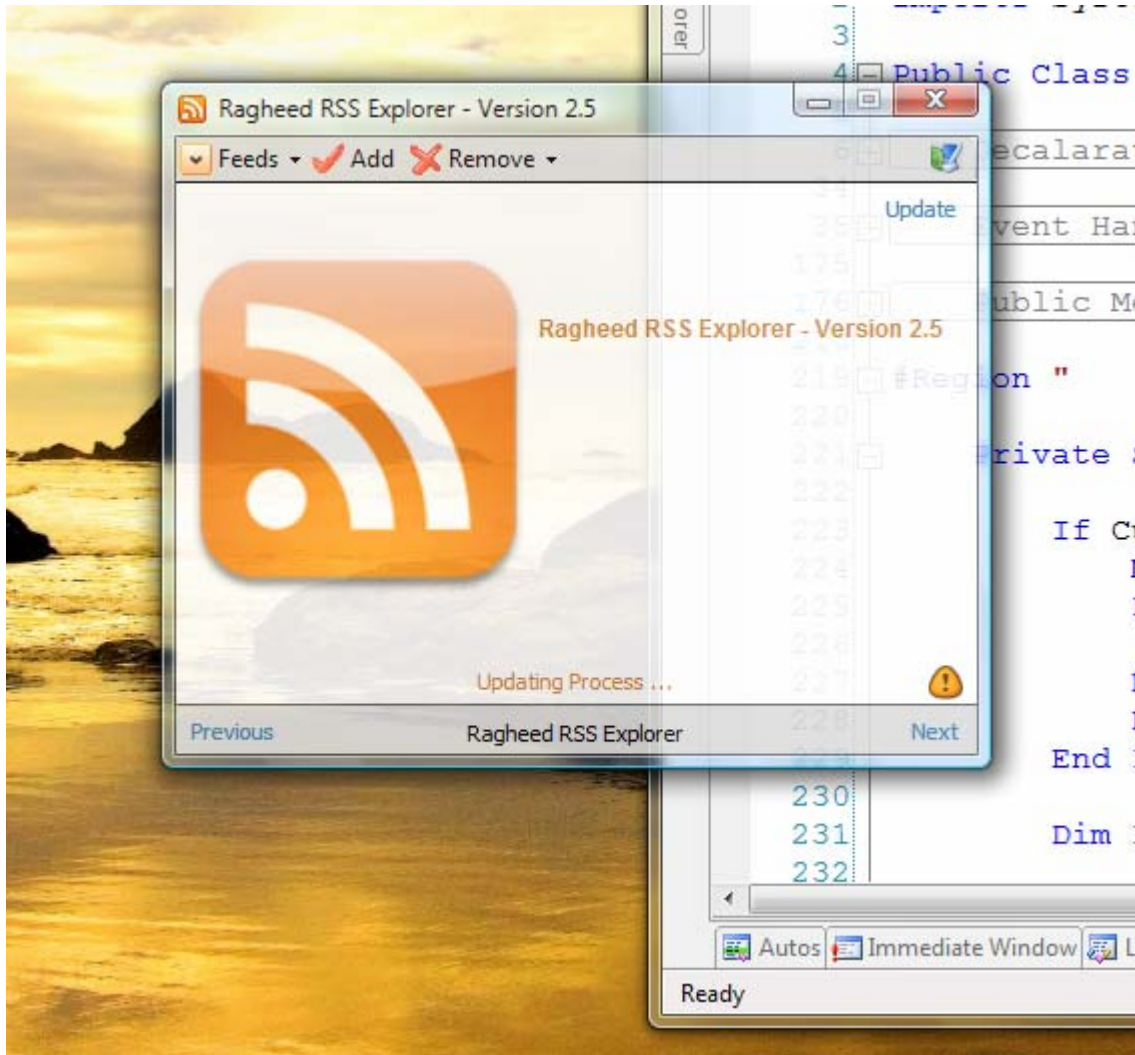
```
Private WithEvents rssChannel As New rtaRSSChannel()
```

