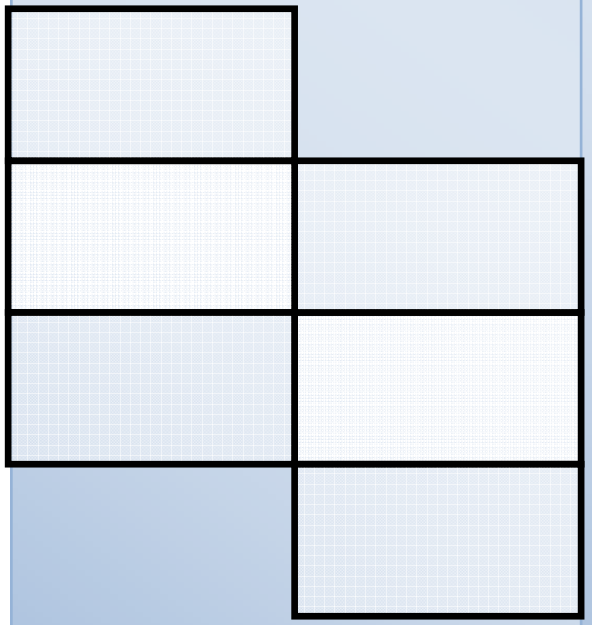
العدد الأول
يوليوز ٢٠٠٦

مجلة الفريق العربي للبرمجة



نظرة على جديد الفيجوال بيسك ٩

كاتب الموضوع : **Crazy-Man**

مشروعLINQ

مشروع ال LINQ او ما يعرف بال Language Integrated Query و هو تقنية جديدة في الفيجوال بيسك ٩ و السي شارب ٣ و هدفها هو تمكين مبرمجي الدوت نت من أن يستعملوا لغات البرمجة التي يستعملونها دائما مثل الفيجوال بيسك و السي شارب من أجل عمل Queryعلى معلومات او بيانات كيفما كان نوعها. يمكن عمل Query على بقواعد بيانات (Project DLinq) أو حتى على بيانات من نوع XMLبما يسمى Project XLinq لكن أعود و أقول انه يمكن عمل Queryعلى أي نوع من البيانات و سيتضح كلامي في باقي المقالة.

الآن من أجل تقريب الفكرة الى جميع دعونا نرى مثال بال Visual Basic 9و سيمكنك تجربة المثال حالما يظهر الفيجوال بيسك ٩، لنفرض أن لدينا كلاس اسمه Personne و متغير اسمه Personnes يحتوي على Collection من الأشخاص ، لنرى اذا هذا الكود الذي أظن أنه جديد بالنسبة لك لكن رغم ذلك أضمن لك أنه جد شهل و سيمكنك فهمه باذن الله من النظرة الأولى.

```
Dim OurResults = From Personne In Personnes_
_
        Select Personne
Where Personne.Age > 50_
AndAlso Personne.Country = "Morocco"_
Order By Personne.Nom ASC
```

الآن المتغير OurResults عبارة عن collection (IEnumarable)للأشخاص و قد قمنا بأخذ هذه المجموعة من المتغير Personnes الذي سبقو ذكرنا أنه يحتوي على Collection من الأشخاص و طبعا كما ترون فقد اخترنا فقط بعض الأشخاص طبقا لمواصفات محددة يعني الأشخاص ذوو العمر أكبر من ٥٠ سنة و أيضا من المغرب ثم رتبنا النتائج حسب الأسماء، كما ترون فهذه جملة SQLواضحة لكنها استعملت بواسطة الفيجوال بيسك ٩ دون الحاجة الى ال SQL و أيضا اذا لم تلاحظة فقد عملنا Query ليس على قاعدة بينات و انما على متغير، أليس هذا رائعا.

استعمال متغيرات دون تحديد النوع:

هل لاحظتم شيئا في الكود السابق ؟؟؟ المتغير ... OurResults نعم صحيح أحسنت هذا هو، اننا لم نحدد له نوع يعني عرفنا المتغير دون ان نحدد نوعه هل هو String ام Array ام ماذا !!! اذا فلتفرح لم يعد هناك حاجة لتحديد نوع المتغير لأن الكومبايلر الخاص بالفيجوال بيسك ٩ أصبح يستطيع أن يتعرف على نوع المتغير مباشرة عندما تسند اليه قيمة ما فيقوم بمعرفة النوع على حسب القيمة حتى لو كنت عامل الخاصية Option Strict التي تلزمك بتعرف المتغيرات قبل استعمالها.

في المثال السابق يعرف الكومبايلر أنني سأستعمل IEnumarable اذا فلا حاجة لتحديد النوع مسبقا.

طبعا هناك استثناءات فمثلا لنأخذ المثال التالي:

```
Dim x = Nothing
```

هنا سينتج خطأ كون الكومبايلر سيجهل نوع المتغير لكن من جهة أخرى لنرى التعبيرات التي يمكننا استعمالها بكل بساطة:

```
Dim x = "coucou"
Dim y = 2.5
Dim z As Integer = 4
Dim i = z
```

عموما الكومبايلر قد يقوم بانشاء أنواع جديدة تلائم متغيراتك فمثلا في المثال أعلاه نحن نستعمل أنواع متغيرات موجودة مسبقا في الفريميم وورك لكن يمكن للكومبايلر عمل انواع تلائم تعاملك مع ملفات ال XML مثلا.

تعريف الكائنات

تعرفون بدون شك الجملة With التي تمكننا من الوصول الى عناصر كائن معين دون اعادة كتابته و التي يكون شكلها كالتالي:

```
Dim p As New Personne
With p
.Nom = "Clark"
.Prenom = "Richard"
End With
```

رائع، الآن و مع مشروع LINQ يمكن استعمال نفس الطريقة لكن دون اللجوء الى كلمة With من أجل تعريف كائن و لتقريب الفكرة أكثر دعونا نرى مثال لذلك:

```
Dim p as New Personne{.Nom = "Clark", .Prenom
                                    = "Richard"}
```

و لنستغل ما رأيناه سابقا حول تعر المتغيرات دون تحديد نوعها لنكتب شيئا من هذا القبيل:

```
Dim p = New {.Nom = "Clark", .Prenom =
"Richard"}
```

طبعا هنا الكومبايلر سيعرف ان المتغير p من نوع Personne بالاعتماد على القيمة المدخلة اليه.

بل و أكثر من ذلك يمكننا عمل Collection من الأشخاص بكل بساطة على الشكل التالي:

```
Dim Personnes = {_
```

```
New Personne{.Nom = "Clark", .Prenom =
"Richard"}_

New Personne{.Nom = "Clark", .Prenom =
"Sabine"}_

}
```

أليس هذا رائعا، و طبعا يمكنك عمل Query على هذه ال Collection مثل التي عملنا في المثال الأول.

الأنواع المجهولة(Anonymous Types)

هنا سنرى شئ رائع و جميل و قوي في نفس الوقت، يعني الروعة الى ما لا حدود، المهم ماذا نريد أن نعرف الآن سوف نرى كيف يمكننا اختيار قيم من داخل كائن.كيف ذلك ؟؟ اذا انظر الى الكود أسفله و ستفهم:

```
Dim Personnes = {_
New Personne{.Name = "Clark", .Prenom =
"Richard", Age = 40}_
New Personne{.Name = "Clark", .Prenom =
"Sabine", Age = 41}_
}
Dim OurPersonnes = Select New{.Nom =
Personne.Name, .Age = Personne.Age }_
From Personne In Personnes_
Where(Personne.Age < 40)
```

يبدو ان المثال غامض لكن لا مشكل دعوني أشرح:
أولا قمنا بانشاء Collection اسمها Personnes طبعا تحتوي على أشخاص و بالتحديد شخصين.
ثم بعد ذلك قمنا بتعريف متغير OurPersonnes طبعا لم نحدد له النوع، الى هنا كل الأمور واضحة لكن على ماذا يحتوي هذا المتغير ؟؟؟ كما ترون عملنا Query ثم أنشأنا كائن جديد بواسطة الكلمة Newلكن لم نعطه اسم!!! ثم حددنا له فقط خاصيتين و هما Name و ال Age و بهذا أصبح المتغير OurPersonnes عبارة عن Collection تحتوي على خاصيتين هما Name و Age .

راينا ان لدينا من بين الامكانيات الجديدة التعريف دون تحديد نوع المتغيرات (Implied Types)و ايضا ال Anonymous Types و لكن رغم ذلك هذه الأمور لا تشرح لنا بشكل دقيق كيفية و امكانية كتابة هذا الكود:

```
Dim mesPersonnes = Select New{.Nom =
Personne.Nom, .Age = Personne.Age }_

From Personne In Personnes_

Where(Personne.Age < 40)
```

من أجل فهم جيد للكود اعلاه يجب ان نعرف ما يسمى بال Extensions Methodsو هو ما سنتكلم عليه في الفقرة الموالية.

Extension Method

هذا النوع من ال Methods هو عبارة عن Shared Methods عندما تنشأها في كلاس معين مثلا تدخل نفسها كخاصية في جميع الكائنات الموجودة في الكلاس باستثناء تلك الموجودة فيها أصلا. كيف ذلك ؟ طبعا سوف نرى عن طريق مثال بسيط ماذا نعني بهذا الأمر

لكن قبل ذلك اود أن أشير الى معلومة بسيطة هي أن تعابير الكود أعلاه مثل Select و From و Where هي عبارة عن Extensions Methods .

الآن دعونا نشرح جيدا معنى ال Extensions Methods و ذلك عن طريق مثال تطبيقي:

سوف نقوم الآن بانشاء Extension method قد لا تكون مفيدة بشكل كبير لكنها تنفع لأن تكون مثال لتقريب الصورة و توضيحها أكثر، ال Extension Methodالتي نريد إنشاءها الآن تقوم بأمر بسيط ألا و هو كتابة محتوي متغير من نوع IEnumerable في شاشة الدوس.

أولا لنرى معا هذا الكود البسيط:

قمنا في البدأ باعلان جدول String و طبعا أعطيناه قيم محددة

```
Dim Personnes = {_

New Personne{.Name = "Jessy", .Prenom =
"Richard", .Age=20}_

New Personne{.Name = "Clark", .Prenom =
"Sabine", .Age=21}_

New Personne{.Name = "Crazy", .Prenom =
"Man", .Age=21}_

}
```

ثم بعد ذلك كما ترى هذا الكود عبارة عن LINQ Query و لاحظ معي أننا لم نحدد نوع المتغير

```
Dim OurResults = Select Personne From
Personne In Personnes_

Where Personne.Age = 21
```

اذا عملنا Query على المتغير Personnes و أخذنا الأشخاص ذوو السن ٢١ اذا اذا ما أردنا ان نكتب النتائج مثلا على شاشة الدوس (Console Application)ما علينا فعله جد سهل حيث أن الكود سيكون كالتالي:

```
For Each Personne In OurResults

Console.WriteLine(Personne.Name)

Next
```

و النتيجة ستكون على الشكل التالي:

Jessy

Crazy

الى هنا كل شئ واضح لكن دعونا كيف يمكننا ان نعلن Extension Methodو قبل لماذا نريدها أصلا ؟؟؟

نحن نريد الآن عندما نرغب في كتابة تلك المعلومات على

الشاشة ان نستدعي ال Method بواسطة المتغير نفسه أي أن ال Methodالخاصة بكتابة النتائج على الشاشة ستصبح عنصرا من عناصر المتغير. دعوني أقرب الأمر و ليكن ذلك عن طريق تعريف ال Extension Methodالخاصة بنا و سيكون اسمها مثلا: Write

```
Public Shared Sub Write(Of T)(ByVal [Me] As
IEnumarable(Of T))

For Each Personne In T

Console.WriteLine(Personne.Name)

Next

End Sub
```

كما ترون لا يوجد فرق بين هذه ال Method و ال Methods العاديين سوى وجود Me و التي تعتبر مؤشر للكائن الذي يستدعي ال Method و T هنا تعبر عن ال Collection الممررة الى ال Extension Methodو الكود الخاص بالحلقة التكرارية طبعا واضح.