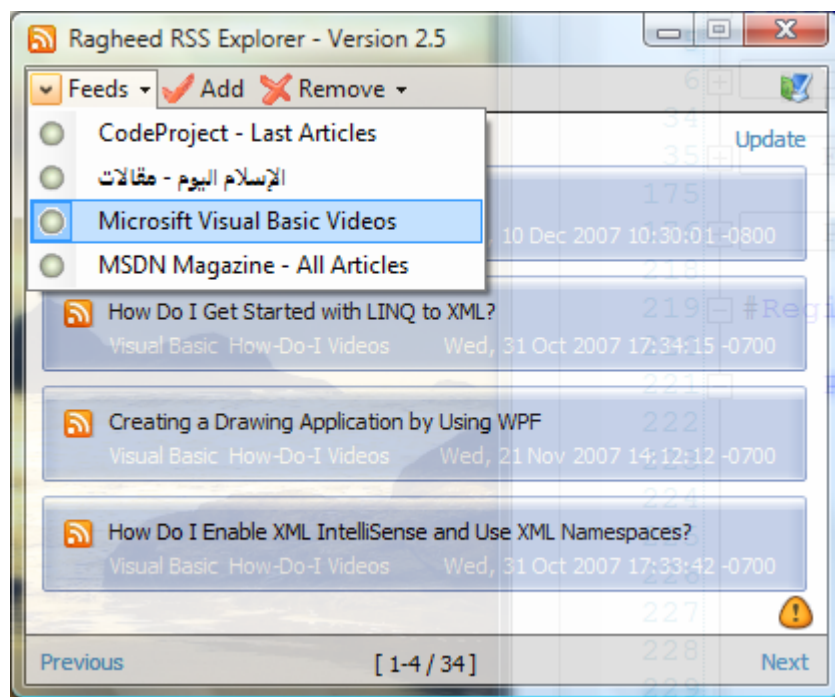
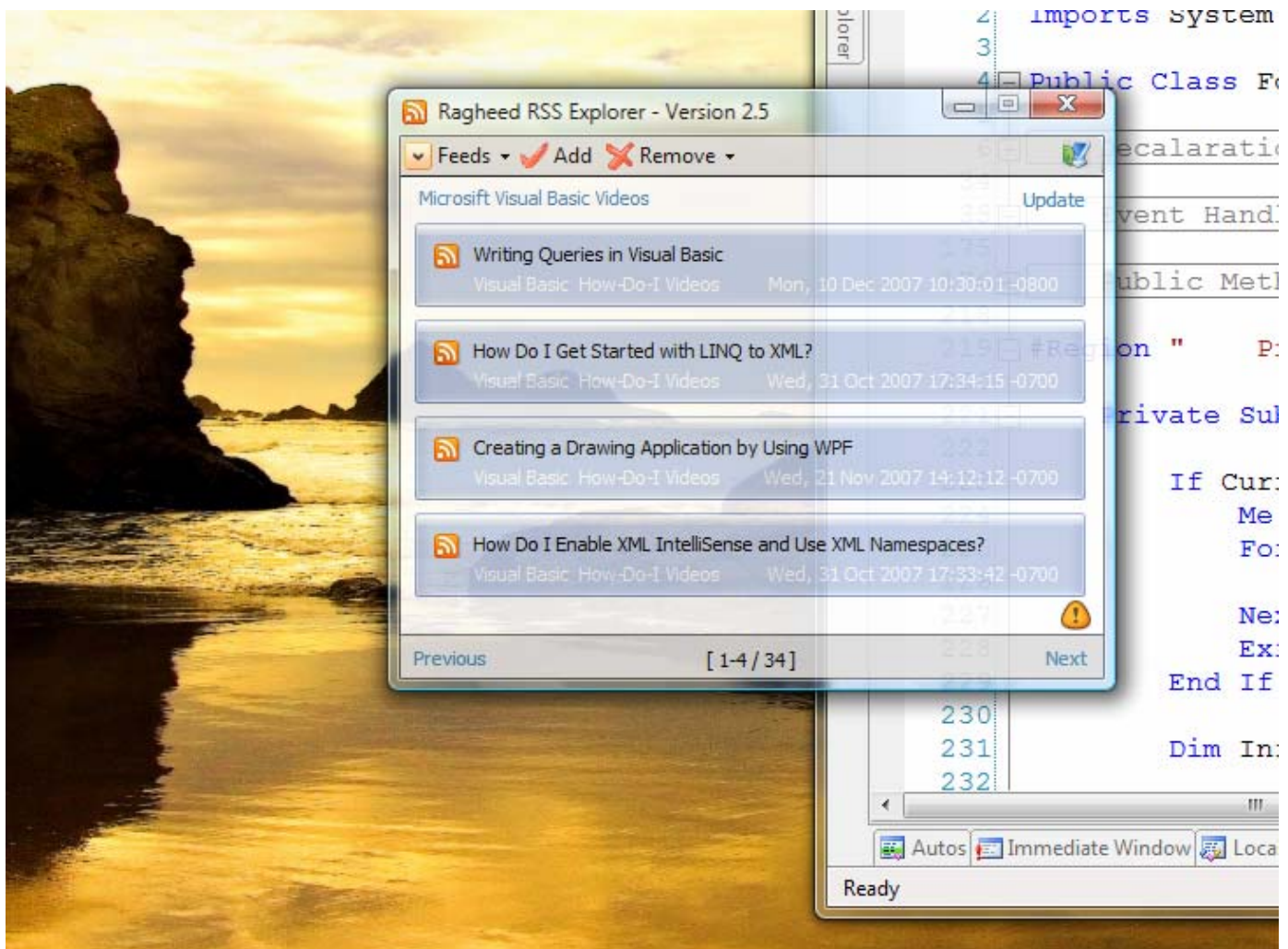
فلا يوجد الكثير لأحدث عنه فهناك الخاصية ChannelTitle و الذي يحتوي على عنوان القناة الذي يتواجد هذا العنصر فيها كما ان اهم خصائصه هي الخاصية Properties و التي تحتوي على جميع خصائص العنصر كال title, description, pubDate, link و خلافه.