الآن بعد أن أنشأنا Extension Method خاصة بنا ماذا يمكننا ان نفعل بها ؟؟ بكل بساطة يمكن بعد ان أنشأنها كتابة الكود التالي دون مشاكل:

```
OurResults.Write()
```

هكذا يمكننا ان نحصل على النتيجة بكل بساطة بل و يمكننا ان نفعل اكثر من ذلك كون ال Extension Method مشتركة: (Shared)

```
Write(OurResuts)
```

اذا الآن يمكن ان يتضح لنا كيف تشتغل اكواد ال LINQ يعني مثل ال Selectو Where فمثلا لو أخذنا Where فستكون تقوم فقط باختبار شرط مثلا Name="Muslim" و ترجع نوع Boolean.
الى هنا و ينتهي المقال و اتمنى ان يكون قد أفادكم و لو قليلا
و السلام عليكم و رحمة الله و بركاته

الإجـــــــــــــــراءات في الفيجول بيسيك دوت نت

كاتب الموضوع : عبد الله فتحي

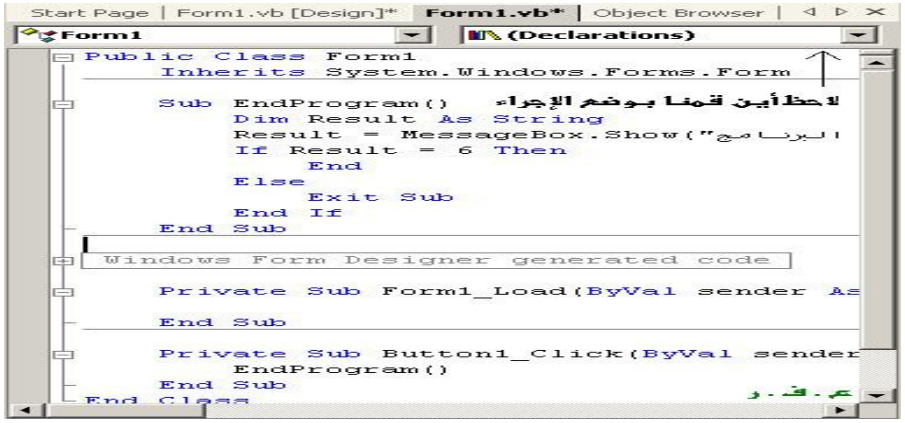
مقدمة :

سنتناول اليوم موضوعاً من مواضيع الفيجول بيسيك المهمة، وهو ليس بموضوع جديد أو تم استحداثه مع الفيجول بيسيك دوت نت، حيث أنه موجود في الإصدارات السابقة، ولما له من أهمية فقد أفردنا له هذه السطور..

ما هو الإجراء ؟

الإجراء - بكل بساطة - هو عبارة عن مجموعة من الجمل والأوامر العادية، يتم تجميعها معاً، وإعطائها عنواناً معيناً كاسم لهذا الإجراء .. وبعد ذلك يمكننا استخدام هذه الأوامر جميعها في أي مكان من البرنامج، عن طريق كتابة اسم هذا الإجراء..

وسنعرض الآن **مثالاً** بسيطاً لذلك:
أولاً سنقوم بإنشاء إجراء جديد اسمه EndProgram وستكون وظيفته الأساسية عرض رسالة تأكيد للمستخدم عند محاولته الخروج من البرنامج..
لاحظ أننا سنقوم بكتابة هذا الإجراء في قسم ال Declarations الخاص بالنموذج وليس في أي مكان آخر .. انظر للصورة التالية:



وسيكون الكود كما في الصورة:

```
Sub EndProgram()
      Dim Result As String

Result = MessageBox.Show(" هل تريد بالتأكيد الخروج
من البرنامج", "تأكيد الخروج",
MessageBoxButtons.YesNo, MessageBoxIcon.Question,
MessageBoxDefaultButton.Button2,
MessageBoxOptions.RightAlign)
```

```
        If Result = 6 Then
            End
        Else
            Exit Sub
        End If
    End Sub
```

لاحظ أننا - في الإجراء السابق - عند ضغط المستخدم على/Yes نعم فإننا نقوم بتنفيذ الأمرEnd ، بينما لو قام المستخدم بالضغط على/No لا فإننا نقوم بالخروج من الإجراء عن طريق الأمر.. Exit Sub
الآن لاستخدام الإجراء بشكل فعلي قم بوضع زر على النموذج واجعل عنوانه" = Caption/خروج.. "
اضغط على هذا الزر مرتين للانتقال لنافذة الكود..
قم بكتابة الأمر Call ثم اكتب اسم الإجراء..
ويمكنك الاستغناء عن الأمر Call وكتابة اسم الإجراء مباشرة ولن يؤثر ذلك على شيء..
إذاً بإمكاننا أن نكتب في حدث النقر على الزر:

```
Call EndProgram()
```

أو

```
EndProgram()
```

تهانينا ..لقد استخدمت الإجراء بنجاح..

ما هي فائدة الإجراءات ؟

لا نريد أن نعدد في فوائد الإجراءات وهي كثيرة بالفعل، وتتحدد هذه الفائدة بحسب الحاجة إليها، ولكن أهم ما يُمكن أن نشير إليه هو:

-دور الإجراءات الواضح في تقليل الكود المكتوب:

على سبيل المثال لو كنا نستخدم في النموذج أكثر من إمكانية لإنهائه (عن طريق زر إنهاء أو عن طريق البند إنهاء من قائمة(..

هل سنقوم بكتابة كود تأكيد الخروج مرتين، الأولى في حدث النقر على الزر إنهاء، والثانية في حدث النقر على البند إنهاء من القائمة ؟؟
بإمكاننا عمل ذلك ولكنه يعني المزيد من الجهد في الكتابة والمزيد من الكود الذي لا فائدة منه ..
الأفضل في مثل هذه الحالة أن نقوم بعمل إجراء كالإجراء EndProgram الذي قمنا بعمله بالأعلى، ثم يمكننا بعد ذلك استدعاء هذا الإجراء من أي مكان في البرنامج عن طريق كتابة اسم الإجراء فقط..

-دور الإجراءات في تنظيم الكود وتقسيمه:

وهذا واضح تماماً حيث أن تقسيم الأكواد إلى إجراءات يجعل الأمر أكثر تنظيماً، كما أنه يجعل من السهل عليك - وعلى غيرك - فهم الكود واستخدامه في أي وقت..
تخيل أنك تريد التغيير في كود يحتوي على آلاف الأسطر المتتالية، إذاً أنت بحاجة إلى قراءة هذه الأسطر كلها، والتغيير في أماكن عديدة، وهذا ما يمكننا تفاديه باستخدام الإجراءات..

تمرير المعاملات أو القيم إلى الإجراء:

من الممكن عند استدعاء الإجراء أن نقوم بتمرير قيمة أو معامل إلى الإجراء، مثلاً لو كان لدينا إجراء يقوم بالتحقق من وجود أرقام فقط في مربع النص أم لا، فمن الطبيعي عند استدعاء هذا الإجراء أننا سنقوم بتمرير اسم

مربع النص - الذي نريد التحقق منه - إليه.
وبصفة العموم فإننا نحتاج عند إتاحة التمرير إلى الإجراء ثلاثة أشياء ..

1.الكلمة ByVal أو الكلمة ByRef.
2.اسم القيمة أو المعامل.
3.نوعهـــا.

ولنأخذ مثالاً على ذلك الإجراء التالي، وهو يقوم بتغيير لون الخلفية وكتابة "بسم الله الرحمن الرحيم" في مربع النص الذي نحدده له، وسيكون كالتالي:

```
   Sub WriteInText(ByVal TextName As Control)
       TextName.BackColor =
System.Drawing.Color.Cyan
       TextName.Text = "بسم الله الرحمن الرحيم"
   End Sub
```

لاحظ أننا استخدمنا الكلمة .. ByVal
وكان من الممكن أن نستخدم الكلمةByRef ، وسيأتي الفرق بينها وبين زميلتها لاحقاً.
ولاحظ أننا استخدمنا لاسم القيمة الكلمة .. TextName
ويمكننا تغيير هذا الاسم لأي اسم نريده إطلاقاً.
ولاحظ أيضاً أننا استخدمنا النوع .. Control
وذلك لأن نوع القيمة الممررة إلى الإجراء هي أداة أي.Control
بعد ذلك يمكننا استخدام هذا الإجراء من أي مكان، وتنفيذه على أي مربع نص عن طريق استدعاء (كتابة اسم) الإجراء وهو WriteInText ثم كتابة اسم مربع النص الذي نريد التطبيق عليه، وفي حالتنا هذه سنجعله TextBox1مع إمكانية التغيير حسب اسم مربع النص..

```
WriteInText(TextBox1)
```

الآن لديك مهلة دقيقتين فقط للتفكير في طريقة تجعلنا نحدد اسم مربع النص وكذلك النص الذي سيُكتب بداخله..

هل عرفت الحل ؟

أعتقد أن الأمر ليس صعباً، فكما أننا قمنا بتمرير اسم مربع النص في المثال السابق، فيمكننا أيضاً بنفس الطريقة تمرير النص المكتوب، حيث أن الإجراء يسمح بتمرير قيمة أو أكثر حسب الحاجة..
وبهذا سيتم تغيير المثال ليصبح الإجراء كالتالي:

```
   Sub WriteInText(ByVal TextName As Control,
ByVal WritedText As String)
       TextName.BackColor =
System.Drawing.Color.Cyan
       TextName.Text = WritedText
   End Sub
```

بعد ذلك نقوم باستدعاء الإجراء مع تمرير القيمتين إليه، كالتالي:

```
WriteInText(TextBox1, "الفريق العربي لتكنولوجيا الويب")
```

والآن ننتقل إلى إمكانية أخرى، ولاحظ أننا نحاول الإلمام بجميع الإمكانيات لاستخدامها فيما بعد طوع إرادتنا
سنسمح الآن للمستخدم بتمرير قيمة النص المكتوب وفي نفس الوقت لن نجبره على ذلك، بمعنى أنه لو قام المستخدم بتمريره فإنه سيظهر كالعادة، وإذا لم يمرر شيئاً فسيقوم الإجراء بتمرير القيمة الافتراضية، والتي سنحددها نحن في الإجراء، وسيكون الكود كالتالي:

```
Sub WriteInText(ByVal TextName As Control,
Optional ByVal WritedText As String = "www.al-
ebda3.info")
       TextName.BackColor =
System.Drawing.Color.Cyan
       TextName.Text = WritedText
   End Sub
```

كما ترى قمنا بإضافة الكلمة optional قبل الكلمةByVal ، كما أننا قمنا بتحديد القيمة الافتراضية والتي سيتم وضعها في حال لم يمرر المستخدم أي قيمة..
لاحظ أنه لابد أن تحدد المعامل أو القيمة الافتراضية..
ولاحظ أنه لابد أن تكون المعاملات - سواء كان واحداً أو أكثر - في آخر الإجراء ولا يمكن أن يتقدم المعامل الاختياري على المعامل الإجباري..

الفرق بين الكلمة ByVal و:ByRef

الكلمة ByVal تقوم بتمرير قيمة المتغير فقط، أي أنه لو قمنا بتمرير متغير إلى الإجراء فإنه سيأخذ نسخة من القيمة الموجودة في المتغير، وبعدها يتعامل مع هذه القيمة بالزيادة أو النقص أو التغيير، ولن يؤثر ذلك شيئاً على القيمة الأصلية الموجودة في المتغير، لأن التغيير كله سيقع على النسخة المأخوذة من قيمة المتغير..

أما الكلمة ByRef فإنها تقوم بتمرير عنوان المتغير نفسه إلى الإجراء، وبالتالي فعند تغيير القيمة الممررة فإن ذلك سيتم في المتغير الأساسي، وسيتغير بناء على ذلك..

مثال لتوضيح المسألة:
في البداية سنقوم بتعريف متغير عام على مستوى النموذج بالاسم TestTextكالتالي:

```
Dim TestText As String
```

بعد ذلك نقوم بكتابة الإجراء التالي:

```
Sub Test(ByVal NameTest As String)
        NameTest = "النص بعد التغيير"
    End Sub
```

لاحظ أننا استخدمنا الكلمة ByVal..
بعد ذلك نقوم بتمرير متغير إلى هذا الإجراء ونرى إن كان سيتغير أم لا!

ضع هذا الكود في حدث النقر على الزر مثلاً..

```
TestText = "النص الأصلي"
Test(TestText)
MessageBox.Show(TestText)
```

كما ترى السطر الأول يقوم بملء المتغير TestText الذي عرفناه سابقاً بقيمة معينة..
وفي السطر الثاني نقوم بتمرير هذا المتغير إلى الإجراء.. NameTest
وفي السطر الثالث نقوم بعرض محتوى المتغير TestText لنرى هل تغير أم لا !!
النتيجة طبعاً لا لأننا استخدمنا الكلمة.. ByVal

أعد المثال نفسه مع تغيير الكلمة ByRef بالكلمة ByVal وستجد أن قيمة المتغير قد تغيرت..
أتمنى أن يكون الأمر واضحاً..

ملاحظات:

-**بإمكاننا** أن نقوم بإنشاء إجراءات تحمل نفس الاسم ولكنها تختلف في النوع والعدد وطريقة توزيع المعاملات .. وهذا متاح تماماً..

-
بإمكاننا أن نجعل إحدى المصفوفات كمعامل للإجراء، بحيث يمكننا تمرير عدد غير محدد من القيم إلى هذا الإجراء، ولذلك نستخدم الكلمة ParamArray ..

في الختام أتمنى أن تكونوا قد استفدتم، وفي حال وجود تصحيحات أو استفسارات أرجو وضعها هنا وسنسعد بها قطعاً..

التعامل مع ال Threads

كاتب الموضوع : **Bashmohandes**

هل جربت من قبل أن تتعامل مع ال Threads؟؟؟؟

اذا كانت اجابتك بلا فاقرأ معي هذا الموضوع و اذا كانت إجابتك بنعم فاقرأه أيضاً ففيه فائدة عظيمة ستجدها بين سطور الموضوع

هل ترى التعامل مع ال Threadsصعب ؟؟؟؟

هو ليس صعباً و لكنه فقط مرحلة أخرى متقدمة و لا يجب أن تكون كل مرحلة متقدمة صعبة و العكس صحيح فأحيانا البداية تكون الأصعب على الاطلاق

ما هي ال Threads؟

ال Threadsهي محاولة تجزئة البرنامج الى مجموعة من العمليات المستقلة و التي يمكن أداءها بطريقة متوازية Parallel بدلاً من اضاعة الوقت في انتظار لا طائل منه و هو النظام المتسلسل Serial Execution و يهدف النظام الى تحسين أداء البرامج عموماً فاذا تخيلنا مثلاً برنامجاً رياضياً يقوم بعمل حسابات طويلة و معقدة Matlab) مثلاً) فان البرنامج أثناء تنفيذ هذه العمليات لا يمكنه اداء اي مهمة اخرى حتى ينتهي من الحسابات و من مساوئ هذا هو عدم استجابة البرنامج الى المستخدم (بالبلدي الشاشة بتهنج...)

و لكن هل هذا مانراه فعلاً في البرامج الحقيقية؟؟؟

بالطبع لا فبرنامج Matlab مثلاً يمكنك من تشغيل برامجك و كتابة برامج أخرى و التفاعل مع واجهاته في نفس الوقت بدون أي مشاكل و هذا لأن جميع هذه العمليات هي عمليات مستقلة يمكن التعامل معها من خلال البرنامج بطريقة متوازية

هل ال Threadingمفهوم جديد

ال Threadingبصفة عامة مفهوم معروف و لكن للأسف لم يتم تطبيقه على مستوى لغات البرمجة بل تُرك مفتوحاً لكل نظام يطبقه كما يشاء و يدعم ما يريد و هذا ما جعل ال Threadingفي ال C++مثلاً من المواضيع السيئة فكل نظام له ال API'sالخاصة به و التي تختلف كلياً عن أي نظام اخر...
و لكن مع ظهور الأجيال الحديثة من لغات البرمجة بدأ من ال JAVAو التي حسب معلوماتي أول لغة ظهر ال Threadingفيها كجزء من اللغة نفسها بحيث أنتقلت مشاكل التعامل مع النظام و طريقة تطبيق ال Threadingنفسها الى ال JVMبدلاً من المبرمج المسكين...
و بالطبع مع ظهور ال .net.كان ال Threading جزء لا يتجزأ من ال Frameworkو يمكن استخدامه من أي لغة تتبع ال.. CLR

اذاً كيف نقوم بعمل Threading في ال C#؟؟؟

أولاً هناك Namespace تختص بال Threading و هي System.Threading و هي تحتوي على جميع ال classesالخاصة بال....Threading
اذاً فأول سطر في برنامج يتعامل مع ال Threadingهو

```
using System.Threading
```

أهم Class في ال Threadingبالطبع هي System.Threading.Threadو هي التي توجد فيها أغلب العمليات الأساسية
اذاً هل الخطوة الثانية هي انشاء object من نوع Thread ؟؟
الحقيقة هذه الخطوة تسبقها خطوة قبلها و هي كتابة ال functionالتي ستقوم هذه ال Threadبتنفيذها و لتكن مثلاً

```
void PrintHello()
{
Console.WriteLine("Hello
Threading...");
}
```

ثم نقوم بعمل ال Objectمن نوع Thread

```
Thread thread = new Thread(new
ThreadStart(PrintHello));
```

ثانية واحدة ايه اللي انت كاتبه ده ؟؟؟؟ مين ThreadStartدي و ازاي كاتب اسم ال function كأنه parameter ؟؟؟؟
لو سألت نفسك السؤال السابق اذن فأنت أول مرة تتعامل مع ال Delegatesو هي باختصار شديد بديل محترم لما كنا نستخدمه في ال C/C++و هو ال ... Function Pointer
الحقيقة المجال لا يتسع لشرح ال Delegate هنا و لكن يمكنك أن ترجع لل MSDN... و لو مستعجل كل اللي محتاج تعرفه في السطر اللي فات اني بقول لل Threadبتاعتي ان ال functionاللي حتنفذها اسمها

```
PrintHello
```

شفت خرجتني من الموضوع ازاي؟؟؟ 😠 على العموم ما علينا.

بقيت خطوة أخرى و هي تشغيل ال Thread نفسها باضافة سطر اخر.

```
thread.Start();
```

هي دي ال Threadsبقة و عاملين الهوليلا دي على ال ٣ سطور دول؟

لأ طبعاً دي البداية بس الموضوع طويل و فيه كلام كبير صحصح معايا كده لسة التقيل ورا.

طيب هل هذه هي الطريقة الوحيدة لانشاءThreads

الاجابة بالطبع لا و لكن توجد طرق أخرى كثيرة و هناك أيضاً طرق يتم فيها انشاء Threads بواسطة النظام دون أن يدخل المبرمج في تفاصيل انشاء ال Threadsو هي ما يسمى بال Asynchronous Operationsو ستجد الكثير منها في ال dotnetفمبدئياً أي عملية تبدأ بكلمة BeginXXXو يوجد لها مثيل يبدأ ب EndXXX تعتبر عمليات غير متزامنه و مثال على هذا في ال Streamsستجد BeginRead و EndRead و أيضاً BeginWrite و EndWrite و هذا موضوع اخر مثير في ال dotnetو سأحاول شرحه لاحقاً
و يوجد أيضاً طريقة أخرى لانشاء ال Threadsو تعتمد على فكرة أن انشاء Thread جديدة و تشغيلها و مايتطلبه هذا من بعض العمليات الأخرى أحياناً يكون وقت هذه العمليات أكبر من العملية التي ستنفذها ال Threadفي النهاية مما يؤثر سلباً على الأداء و هنا ظهر مفهوم ال ThreadPoolأي مستودع ال Threadsو هي مجموعة من ال Threadsالمنشأة (بضم الميم) مسبقاً و هي جاهزة و تحتاج فقط للعملية التي ستقوم بها و يكون لها عدد محدد و يمكن للمستخدم أن يستخدمها ببساطة جداً
طريقة استخدامها تكون من ال ThreadPool Classعن طريق وضع ال functionالمُراد تنفيذها في ال Queueحتى تفرغ احدى ال Threads لتنفيذ هذه العملية و يكون هكذا.

```
ThreadPool.QueueUserWorkItem(new
WaitCallback(PrintHello));
```

و هكذا سيتم وضع العملية في ال Queueو سيكمل البرنامج عمله دون انتظار و عندما تكون هناك Thread جاهزة سيتم التنفيذ.

كيفية التعامل مع الـ Registry, من خلال مكتبة الـWindwos Scripting Host

كاتب الموضوع : **عبد الله فتحي**

في هذا الموضوع سنشرح طريقة أخرى للتعامل مع الريجستري وهي طريقة سهلة جداً سنعتمد فيها على استخدام الكائن.. Wscript

الأمر لا يتعدى ثلاثة أسطر من الكود..
في البداية لابد أن نقوم بتعريف الكائن بواسطة الكود التالي:

```
Dim WSH As Object
Set WSH = CreateObject("Wscript.Shell")
```

بعدها سيكون تعاملنا مع الريجستري إما بالإضافة أو الحذف أو استرجاع القيم، وكل عملية من هذه العمليات لا تحتاج إلى أكثر من سطر كودي واحد لعملها .. ولنر ذلك معاً:

لإضافة مدخل قيمة جديد إلى الريجستري:

```
WSH.RegWrite "HKCU\ArabTeam\Example", 1,
"REG_DWORD"
```

لاحظ أننا استخدمنا الوظيفة RegWrite وقمنا بتمرير ثلاث قيم إلى هذه الوظيفة، وهي:
1. اسم مدخل القيمة الجديد الذي نريد إضافته ومساره.
2. قيمة هذا المدخل الجديد.
3. نوع هذا المدخل ولاحظ أنه هنا من نوع REG_DWORD ، ويمكن أن يكون REG_STRINGأو REG_BINARY أو REG_MULTI_SZ أو أي نوع تريده..

ولإضافة مفتاح سنستخدم نفس الطريقة السابقة، مع تغيير طفيف، لاحظ:

```
WSH.RegWrite "HKCU\ArabTeam\Codes\", 1,
"REG_DWORD"
```

لاحظ أننا سنمرر اسم المدخل الذي نريد حذفه إلى الوظيفة.. RegDelete

ولحذف مفتاح موجود مسبقاً نستخدم نفس الطريقة السابقة مع تغيير

بسيط، لاحظ:

```
WSH.RegWrite "HKCU\ArabTeam\Codes\"
```

ولمعرفة قيمة مدخل موجود بالريجستري:

```
Print WSH.RegRead("HKCU\Al-ebda3.info\Example")
```

ولمعرفة قيمة مفتاح موجود بالريجستري:

```
Print WSH.RegRead("HKCU\ArabTeam\Codes\")
```

أتمنى أن أكون قد وُفقت في شرحي لذلك .. وأتمنى للجميع التوفيق

Inline Directive

كاتب الموضوع : **Waeldalol** (موضوع مترجم بتصرف من كتاب **INSIDE DELPHI 2006**)

ما هي Inline directive:

مقدمة:

دائما كان يجري نقاش بيني و بين المبرمجين الاخرين حول السؤال التالي:

هل كتابة الكود في Block واحدة اي في Function واحد مثل Main function هي افضل من كتابة الكود بشكل مجزء و مقسم الى عدة توابع؟

و في الاخير اتفقنا انه عند كتابة الكود في Block واحدة هي اسرع بكثير من كتابة الكود مجزء الى توابع و اجرائيات و لكن من الاسباب التي لم تجعلنا نستخدم هذه الطريقة هي ان حجم الكود الناتج هو اكبر بكثير من الحجم الذي قد ينجم عند كتابة توابع و اجرائيات و انه ايضا غير منظم و صعب القراءة و غير قابل لاعادة الاستخدام الخ الخ...

و لكن لماذا اسرع؟ لم نعرف السبب!! حاول ان تفكر في الجواب قليلا

و طبعا لم نكتب اي برنامج نستخدم فيه هذه الطريقة و مع صدور دلفي ٢٠٠٥

ظهر مصطلح جديد الا و هو Inline function و الذي يعطي تحكم بكيفية ترجمة(Compile) الاجرائيات او التوابع(الدوال) بحيث يمكنك من تسريع عمل التوابع الصغيرة و التي يتم استخدامها بكثرة في الكود حيث ممكن ان تعتبر انّ الاجرائيات او التوابع Inline هي اسرع من التوابع التي ليست Inline اي التوابع العادية و لكن هناك سيئة واحدة لاستخدام توابع او اجرائيات Inline الا و هي زيادة في حجم الملف التنفيذي اي ملف Exe الناتج عن تنفيذ البرنامج.

على ما اعتقد انّ Inline function قد وجدت منذ زمن بعيد في C++ و ها هي اليوم في دلفي.

الان كيف نقوم بتعريف تابع او اجرائية بحيث تكون Inline:

ما عليك الا وضع الكلمة المحجوزة Inline في نهاية تعريف الاجرائية كأنك تستخدم Virtual او override..