و كذلك الحال بالنسبة للفئة **rtaRSSImage** وحقيقة هي فئة لم اختبارها بعد (يمكنك ان تختبرها و تفيدني بالنتائج) و المهم في الموضوع هو ان الفئة لازالت فئة ناضجة ولكن يمكنك استخدامها لبناء برامج تستخدم خدمة ال RSS بسهولة أو يمكنك تطويرها ايضا كما تشاء.

ستجد مرفقا مع هذه المقالة مثال بسيط يستخدم هذه الفئات بإذن الله تعالى لعله يفيدك.

قبل أن اتركك اخبرك بانه سيكون لنا لقاء بإذن الله تعالى مع برنامج بسيط بنيته اعتمادا على هذه الفئة البسيطة يقوم بتصفح خدمات ال RSS التي قمت بالاشتراك بها واستعراضها بالشكل الذي يروقني، وفي الاسفل بعض اللقطات السريعة له.





حقيقة لقد اطلت كثيرا في الموضوع ولكن سيكون من الجميل ان اضع آخر صورة كمثال عن الحال اذا قمت بتفعيل الخدمة بالنسبة للمنتديات التي تشارك فيها و لنقل كمثال اذا اردت ان اراقب المواضيع الجديدة في موقع منتدى الفريق العربي للبرمجة - قسم ال Visual Basic .NET

ما احتاج إلى معرفته هو مسار خدمة الـ RSS لذلك القسم وهي :

<http://www.arabteam2000-forum.com/index.php?act=rssout&id=26>



لنا لقاء (ربما لغات أخرى) مع تطبيقات لهذه الخدمة و إستغلالها في برامج متنوعة بإذن الله تعالى.

السلام عليكم و رحمة الله و بركاته

الكود الكامل للغات التواصل بخدمة ال RSS .. وهي ال rtaRSS

```
Imports System.Xml
Imports System.Net

Class rtaRSSChannel

#Region " Declaraction "

    Public Event Updated(ByVal Sender As rtaRSSChannel)
    Public Event UpdateError(ByVal Sender As rtaRSSChannel, ByVal ErrorString As String)

    Sub New()

    End Sub

    Sub New(ByVal URL As String)
        mUrl = URL
    End Sub

#End Region

#Region " Public Methods "

    Function UpdateNow() As Boolean
        mUpdating = True

        Try
            Dim web As New System.Net.WebClient()
            Dim xmlString As String = web.DownloadString(mUrl)

            mXMLSource = xmlString.Substring(xmlString.IndexOf("<rss"))

            Return ReadXMLSource()
        Catch ex As Exception
            mUpdating = False
            mLastError = ex.Message
            RaiseEvent UpdateError(Me, ex.Message)
            Return False
        End Try
    End Function

    Sub Update()
        Dim Trd As New Threading.Thread(AddressOf StartDownloading)
        mUpdating = True
        Trd.Start()
    End Sub

#End Region

#Region " Private Methods "

    Private Sub StartDownloading()
        Try

            Dim web As New System.Net.WebClient()
            Dim xmlString As String = web.DownloadString(mUrl)
            Dim r As New Xml.XmlTextReader("d")

            mXMLSource = xmlString.Substring(xmlString.IndexOf("<rss"))

            mUpdating = False

            If ReadXMLSource() Then
```

```

        RaiseEvent Updated(Me)
    Else
        RaiseEvent UpdateError(Me, mLastError)
    End If

Catch ex As Exception
    mUpdating = False
    mLastError = ex.Message
    RaiseEvent UpdateError(Me, ex.Message)
End Try
End Sub

Private Function ReadXMLSource() As Boolean
    Try
        mItems.Clear()
        mProperties.Clear()
        mImage = New rtaRSSImage

        Dim doc As New XmlDocument()
        doc.LoadXml(mXMLSource)

        Dim Channel As XmlNode = Nothing

        ' Look for the Channel Element.
        For Each node As XmlNode In doc.ChildNodes
            If node.Name.ToLower = "rss" Then
                For Each cnl As XmlNode In node.ChildNodes
                    If cnl.Name.ToLower = "channel" Then
                        Channel = cnl
                        Exit For
                    End If
                Next
            Exit For
        End If
    Next

    ' Go into its items.
    If Channel IsNot Nothing Then

        Dim feed As rtaRSSItem
        Dim title As String = ""

        For Each element As XmlNode In Channel.ChildNodes
            Select Case element.Name.ToLower
                Case "item"
                    feed = New rtaRSSItem(element)
                    feed.ChannelTitle = title
                    mItems.Add(feed)
                Case "image"
                    mImage = New rtaRSSImage(element)
                Case Else
                    If element.Name.ToLower = "title" Then
                        title = element.InnerText
                    End If

                    If Not mProperties.Contains(element.Name) Then
                        mProperties.Add(element.Name, element.InnerText)
                    End If
                End Select
            End Select
        Next

        End If

        Return True
    Catch ex As Exception
        mLastError = ex.Message
        Return False
    End Try
End Function