و ذلك كما يلي:

```
procedure MyProc(x:Integer);
inline;
begin
    // ...
end;

function MyFunc(y:Char) :
String; inline;
```

```
begin
   // ..
end;
```

ملاحظة: سوف استخدم الكلمة Method للدلالة على التوابع و الاجرائيات

حتى الان لا يوجد اي طريقة لمعرفة انّ التابع سوف يتم تنفيذه Inline او سوف يتم تنفيذه بشكل عادي و لكن هناك بعض الشروط التي لا يمكن استخدام Inline معها مثل ما يلي:

- يجب ان لا يحتوي التابع على assembly code و الا سوف يتم تنفيذه عادي

- تابع البناء(Constructors) و تابع الهدم(destructors) لا يمكن ان يكونا Inline ابدا.

- الجزء الرئيسي من البرنامج و الذي يبدأ ب Begin و ينتهي ب End و الذي يتم بدأ تنفيذ البرنامج من عنده لا يمكن ان يكون Inline كذلك unit initialization و ال unit finalization.

- Method لم يتم تعريفها مسبقا لا يمكن استخدامها ك Inline.

- Method التي تأخذ في بارمتراتها مصفوفة مفتوحة لا يمكن ان تكون Inline.

و هناك العديد من الامور الاخرى التي يمكنكم الاطلاع عليها في Help المرفق مع اللغة.

كيف تعمل Inline functions: اي Method معرف Inline:

نحن نعلم انه عند استدعاء اي Method يتم الحجز لجميع المتحولات المعرفة ضمن هذا ال Method و يتم الحجز ايضا لجميع المتحولات المعرفة في Parameters.

هل تعلمون يا اصدقائي اين يتم هذا الحجز؟ نعم نعرف يتم في Stack اي المكدس و لكن هل تعلمون ما هو المكدس؟ نعم نعرف هو عبارة عن قطعة او قسم من الذاكرة يتم استخدامها من قبل التطبيق عند التعامل مع متحولات المحلية اوعندما تنادي على إجرائية، يتم وضع محدّداتها و نوع ارجاعها في Stack .

جواب اخر: ال Stack هو واحد من ثلاث مناطق للذاكرة متوفرة للتطبيق. المنطقتان الأخريان تسمّيان ذاكرة جامعة global memory و ركام . Heap

يتم استخدامه لحجز المتحولات المحلية و لتخزين محددات اجرائية و قيمها المرجعة.

عندما ينتهي تنفيذ الاجرائية يتم بشكل تلقائي الغاء حجز المتحولات و البارمترات من الذاكرة هذا هو السبب الذي لا يجعلك تتمكن من استخدام المتحولات المحلية المعرفة ضمن الاجرائية و البارمترات خارج هذه الاجرائية.

يمكن لتطبيقات ويندوز أن تحتفظ بكمية ضخمة من البيانات في ذاكرة ال Stack حيث انه في في دلفي يمكنك تحديد هذا الأمر في صفحة الرابط Linker من خيارات المشروع. اذا ما حصلت على رسالة خطأ لإمتلاء الصفّ stack fullفالأغلب ان هذا سببه أن لديك وظيفة تنادي نفسها تكرارا و بدون توقف .وليس بسبب محدودية فضاء ال Stack.

لن اتحدث كثير عن ال Stack و انما سوف ابقى في Inline Function و كيف تعمل!؟

ولكن يجب علينا الغوص قليلا في بعض الامور و التي تعتبر Low level و عادة اغلب المطورين لا يعيرون لها انتباههم و ذلك لوجود صعوبة بعض الشيئ و لانهم ليسوا بحاجة اليها في بناء تطبيقاتهم و لا لمعرفة كيف تعمل. و لكن نريد ان نفهم كيف تعمل Inline function و لماذا هي اسرع من Function العادية ما السر الذي تخفيه وراءها Inline function حتى يكون تنفيذها اسرع من تنفيذ ال function العادية و لماذا حجم الملف التنفيذي يكبر عند استخدامها؟

دعونا نبدأ رويدا رويدا خطوة بخطوة!

عندما يتم عمل Compile لاجرائية او تابع(Method) ماذا يحدث؟

سوف يتم ضم كل من لغة الالة التي تمثل تعليمات ال Method و كود الدخول و الخروج من ال Method الى الملف التنفيذي.

ايضا سوف يتولد لكل استدعاء ل Method تعليمات لغة الة تقوم بتمرير البامترات و تعليمات اخرى تقوم باستدعاء ال Method.

دعونا نلقي نظرة على بعض تعليمات لغة الالة:

ليكن لدينا التابع MyAbs و الذي يعيد القيمة المطلقة لاي رقم ممرر له حيث انّ كود دلفي لعمل هذا التابع و لاستدعاءه يكون على الشكل التالي

```
Function MyAbs(I: Integer): Integer;
begin
  if I < 0  then
    Result := -I
  else
    Result := I;
end;

begin
  x := MyAbs(-20);
end.
```

الان لاستدعاء تابع Myabs في القسم الرئيسي من البرنامج يقوم دلفي بتوليد كود لغة الالة التالي

```
mov eax,$fffffffec
call MyAbs
mov [$0040565c],eax
```

ما هذا طلاسم؟ خربيش جاج؟!! لا انه كود اسمبلي كود الالة الذي كنا نتحدث عنه

التعليمة الاولى تقوم بوضع الرقم -٢٠ في المسجل eax اي ال Stack

التعليمة الثانية تقوم باستدعاء التابع Myaps

التعليمة الثالثة و الاخيرة تقوم بوضع النتيجة الراجعة من التابع بالمتحول X حيث انّ $0040565c هي عنوان المتحول x في الذاكرة

دعونا الان نرى كود الالة الذي يولده دلفي للتابع MyAps

```
// begin
push ebp
mov ebp,esp
add esp,-$08
mov [ebp-$04],eax

// if I < 0  then
cmp dword ptr [ebp-$04],$00
jnl $00403a19

// Result := -I
mov eax,[ebp-$04]
neg eax
mov [ebp-$08],eax
jmp $00403a1f

// Result := I;
mov eax,[ebp-$04]
mov [ebp-$08],eax
mov eax,[ebp-$08]

// end;
pop ecx
pop ecx
pop ebp
ret
```

القسم الاول:

```
// begin
push ebp
mov ebp,esp
add esp,-$08
mov [ebp-$04],eax
```

التعليمتين الاولتين فقط يتم حجز المتحولات اللازمة للاستخدام فقط هكذا لا اريد ان اتوسع في هذا الامر

التعليمة الثالثة تحجز ٨ بايت من ذاكرة ال Stack و هي مساحة كافية لحفظ قيمتين من النوع Integer.

التعليمة الرابعة تقوم بنقل القيمة من البارميتر الى ال Stack

القسم الثاني:

```
// if I < 0 then
cmp dword ptr [ebp-$04],$00
jnl $00403a19
```

هنا يتم فحص فيماذا كانت قيمة I تحتاج لكي تحول الى قيمة موجبة ام لا.

التعليمة الاولى: تقوم بالمقارنة بين البارميتر I الموجود في ال Stack مع قيمة الصفر ٠

التعليمة الثانية: اذا كانت قيمة البارميتر I هي اكبر من صفر ٠ اي انّ القيمة الموجودة في المتحول I لا تحتاج الى تحويل هنا يتم القفز الى القسم الثالث. و من ثم يتم اعادة القيمة الى الخرج.

القسم الثالث:

```
// Result := I;
mov eax,[ebp-$04]
mov [ebp-$08],eax
```

```
mov eax,[ebp-$08]
```

القسم الرابع:

في حال كانت قيمة I هي قيمة سالبة سوف يتم هنا عكس اشارة I و من ثم يتم اعادة القيمة الى الخرج.

```
// Result := -I
mov eax,[ebp-$04]
neg eax
mov [ebp-$08],eax
jmp $00403a1f
mov eax,[ebp-$08]
```

القسم الخامس

```
// end;
pop ecx
pop ecx
pop ebp
ret
```

هنا يتم الغاء حجز المتحولات التي تم حجزها في البادية حيث باستدعاء هذه التعليمات يتم ارجاع ال Stack الى الوضع الذي كان عليه قبل استدعاء التابع كاملا.

اخر تعليمة و هي ret تعيد التحكم الى المستدعي لكي يتم اكمال التنفيذ.

الان دعونا ننظر كيف سوف يتعامل دلفي مع الكود السابق في حال وجود كلمة Inline بجانب تعريف التابع:

في الحقيقة عند استخدام Inline تختلف طريقة اصدار دلفي للكود كليا عن الطريقة السابقة حيث انه عند استخدام كلمة Inline لا يستدعي دلفي التابع بل يقوم بنسخ التابع ووضعه عند كل استدعاء للتابع اي عمل عدة نسخ ووضع كل منها في مكان الاستدعاء طبعا عدة نسخ اسمبلي اي كود الالة.

بهذه الطريقة لا يقوم ال Compiler بتوليد كود لا لتعريف البارمترات اي القسم الاول الذي تحدثنا عنه و لا يقوم ايضا بارجاع القيم اي لا يقوم بتوليد الكود الذي في القسم الاخير.

عند استخدام ال Inline functions في التوابع الصغيرة سوف تلاحظ زيادة كبيرة في سرعة التنفيذ و بشكل خاص عند استخدام التابع في الحلقات.

الان نسخة من التابع السابق و لكن هذه المرة مع وجود كلمة Inline اي استخدام Inline function:

```
function MyAbs(I: Integer):
Integer; inline;
begin
  if I < 0  then
    Result := -I
  else
```

```
    Result := I;
end;
```

عند استخدام توابع Inline لا يتم توليد اي كود اضافي حيث يتم فقط توليد التعليمات لتنفيذ المطلوب دون ادخال اي شيئ في ال Stack او استرجاع اي شيئ منه.

هذا هو كود الالة الذي يتم وضعه في حال استخدمت Inline function

```
mov eax,$fffffec
test eax,eax
jnl $00403a6c
neg eax
mov ebx,eax
```

الان وضح كل شيئ وانكشف الغطاء وعرفنا ما هو سبب سرعة Inline function عن ال function العادي؟ بسبب نقص عدد التعليمات اللازمة لتنفيذ المطلوب ، اذا اعيد و اكرر حتى توضح الفكرة اكثر عند كتابة Inline function كل التعليمات التي تؤلف هذا التابع سوف يتم نسخها الى كل مكان استدعاء لهذا التابع مما يؤدي الى حجم كبير في الملف التنفيذ النهائي و ذلك بسبب تكرار الكود و سوف يؤدي ايضا الى زيادة كبيرة في السرعة بسبب انّ حجم التعليمات الازمة لتنفيذ الامر بدون استدعاء تكون اقل من التعليمات اللازمة لتنفيذ الامر مع استدعاء.

فكرة مفيدة و هامة للجميع:

قد يقول البعض انّ Inline function اتت مع نسخة دلفي ٢٠٠٥ اي لم تكن موجودة في النسخ السابقة دلفي٧ و ما سبق و بالتالي فإن ال Compiler للدلفي ٧ و ما سبق لا يستطيع ترجمة Inline function ؟!!!

ماذا لو اردت ان اترجم برنامجي على نسخة دلفي٧ ماذا يحصل؟!! اكيد لن تتمكن من ترجمته

فما رأيكم بماذا نرد على هذا المبرمج الفذ و الذي يفكر للبعيد و يحسب حساب كل شيئ؟

ما رأيكم هل من جواب فكر قليلا بالامر!!؟

الجواب هو:

في دلفي هناك شيئ يسمى conditional compilation اي الترجمة المشروطة اي استطيع ان افحص شيئ ما اذا لم يكن مثلما اريد فلا تتم الترجمة.

مثال: اذا كتب الجملة التالية:

```
{$IFDEF LINUX}
WriteLn('Linux');
{$ENDIF}
```

هذه الجملة التي كتبناها لن يتم تنفيذها الى اذا كنت تنفذ برنامجك على Linux

$IFDEF نضع بعدها ما نريد ان نفحصه و نختم بعد ذلك $ENDIF

الان نحن نريد ان يكون التابع Inline في حال كنا نعمل على نسخة دلفي ٢٠٠٥ او ٢٠٠٦ و ان يتم تنفيذ التابع بشكل عادي في حال كانت النسخة هي دلفي٧ او ما سبق الان لفحص رقم النسخة نستخدم ما يلي

```
{$IFDEF VER170}

// Delphi 2005 specific code

{$ENDIF}
```

حيث انّ VER170 تدل على انّ النسخة المستخدمة هي دلفي ٢٠٠٥ و VER180 تدل على انّ النسخة المستخدمة هي دلفي ٢٠٠٦.

الان في برنامجنا كيف نطبق هذا:

```
function MyAbs(I: Integer):
Integer; {$IFDEF VER170} inline;
{$ENDIF}
```

لاحظوا وضعنا الكلمة Inline بين العبارتين {$IFDEF VER170} و {$ENDIF} اي فقط في حال كانت نسخة ال compiler هي دلفي ٢٠٠٥ يتم وضع التابع Inline و الا يبقى التابع عادي.

ملاحظة:

VER170:رقم نسخة دلفي ٢٠٠٥

VER180:رقم نسخة دلفي ٢٠٠٦

الان النسخ من ١ الى ٧ محجوزة لنسخ Turbo Pascal اي VER1 هي نسخة Compiler ل Turbo Pascal

النسخ من ٨ الى ١٥ هي نسخ دلفي من ١ الى دلفي ٧

و النسخة ١٦ هي دلفي ٨

امل ان اكون قد افدتكم بقليل من المعلومات

Miracles

كاتب الموضوع **Romanof** (موضوع مترجم من موقع http://www.citforum.ru/programming/delphi/miracles/)

هل تؤمن بالمعجزات ؟ انا ابضا اؤمن بالمعجزات ..بالتاكيد كل منا سيسلم ان شيئا من هذا القبيل يحدث مع اكواد برماجنا بل واحيانا لايرفض البرنامج عملcompile له !؟ او ان البرنامج يظهر نتائج غيرمتوقعة..وهنا تراودك افكار عن مشاركة قوى خفيه في صنع هذه المعجزات

في هذا المقال سوف نتطرق الى ٧ (من العديد) من تلك المعجزات وسوف نحاول ان نزيل الغطاء السحري ونتاكد ان هذا ما هو الا خدعة...(Illusion) وهم ..

معجزة التقريب (Round Miracle)

انشي مشروعا جديدا سمه مثلا AllMiracles ضع مكونا من النوع button على الفورم.. الان بالضغط مرتين على الزر اكتب مايأتي

```
procedure TForm1.Button1Click(Sender: TObject);
begin
ShowMessage( IntToStr( Round(3.5) - Round(2.5) )
);
end;
```

توقف قليلا وقل لي ما هي النتيجة المتوقعة ؟؟اتمنى ان لا تكون اجابتك بواحد والا لم تكن معجزة ..واذا كان حدسك ممتازا فستكون اجابتك • وهذا ابعد عن الحقيقة مقارنة عن ذي قبل واذا كنت مقامرا جيدا وتقرا الهلب جيدا فكنت ستجيب بالرقم ٢!!

لاتصدق.. تضغط F9

هل تعرف لماذا؟؟؟

لو كنت قرات الهلب جيدا فانظر مالمكتوب عن الداله (التابع) Round :

Round returns an Int64 value that is the value of X rounded to the nearest whole number. If X is exactly halfway between two whole numbers, the result is always the even number.

هل عرفت السبب !!! p:

الان اتمنى ان تملك تصورا عن ما اتحدث عنه اليوم ... في هذا المقال لايوجد ما صعب تعمدنا تبسيط الكود حتى تتضح المشكلة

(Absolute Miracle)

الان اضف زرا اخر الى الفورم .. الان بالضغط مرتين على الزر اكتب ماياتي

```
procedure TForm1.Button2Click(Sender: TObject);
var
i1: int64;
begin
  i1:= abs(low(integer));
  ShowMessage(IntToStr(i1));
end;
```

قبل ان تقوم بالضغط على F9 سنقوم بتحليل الموقف

من المعروف ان Low مع ال integer تعطي القيمة -2147483648 (انظر الهلب) كما ان الهلب لا تعطي معلومة جديدة عن الabs

Abs returns the absolute value of the argument X. X is an integer-type or real-type expression.

المتغير i1 عبارة عن int64 ..طبعا هذا صحيح لان 2147483648 تخرج عن حدود ال integer وهذه القيمة هي القيمة التي ننتظرها اليس كذلك . اضغط F9 وتاكد !!! ماذا ..غير معقول ان تعيد الداله abs عددا سالبا ..فكيف حدث ذلك...

تعالوا نتامل الموضوع مره اخري سوف نرى العباره abs(low(integer)) .ماذا نستطيع ان نقول عن هذه العبارة رغم استدعاء دوال في هذه العبارة الا انها في الحقيقة قيمة ثابته

ماذا يقول الهلب الثوابت Constant expressions :

...Constant expressions cannot include variables, pointers, or function calls, except calls to the following predefined functions: Abs...Low...

سوف نعرف ثابتا لهذه الطريقة :

```
const
ci = abs(low(integer));
...
```

هل تمكنت من تشغيل برنامج يحتوي على هذه العبارة ؟؟ على ماذا يدل هذا؟ على انها قيمة ثابته ؟ من كان يصدق هذا!!

ولكن لم لا تعطينا العبارة abs(low(integer)) عددا سالبا وكيف السبيل لفعل العكس

انظر المثال التالي

```
procedure TForm1.Button3Click(Sender: TObject);
const
ci = abs(low(integer));
var
i1: int64;
begin
```

```
i1:= abs(int64(low(integer)));
ShowMessage(IntToStr(i1));
end;
```

جرب هل رايت ؟ حصلنا على العكس وعرفنا السر ! :p

معناها ان العدد abs(low(integer)) خارج نطاق integer ويدخل نطاق int64

يالمناسبه ايضا العبارة ;abs(int64(low(integer))) تعتبر قيمة ثابته .

(One more low integer miracle).

```
procedure TfrmAllMiracles.btnLowIntMrclClick(
Sender: TObject);
var
lowInt: integer;
begin
lowInt := -2147483648;
ShowMessageFmt('%d',[lowInt]);
end;
```

مالمشكلة اجراء عادي جدا والقيمة قانونيه جدا؟؟ ومع ذلك لا نستطيع اجراء compile بحجة ان

Overflow in conversion or arithmetic operation

اضغط على F1 ماذا تلاحظ :

The compiler has detected an overflow in an arithmetic expression: the result of the expression is too large to be represented in 32 bits

كيف overflow اذا كانت القيمة قانونية ؟ اذا لابد ان المترجم يقوم بانشاء قيمة موجبة ةعد ذلك يضربها في -١ p:

لكن كيف الخروج من هذه المحنه انظر الكود:

```
procedure TForm1.Button4Click(Sender: TObject);
var
lowInt: integer;
begin
lowInt := -int64(2147483648);
ShowMessageFmt('%d',[lowInt]);
end;
```

ربما كان المثال ساذجا لكنه يعطي تصورا حقيقيا لما يفعلة مترجم الدلفي لانشاء الثوابت ومنح القيم !

(String Trick)

سيدي العزيز انظر الى هذا المثال طبعا اضف زر الى الفورم بالضغط مرتين على هذا الزر

واكتب مایاتي

```
procedure TfrmAllMiracles.btnCopyMrclClick
(Sender: TObject);
const
cs: array[0..1] of char='01';
begin
ShowMessage(copy(cs,0,1)+copy(cs,1,1));
end;
```

ما هي الاجابه بالنظر الى الكود السابق؟

هه خطا طبعا . الاجابه هي "٠٠" هل تعرف لماذا عد الى الهلب وانظر الى الى تعريف الcopy

```
function Copy(S; Index, Count: Integer): string;
function Copy(S; Index, Count: Integer): array;
```

اعتقد ان الامر مفهوم الان...

في المره الاولي تم استدعاء copy لمعالجة (arry)مصفوفه ديناميكة تبدا بالعنصر ذو العنوان صفرا!

في المراة الثانية تم استدعاء الداله وكانها string يبدا من العنصر ذو العنوان صفرا!

(Is-Miracle) اكتب في الجزء protected من الفورم الحقل FControl من النوع TControl

والان ضع في على الفورم زرا جديدا وفي الحدث onClick اكتب الكود

```
procedure TfrmAllMiracles.btnIsMrclClick(Sender:
TObject);
begin
if (FControl is TControl) then
begin
if not Assigned(FControl) then
FControl := TControl.Create(Self);
end
else
ShowMessage('Not a Control');
end;
```

قم بعمل لاحظ انك مهما قمت بعمل compile فانك لن ترى غير الرساله 'Not a Control' والباني للصنف TControl

لن يستدعى ابدا!!

ماذا يقول الهلب لنا في هذا الامر

The expression object is class returns True if object is an instance of the class denoted by class or one of its descendants, and False otherwise. (If object is nil, the result is False.)

هل فهمت سر الموضوع ؟ الير يكمن في كون is كانت تتعامل مع متغير في الحقيقة لم يخصص لها اي ذاكرة يعني في الحقيقة ما هي الا مؤشر لا يشير الى شيء (null)

انوه الى كون اختيار الصنف Tcontrol كان اختياريا كان من الممكن ام يحصل نفس الشي مع اي صنف اخر سوف نرى شيئا مشابها في المثال القادم

(Is-Miracle II)

كون مكتبه dll وسمها يالاسم AllMirrLib

```
library AllMirrLib;
uses
Controls;

function IsControlLib(const anObj: TObject):
boolean;
begin
Result := anObj is TControl;
end;

exports
IsControlLib;
end;
```

قم بعمل نفس الخطوات في المثال الماضي مع الحقل FControl
ثم اضف مكون Button وفي الحدث onClick واكتب الاتي

```
procedure TForm1.Button2Click(Sender: TObject);
begin
   FControl := TControl.Create(nil);
    try
     if not IsControlLib(FControl) then
       ShowMessage('Not a Control');
     finally
       FreeAndNil(FControl);
   end;
end;
```

اضغط F9 ؟

ظهور الرساله مرة اخري ؟؟ معني هذا ان FControl =Null رغم اننا استدعينا الباني قبلا فما المشكلة اذن

سوف نعدل الكود في مكتبة ال dll الى الكود التالي

```
function IsControlLib(const anObj: TObject): boolean;
var p3,p4: pointer;
begin

  p3 := pointer(anObj.ClassType);
  p4 := pointer(TControl);
  Result := (p3=p4);
end;
```

لا تنسى ان تعمل Build لمكتبة الdll ثم عد الى البرنامج الاصلي

بينما في برنا مجنا الاصلي نعدل كالاتي في الحدث onClick من الزر اكتب

```
procedure TForm1.Button1Click(Sender: TObject);
var
p1, p2: pointer;
begin
   FControl := TControl.Create(nil);
   try
   p1 := pointer(FControl.ClassType);
   p2 := pointer(TControl);
      if (p1=p2)then showmessage('yes')
       else showmessage('No');
      if  IsControlLib(FControl) then
        ShowMessage('yes dll')
        else  ShowMessage('No dll')
        finally
      FreeAndNil(FControl);
   end;
end;
```

قم بالتشغيل والان اضغط الزر ماذا تلاحظ p1, p2 متساويان في البرنامج لذلك ترى الرساله yes

بينما ظهور ال NO dll في الرساله الثانية يدل على عدم تساوي هاتين القيمتين في Dll !!!

رغم انهما يشيران الى نفس الشي اين ترى اشارة الى هذا الشي في الهلب ؟؟

Use ClassNameIs when writing conditional code based on an object's type or to query objects across modules, or DLLs.

هذا في الحديث عن المنهج (method) ClassNameIs

واضح انه في ال dll نوعان من الTcontrol !!

(Variant trick)

ماذا يقول الهلب عن Variants in expressions

...In a binary operation, if only one operand is a variant, the other is converted to a variant..

لايبدوا هذا مثير انك تستطيع ان تجمع كل شي مع الvariant ناتج الجاع يكون variant دائما

اتظر الى هذا الكود

```
procedure TForm1.Button3Click(Sender: TObject);
var
v: variant;
b: boolean;
i: integer;
s: string;
d: TDatetime;
x: Double;
begin
  v:=0;
  b := true;
  i := 2;
  s := '3';
  d := StrToDateTime('01.01.01');
  x := 5;
  v := v+i+s+d+x+b;
  ShowMessage(VarToStr(v));
end;
```

كيف ممكن ان تجمع كل هذه الاشياء ؟

الاغرب ان هذه الكود لا يكتفي بالعمل وبل يعطي ناتجا ايضا :p

Regions

كاتب الموضوع : **Xacker**

بسم الله الرحمن الرحيم

في هذا المقال سوف أتناول في حديثي طريقة تشكيل واجهات للتطبيقات البرمجية التي تملك أشكالاً غير هندسية.