```

End Function

#End Region

#Region " Public Properties"

```
Private mUrl As String = ""
Property URL() As String
    Get
        Return mUrl
    End Get
    Set(ByVal value As String)
        mUrl = value
    End Set
End Property
```

```
Private mXMLSource As String
Property XMLSource() As String
    Get
        Return mXMLSource
    End Get
    Set(ByVal value As String)
        mXMLSource = value
        ReadXMLSource()
    End Set
End Property
```

```
Private mTag As Object
Property Tag() As Object
    Get
        Return mTag
    End Get
    Set(ByVal value As Object)
        mTag = value
    End Set
End Property
```

```
Private mLastError As String = ""
ReadOnly Property LastError() As String
    Get
        Return mLastError
    End Get
End Property
```

```
Private mProperties As New Hashtable
ReadOnly Property Properties() As Hashtable
    Get
        Return mProperties
    End Get
End Property
```

```
Private mItems As New List(Of rtarRSSItem)
ReadOnly Property Items() As List(Of rtarRSSItem)
    Get
        Return mItems
    End Get
End Property
```

```
Private mImage As New rtarRSSImage
ReadOnly Property Image() As rtarRSSImage
    Get
        Return (mImage)
    End Get
End Property
```

```
Private mUpdating As Boolean = False
ReadOnly Property Updating() As Boolean
    Get
```

```

        Return mUpdating
    End Get
End Property
#End Region

End Class

Class rtaRSSItem
    'Implements

    Sub New()

    End Sub

    Private mChannelTitle As String = ""
    Property ChannelTitle() As String
        Get
            Return (mChannelTitle)
        End Get
        Set(ByVal value As String)
            mChannelTitle = value
        End Set
    End Property

    Sub New(ByVal Node As XmlNode)
        If Node.NodeType = XmlNodeType.Element AndAlso Node.Name.ToLower = "item" Then
            For Each Item As XmlNode In Node.ChildNodes
                If Item.Name.ToLower = "image" Then
                    mImage = New rtaRSSImage(Item)
                Else
                    mProps.Add(Item.Name, Item.InnerText)
                End If
            Next
        End If
    End Sub

    Private mProps As New Hashtable
    ReadOnly Property Properties() As Hashtable
        Get
            Return mProps
        End Get
    End Property

    Private mImage As New rtaRSSImage
    ReadOnly Property Image() As rtaRSSImage
        Get
            Return mImage
        End Get
    End Property

    Public Overrides Function ToString() As String
        Return "rssItem: " & mProps("title")
    End Function

    Private mTag As Object
    Property Tag() As Object
        Get
            Return mTag
        End Get
        Set(ByVal value As Object)
            mTag = value
        End Set
    End Property
End Class

Class rtaRSSImage

```

```

Sub New()

End Sub

Sub New(ByVal Node As XmlNode)
    If Node.NodeType = XmlNodeType.Element AndAlso Node.Name.ToLower = "image"
Then
        For Each Item As XmlNode In Node.ChildNodes
            mProps.Add(Item.Name, Item.InnerText)
        Next
    End If
End Sub

Private mProps As New Hashtable
ReadOnly Property Properties() As Hashtable
    Get
        Return mProps
    End Get
End Property

Public Overrides Function ToString() As String
    Return ""
End Function

End Class

```

ثم بحمد الله تعالى