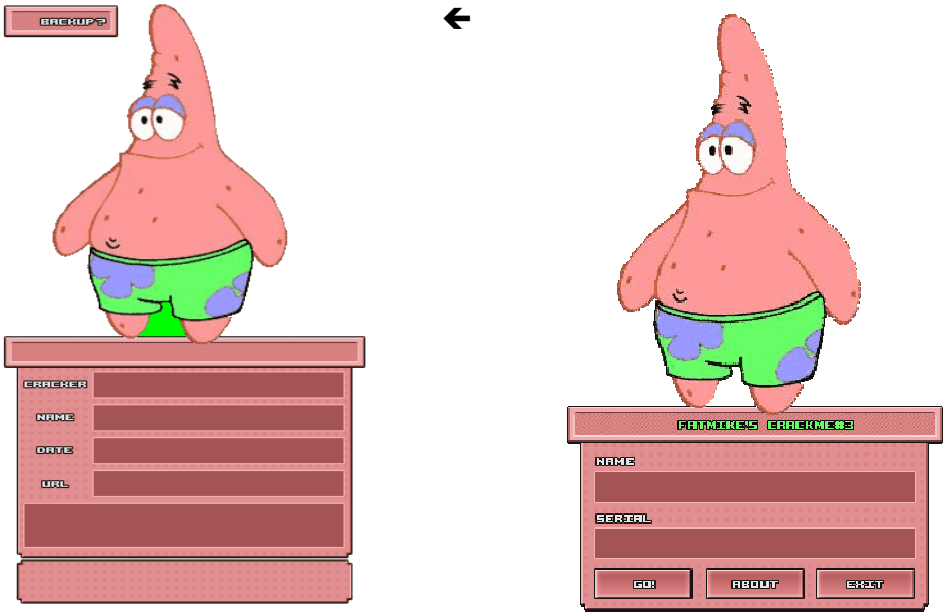
للتذكير بشكل هذه التطبيقات فلا بد أنك تذكر الـ Skins التي يقدمها برنامج Windows Media Players بإصداراته المختلفة، ولا بد بأنك قمت بالتنقل بين مختلف الأشكال التي من الممكن استخدامها كواجهة للبرنامج..

المهم، يتبادر السؤال للعديد من المبرمجين ذوي الخبرة القليلة في التعامل مع الرسوميات عن كيفية القيام بهذا أو ذاك أو استثمار مكتبة GDI32.dll بالشكل الأمثل، لن أتناول في هذا المقال بالتحديد بعض أشكال التعامل مع مكتبة GDI32 وسأدع ذلك لوقت لاحق.

لأختصر الوقت دعوني ألخص الأفكار بالنقاط التالية:

- أريد أن أقوم بكتابة تطبيق (برنامج قياسي، patch، keygen ☺ ...) لكني لا أريد له أن يملك واجهة تقليدية مللنا منها، مثلاً أريده أن يظهر بشكل دائري أو أي شكل آخر غير هندسي لا على التعيين.. أو أريد مثلاً أن أستخدم خلفية شفافة له بحيث تبدو خلفيته بخلفية الصورة المستخدمة وليست كواجهة تقليدية.

على سبيل المثال الصورة التالية تظهر أحد التطبيقات التي كنت قد شكلتها من قبل، الصورة الأصلية مأخوذة من unpack.me وقمت بإجراء تعديلات عليها لاحقاً ببرنامج Adobe PhotoShop CS لتناسب احتياجاتي.



ما سيتساءله الجميع الآن، هل تملك هذه الصور خلفيات بيضاء اللون لا تظهر لأنها مدرجة على خلفية بيضاء.. أم أنها بحق شفافة؟ والجواب بسيط..

كما ترون فإن خلفية سطح المكتب لدي تظهر في الخلفية.. لو أردت أن أقوم بسحب النافذة إلى موضع آخر فإنه سيتوجب علي أن أقوم بالنقر على منطقة مرئية من النافذة لأن بقية الأجزاء الأخرى الموجودة ضمن إطارها (بين القدمين، تحت لافتة ?Backup، إلى يمين ويسار.. محاذاة أي حد من حدود الجدول)، جميع هذه الأماكن فعلياً الآن لا تعود لنافذة البرنامج وإنما الى سطح المكتب نفسه!

▪ نعود الآن إلى النقاط الأساسية من محور الحديث والسؤال الآن، أي اللغات هي لغات عملية للتعامل بهذه الأمور.. ولكوني من مبرمجي الأسمبلي ومشجعي الـ C أستطيع القول بأن هاتين اللغتين توفران القاعدة الأمتن لبناء مثل هذه التطبيقات.. أما لو أردت أن تتعامل بهذا بأي لغة أخرى (VB?) فعليك أن تكون ملماً إلماماً جيداً بكيفية استخدام الدوال البرمجية التابعة لمكاتب الربط الخاصة بالنظام بشكل جيد، من ما زال لا يعلم كيف يقوم بإنهاء برنامج بطريقة غير النقر على [x] فلينس الأمر مبدئياً!

▪ وما هي الأدوات المطلوبة؟

a. لغة برمجة (في هذا المقال سأستخدم الأسمبلي.. طبعاً).

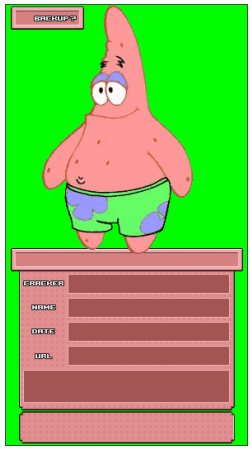
b. برنامج RGN Generator المكتوب من قبل Richard de Oude.

c. صورة بخلفية ذات لون أحادي بتنسيق PNG.

قبل أن نتابع، سيخطر للبعض سؤال حول الحاجة إلى صورة ذات خلفية بلون أحادي، ما المقصود بلون أحادي بأي حال!؟

في الواقع أنا لست خبير تصميم، لذا لا أعلم إن كان لمثل هذا النوع من الصور أو الألوان اسم معين وما أعنيه باللون الأحادي هو أن تملك الصورة خلفية بلون واحد، مثلاً أن تكون خلفية الصورة بلون أسود ذي نفس الدرجة اللونية RGB على سبيل المثال.. أو أي لون آخر بحيث لا يكون هناك أي تداخل بين الألوان، وإلا ستكون مسألة الـ regioning معقدة جداً أنا نفسي لن أجد حلاً لها إلا بعد الكثير من الـ trail & errors.

هل كانت الصورة في الأعلى تملك لون واحد؟ نعم



من السهولة التعامل مع مثل هذه الصورة، لأن مهمتنا ستكون مختصرة في تفريغ اللون ذاك أثناء عملية تشكيل الواجهة بحيث لا يتم عمل تحديث لذلك الجزء مما يبقى على الخلفية الأصلية للشاشة دون ذلك الجزء ذي اللون الموّحد.

من الممكن أن نبدأ الآن لكن علي أن أحذرك بأنه سيكون هناك

الكثير من الـ.. أسمبلي

- ملفات المشروع المرفق واستخداماتها:

ملف regions.inc:

سيحتوي هذا الملف على التصاريح ومسارات المكاتب التي نريد التعامل معها، بما اننا سنقوم بالتعامل مع الرسوميات الان فعلينا ان نضيف الى مجموعة المكتبات التي تعودنا عليها مكتبة gdi32.dll والتي يتم التصريح عنها بواسطة السطرين:

```
include gdi32.inc
includelib gdi32.lib
```

ايضا سنحتاج الى اضافة مكتبة اخرى خارجية سنقوم باستخدامها هي pnglib.dll من خلال ملفيها:

```
include pnglib.inc
includelib pnglib.lib
```

سيحتوي الملف ايضا على تصريح لوظيفة DlgPrc التي تعرفنا عليها من قبل، اضافة الى تصريح اخر لوظيفة سنقوم باستخدامها من اجل التعامل مع مكتبة الصور هي PNG_GetResource والتي سنصرح عنها بالشكل:

```
PNG_GetResource proto :DWORD, :DWORD, :DWORD
```

والتي كما يبدو تملك ٣ بارامترات.

بقية المعلومات الموجودة ضمن هذا الملف يمكن الاطلاع عليها من خلاله حيث تملك تعليقات توضح اهميتها.

ملف regions.rc:

هذا الملف يحوي على تصريحات لاستخدام ٣ ملفات rc (ملفات rc هي ملفات خاصة بالـ resource قبل تحويلها الى res)

اهم هذه الملفات هي:

GdiAppRes.rc ويحوي معلومات لاستخدام ايقونة للبرنامج مع تعريفها.

template.rc ويحوي معلومات لاستخدام الصورة المراد اختيارها للخلفية وملف الـ rgn اللازم لازالة اللون المطلوب من الصورة.

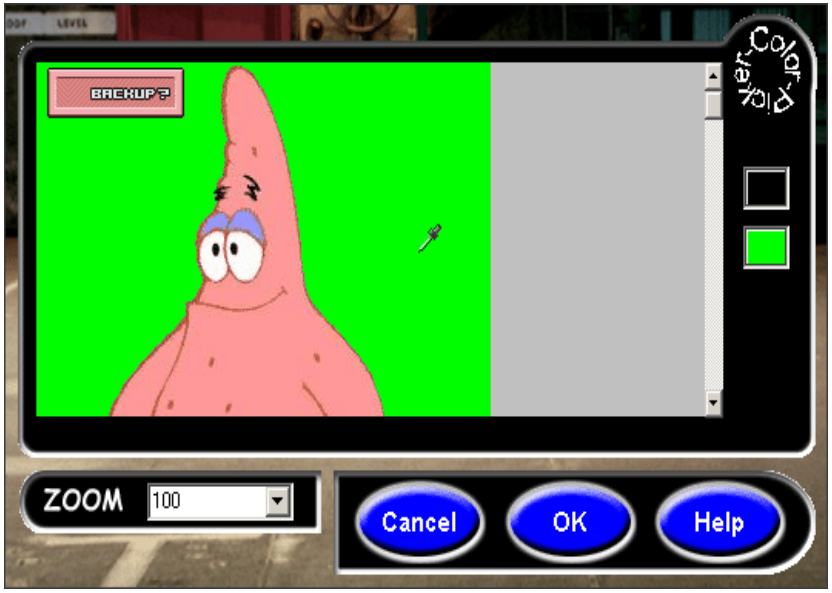
ملف regions.asm:

وهو الملف الذي يحوي على جميع التعليمات اللازمة لتنفيذ هذا العمل، في الواقع التعليمات او الاستخدام للوظائف ليس صعب ويمكن فهمه بسهولة لذا ساترك لكم مسالة مراجعته وعند وجود اي سؤال يمكن مراجعتنا في قسم الاسمبلي التابع للفريق العربي للبرمجة.

باي حال، ساقوم بتفصيل هذه اجزاء البرنامج بشكل دقيق في العدد القادم ان شاء الله لكي يتم استيعاب استخدام الدوال والوظائف بالشكل المطلوب.

- تهيئة الصورة وانشاء region (ملف rgn) خاص بها:

قم بتحميل الصورة في برنامج RGNerator الموجود في مجلد tools، الان انقر على Pick وقم بانتقاء اللون المراد تفريغه..



في الواقع بعد ان تنقر على OK قد تلاحظ انه لم يتم اختيار القيمة اللونية بتنسيق RGB في المربعات الخاصة بذلك في النافذة الرئيسية.. لا ادري ما السبب، لعل الامر يتبع لنظامي او ما شابه..

باي حال اللون المطلوب افراغه يحمل القيمة: RGB(0,254,0)

لذا قم بادخال هذه القيمة في النافذة الرئيسية وانقر على Create.

سيتم تشكيل ملف rgn الان، قمت بحفظه لدي باسم template.rgn في المسار Res\Region

قم بتنفيذ ملف regions.exe الان وستحصل على برنامج بدون خلفية ☺

- مثال آخر؟

قمت بانشاء صورة فيها عبارة "الفريق العربي للبرمجة" ذات خلفية بلون زهري (RGB(255,0,210، وبنفس الطريقة بواسطة RGNerator قمت بتشكيل ملف الـ rgn الخاص بالصورة تلك واسميته arabteam.rgn. الناتج هو:

طبعا يشوبها القليل من اللون الزهري لكن لا باس، فهذا مجرد مثال سريع..

تحميل ملفات المشروع

LINUX

Klik And Run

مجلة TUX عدد ٩، ديسمبر ٢٠٠٥ (http://klik.atekon.de/)

يوفر نظام klik طريقة سهلة و بسيطة لتحميل و تثبيت برامج واجهة KDE.و بسهولة تامة يمكنك حذف البرامج بعد الإنتهاء من تجربتها بإرسالها لسلة المهملات.
توزع برامج klik في صورة ملف فردي يضم جميع المكتبات و المكونات التي يحتاجها البرنامج، و هي تشبه نوعا ما الطريقة التي توزع فيها برامج نظام تشغيل.Apple
يقدم klik روابط مختصرة على الشكل klik://myapp. هذا الرابط السهل يترجمه klik هكذا : "أحضر لي آخر نسخة متوفرة من برنامج myapp من خوادم klik". حالما تنتهي من تثبيتklik ، يمكنك كتابة klik://skype في عارضة العنوان في متصفحك المفضل Konqueror أو Firefox لتحصل على آخر نسخة من برنامج.Skype

كيف يعمل klik ؟

يتعامل klik مع ملفات بإمتداد .cmg تضم جميع المكتبات و الملفات الإعتمادية التي يحتاجها لتشغيل البرنامج. على إفتراض أن لديك أحدث توزيعة لينوكس، لن تحتاج إلى البحث ثم تثبيت الملفات الإعتمادية من أجل حزم.klik
أنت تطلب من خوادم klik وصفة لإعداد حزمة برنامج خاصة بنظامك، إضغط على رابط البرنامج لتحميل جميع الملفات التي يحتاجها مجتمعة كلها في صورة ملف واحد على سطح مكتبك بإمتداد . .cmg عندما تثبت عميلklik ، يتم تثبيت الدعم الضروري لـ Konqueror حتى يتعرف على ملفات .cmg و يمررهم إلى klik عند الضغط عليهم.

ميزات: klik

تثبيت البرامج هي عملية Point-and-click التي لا تحتاج إلى صلاحياتroot klikمنفصل عن حزم النظام، لذلك يسمح للمستخدمين بتجربة البرامج دون الخوف على إستقرارية النظام.
حذف البرامج سهل جدا بسحبها إلى سلة المهملات.
إنقاذ المستخدمين من "جحيم الإعتمادية" بعد تحزيم ملفات الإعتمادية مع البرامج.
إمكانية تغيير مكان برنامج بسهولة، قم بنسخ ملف .cmg إلى مفتاح USB أو CDأو أي مكان آخر في حاسبك.

عيوب: klik

الحزم تأتي من مصادر غير موثوقة، لذلك من المحتمل أن تتضمن شفرات تخريبية.
حزمklik ، في هذه المرحلة، لا تخضع لعملية مراقبة الجودة.
تعدد و إختلاف توزيعات اللينوكس، جعل من الصعب تحزيم برامج لتعمل على جميع الأنظمة، هذا يعني أنك ربما ستصادف أثناء إستخدامك لـ klik حزما لن تعمل على نظامك.

أنظمة التشغيل المدعومة:

هل توجد أنظمة تم إعدادها مسبقا لتعمل مع klik ؟

نعم، هذه الأنظمة تحتوي على klik مثبتا:

```
Kanotix 2005-03 و لاحقاتها
SUSE SUPER SLICK
CPX Mini
RR4 Linux
```

هل توجد أنظمة يمكن تهيئتها للعمل مع klik ؟

نعم، هناك أنظمة تم تجربة klik معها، لكن يجب عليك أن تثبت عميل klik أولا:

```
9.3 SUSE Linux ولاحقتها
Knoppix 3.9 ولاحقاتها
Debian 3.1 Sarge and Sid
```

هل هناك أنظمة غير مدعومة أو غير مجربة مع klik ؟

نعم، و نعرف هذا من إنطباعات المستخدمين حول تجربة klik مع التوزيعات المبينة أسفله

```
Fedora Core 4
Ubuntu و ما بني عليها
Gentoo
Slackware Linux
Mandriva Linux
Xandros
```

هل توجد أنظمة لا تعمل جيدا مع klik ؟

نعم، و هي:

```
أنظمة BSD
أي نظام non-x86
```

هل هناك أنظمة غير متوافقة تماما مع klik ؟

نعم، klikغير متوافق مع الأنظمة التالية:
Microsoft Windows XP (وأي نظام غير اللينوكس)
RedHat Linux 5.0 (وتوزيعات اللينوكس القديمة)
أنظمةBSD

بتجريب توزيعات غير المبينة أعلاه إتضح لنا أن أغلب الحزم تعمل مع التوزيعات الحديثة، قمت بتجربة بعض الحزم على SUSE 10 و SUSE 9.3 و Slackware 10.مع أن بعض الحزم لم تعمل مع جميع التوزيعات إلا أن دعمها يكبر يوما بعد يوم.

تذكر أن klik لا يتعامل مع الملفات أو المجلدات التي يوجد في إسمها فراغ. ماذا يعني هذا ؟ إذا أنشأت مجلدا لبرامجklik ، يجب أن تتأكد من أن إسم المجلد لا يضم فراغا. فبدلا من أن تسميه klik Apps ، سمه klik_Apps.

كيف أثبت klik ؟

تثبيت klik سهل جدا، من قائمة KDE إختر Run Command و أكتب ما يلي:

```
Wget klik.atekon.de/client/install -O - | sh
```

شكل ١

هذا يتطلب وجود برنامجwget
مثبتا على جهازك مسبقا، إذا صادفتك رسالة خطأ تقول "command not found"حاول تثبيتwget
من خلال الأدوات التي توفرها توزيعتك. إذا ظل لا يعمل بعد التثبيت قم بفتح Konsoleو جرب كتابة نفس الأمر فيه. مثلا لم يفلح صديقي في تثبيت klik من خلال Run Command في نظامKubuntu Breezy Badger ، و لكنه نجح عندما كتب الأمر في Konsole على نفس النظام.

في المرحلة التالية، ربما سترى رسالة مثل الرسالة المبينة في الشكل ٢:
شكل ٢
يجب عليك تشغيل script المذكور لإضافة بعض المعطيات إلى ملف .fstab
ماذا يعني هذا ؟
fstabهو ملف يضع قائمة بأنظمة الملفات المختلفة و من يسمح له بالدخول إليها .يحتاج klik إلى أن يكون قادرا على وضع بعض المدخلات في ملف fstabليسمح لك بتهيئة ملفات .klik هذه الملفات تشبه ملفات ISO و للوصول إليها تحتاج إلى تهيئتها مثلما تقوم بتهيئة CD-ROM لتصفحه في مجلد على حاسبك.
لتنفيذScript ، يمكنك إستعمالRun Command ، يجب أن ترى Script في مجلد home الخاص بك أو على سطح مكتبك، إسحب Script إلى مربع Commandحيث تكتب الأوامر عادة. حالما ترى أن مؤشر الفأرة تغير ليسمح لك بإسقاط الملف، أوقف الضغط على زر الفأرة. إضغط بعدها على زر Options، ضع علامة أمام Run as a different user و أكتب كلمة سر المستخدم.root

تثبيت برامج: klik

عند إنتهائك من المرحلة السابقة، سيقوم klik بتشغيل Konqueror و أخذك إلى مستودع .klik سأقوم هنا بشرح طريقة تثبيت و تشغيل آخر نسخة من متصفحOpera ، فهو متصفح جميل و أنا أحب إستعماله لكن توزيعتي لم تقم بتحزيمه .لكن مع klik أستطيع تشغيله خلال دقائق قليلة.
يمكنك إيجاد Opera في قسم Contributed من موقع .klik رسالة تحذير ربما ستظهر لك لتخبرك بأن هذه الصفحة تربطك إلىklik://opera ، هذه ليست مشكلة، هذا بالضبط ما نريده نحن.
سيظهر مربع حوار يسألك ما إذا كنت توافق على تحميل قائمة من الملفات، هذه هي المكونات المنفصلة لوصفة klik و التي ستنشأ ملفOpera.cmg
إضغط Yes و لتظهر نافذة جديدة توضح تقدم عمليتي التحميل و التثبيت.
عندما ينتهي -إذا تم التثبيت بنجاح- سيقوم klik بتشغيل متصفح Opera تلقائيا.
حسنا، كان هذا سهلا جدا، تم حفظ الصورة على سطح مكتبك، كلما أردت إستعمال Opera مجددا، إضغط بكل بساطة على الملف.
إذا كان لديك أقرباء أو أصدقاء يستعملون اللينوكس، فلماذا لا تثبت لهم klik إذ كانوا يستعملون توزيعات مدعومة؟ يمكنهم بهذا تجربة جميع أنواع برامج اللينوكس بطريقة سهلة، آمنة و ممتعة.
يمكنك إستخدام klik لإختبار النسخ التجريبية من البرامج التي لا تقوم التوزيعات بتحزيمها. لأنهم إذا فعلوا، فهذا يعني إستبدال النسخ المستقرة من البرامج، وهذا لن يكون مناسبا خاصة إذا كانت النسخ التجريبية غير مستقرة.
لأن حزم برامج klik منفصلة عن برامج نظامك، لا يجب أن تقلق بخصوص إستبدال النسخ المستقرة. لأنها مستقلة ذاتيا، كما أن حذفها أو تحديثها بسيط جدا. الشكل ٤ يبين الواجهة Enlightenment DR17 تعمل على سطح مكتب .KDE هذه النسخة ما زالت قيد التطوير.
شكل ٤

ألا تريد أن تعرف كيف تحصل عليها ؟

```
klik://Enlightenment
```

ما إن ينتهي تحميلها، إضغط على ملفها من سطح المكتب ليتم تشغيلها.
ميزة جميلة في هذه الحزمة الخاصة و هي أنه تم إنشاؤها لتعمل في Xnestمعدل. بإختصار تعمل تحت خادم X Server آخر و مصغر. هذا يسمح لك بإستخدام آخر ميزات E17 مع بقائك في سطح مكتبك الآمن المألوف.
يمكنك تعديل عرض النافذة بالطريقة الإعتيادية، أو بالضغط على

Ctrl+Shift+Fللإنتقال إلى وضع الشاشة الكاملة.

برامج أخرى:

```
klik://realplayer10gold
```

نسخة اللينوكس الأخيرة من قارئ ملفات Multimedia المتعدد الإستعمالات.

```
klik://apollon
```

برنامج للينوكس P2P لمشاركة الملفات، يستطيع الإتصال بشبكة Gnutella يستعملها LimeWire و (BearShare و) FastTrack يستعملها (Kazaa و آخرين...)

```
klik://nvu
```

برنامج شبيه بـ FrontPage وDreamweaver

```
klik://ar71
```

نسخة اللينوكس من برنامجAcrobat Reader 7

```
klik://opera9
```

النسخة الأخيرة من متصفحOpera

```
klik://skype
```

نسخة اللينوكس من برنامج Skype المعروف.

```
klik://flock
```

متصفح Flock المبني علىMozilla ، يدعمdel.icio.us ، خدمة تشارك قائمة المفضلة التي تمكنك من مشاركة مواقعك المفضلة مع أصدقائك. و Flickr أشهر خدمة نشر الصور. كذلك تم دمجه مع عدة مواقع Blogs عالمية.

مع أن klik تقنية جديدة نسبيا، إلا أنني أتوقع له مستقبلا باهرا. هناك دعم بدئي لسطح مكتب GNOME لهذا سيستطيع مستخدموه إستعمال klik أيضا.
إذا كنت تواجه مشكلة في تشغيلklik ، لا تيأس. توزيعة Kanotix هي توزيعة لينوكس حية تأتي مع klik مثبتا مسبقا.و CPX-MINI التي تعمل من مفتاح USB تأتي أيضا مع.klik
لمزيد من المعلومات، زر موقع .klik و إذا كنت تحبIRC ، يمكنك أن تجد المطورين في غرفة #klik على خادم.freenode

مستودعات البيانات، Datawarehousing

كاتب الموضوع : **Walcom**

أولا يجب تعريف هذه المصطلحات:
OLAP, On-line Analytical Processing: *هي عملية تحليل البيانات المتراكمة في مستودع البيانات.

DSS, Decision Support Systems: *وتعرف كذلك ب EIS, Executive Information Systemsوهي أنظمة تساعد القياديين في المؤسسات والمنظمات على اتخاذ القرارات الحاسمة والمعقدة وذلك بتوفير بيانات من مستوى عالي.

*تنقيب البيانات :Data Mining هي عملية البحث داخل مستودع البيانات عن معرفة غير مستكشفة (اكتشاف المعرفة.)

OLTP, On-line Transaction Processing: *هي مجموعة عمليات الإضافة والحذف والتعديل بالإضافة إلى الاستعلام مع بعض التحليل الذي لايرقى لأن يكون مساعداً في اتخاذ القرارات. ويتوفر دعم هذه العمليات في قواعد البيانات التقليدية.Transactional Databases

ماهي مستودعات البيانات؟

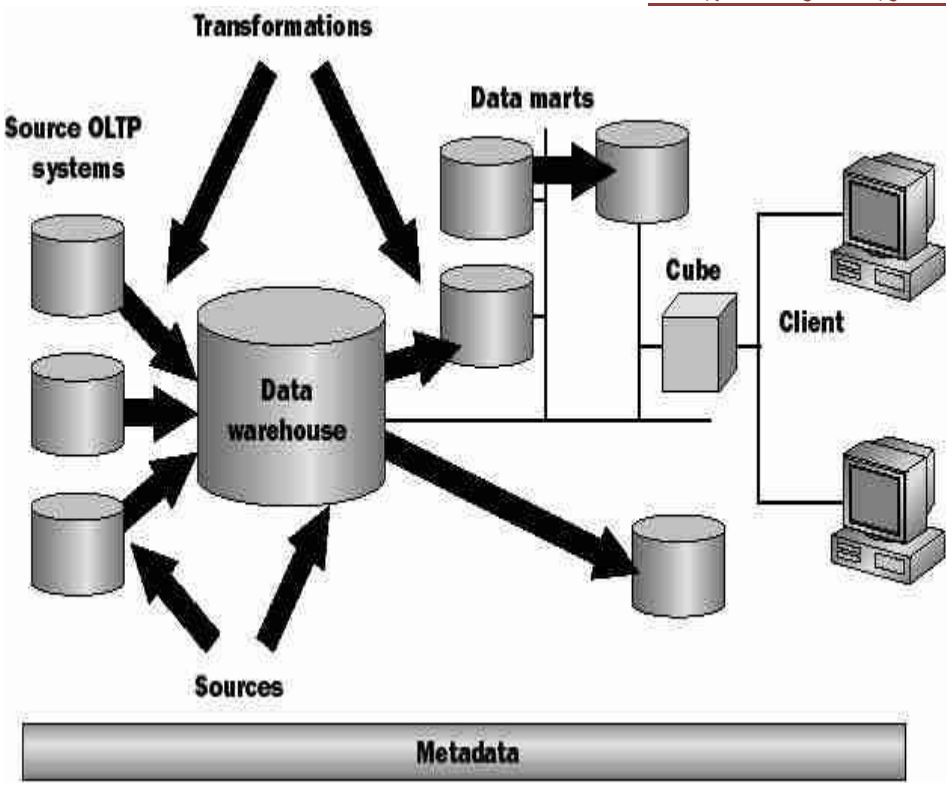
هي مجموعة من البيانات دائمة تاريخية متكاملة للمساعدة في اتخاذ القرارات الإدارية. فهي تساعد على الوصول للبيانات لأغراض التحليلات الزمنية واكتشاف المعرفة واتخاذ القرارات لأنها مصممة خصيصا لاستخلاص البيانات ومعاجتها وتمثيلها وتقديمها بصورة مناسبة لهذه الأغراض، إضافة لذلك فهي تتضمن كمية ضخمة من البيانات قد تكون من مصادر مختلفة، مثلا عدة قواعد بيانات من عدة نماذج بيانات، وأحيانا من أنظمة ومنصات مختلفة.

خصائص مستودعات البيانات:

(1)تستخدم النموذج متعدد الأبعاد.(Multidimensional Model)
(2)تدعم تحليل السلاسل الزمنية (Time Series) وتحليل التوجهات (Trends Analysis)اللذين يحتاجان لبيانات تاريخية لاتستطيع قواعد البيانات العادية (Transactional Databases)توفيرها.
(3)تحديث البيانات فتري (Periodic) أي يتم كل فترة بواسطة أجزاء منه تختص بهذا الأمر.
(4)استرجاع البيانات وتحليلها هو صميم عملها، وتهتم به أكثر.
(5)مستويات تجميع (Aggregation) وأبعاد (Dimensions) غير محدودة.
(6)دعم معمارية Client/server وتعددية المستخدمين.
(7)الاحتفاظ بكمية ضخمة من البيانات قد تصل إلى عدة تيرابايتات (١ TB = 1024 GB) .

الخاصية الأخيرة شكلت مشكلة، ولكن تم حلها بابتكار كل من الآتي:
1-مستودعات البيانات الشاملة :(Enterprise-wide Data Warehouses) وهي مشاريع ضخمة تتطلب استثمارا ضخما في الوقت والموارد.
2-مستودعات البيانات الافتراضية :(Virtual Data Warehouses) وهي استعلامات على قواعد البيانات الوظيفية مصممة بكفاءة عالية للوصول السريع للبيانات.
3-متاجر البيانات :(Data Marts) هي أجزاء من مستودع البيانات موجهة لجزء من المؤسسة (كقسم معين منها.)

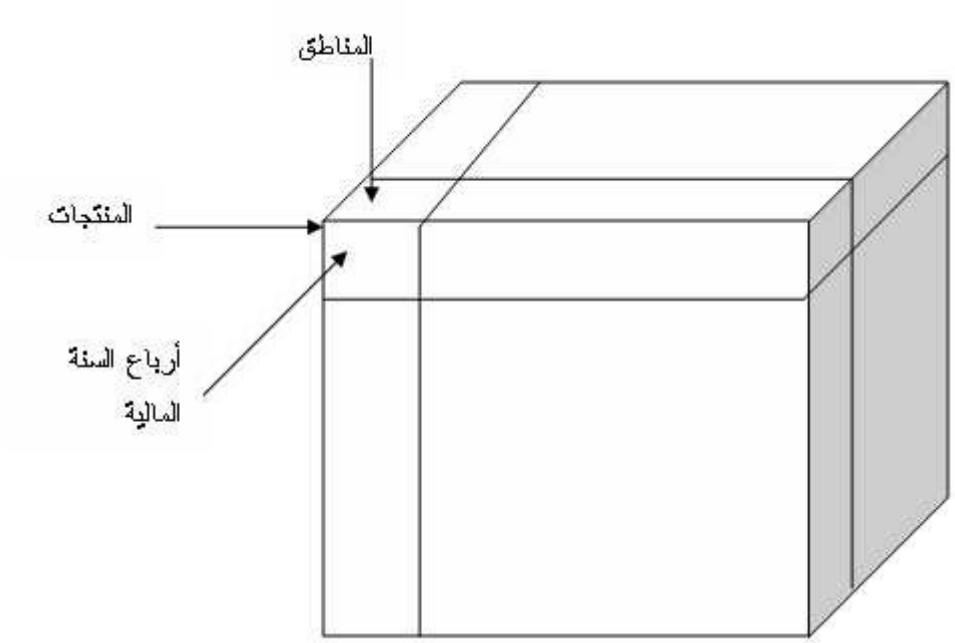
معمارية مستودعات البيانات:



كما هو موضح في الصورة أعلاه، يتم استيراد البيانات للمستودع من قواعد بيانات عادية وتخزينها في المستودع، ومن ثم يمكن تقديم هذه المعلوماتى للمستفيدين منها بصورة مباشرة، أو تجزئتها لمتاجر بيانات ومن ثم تحليل واستخلاص المعلومات المحتواة في متاجر البيانات عبر تكوين مكعبات بيانات يمكن تقديمها للمستخدمين، أو يمكن تكوين مكعبات بيانات أصغر تستخلص البيانات بصورة أكبر من المكعب الذي تم تكوينها منه.

نمذجة البيانات داخل مستودع البيانات:

كما ذكرنا، مستودعات البيانات تستخدم النموذج متعدد الأبعاد، وتستفيد من العلاقات بين البيانات لتسكينها في مصفوفات متعددة الأبعاد تسمى مكعبات البيانات(Data Cubes) ، وتسمى بالمكعبات الفائقة (Hyper Cubes) إذا احتوت على أكثر من ثلاثة أبعاد. البيانات المخزنة في هذا النموذج أفضل من ناحية أداء الاستعلامات من مثيلاتها المخزنة في النموذج العلائقي.
مثال للأبعاد: في مستودع بيانات خاص بشركة تجارية: أرباع السنة المالية، المنتجات، المناطق .وبإضافة بعد آخر (الزبائن مثلا) يتحول لمكعب فائق، مع أنه صعب التخيل والتمثيل.

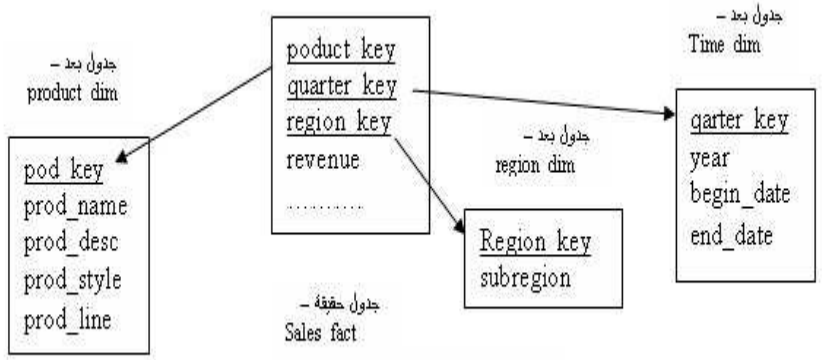


في هذا المثال كل خلية في المكعب تحتوي على بيانات بضاعة معينة في ربع سنة مالية معين في منطقة معينة. عملية الانتقال من تدرج بعد لآخر تسمى بالارتكاز (Pivoting) أو الدوران(Rotation) ، مثلا تدوير المكعب لعرض بيانات المناطق كصفوف ومجموع المبيعات في أرباع السنة المالية كأعمدة، وأنواع البضائع كبعد ثالث. لاحظ تشابهه مع دراسة مجسم دالة متعددة المتغيرات في فضاء متعدد الأبعاد.

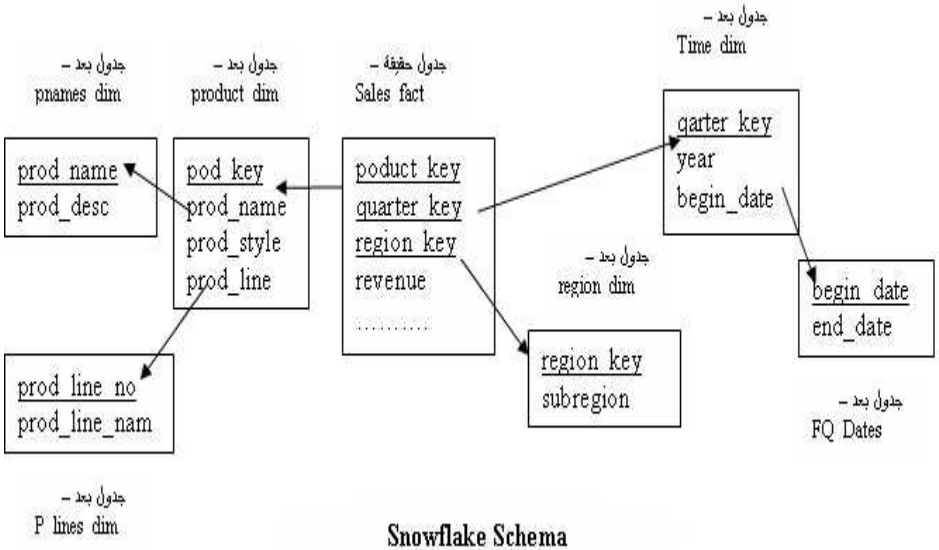
تخزين النموذج متعدد الأبعاد يتضمن نوعين من الجداول:
-1جدول البعد :(Dimension Table) وصفوفه تصف سمات attributes البعد.
-2جدول الحقيقة :(Fact Table) ويتضمن مقاييس أداء العمل (Business Key Performance Indicators)، وتعرَف بمؤشرات لجداول الآبعاد، ويتضمن هذا الجدول البيانات.

توجد بنيتين شائعتين لتعريف هذه الجداول، هما:
1/ Star Schema: وتتكون من جدول الحقيقة مع جدول وحيد لكل بعد.
ويعيبها احتياجها لمساحة كبيرة لتخزين البيانات.
2/ Snowflake Schema: وهي تعديل لل Star Schemaوذلك بتطبيق قوانين التبسيط (Normalization)عليها. ويعيبها تضييعها لطاقة الجهاز المضيف في عمليات الربط.(Joining)

مثال: نموذج قاعدة بيانات مستودع البيانات لشركة تجارية:

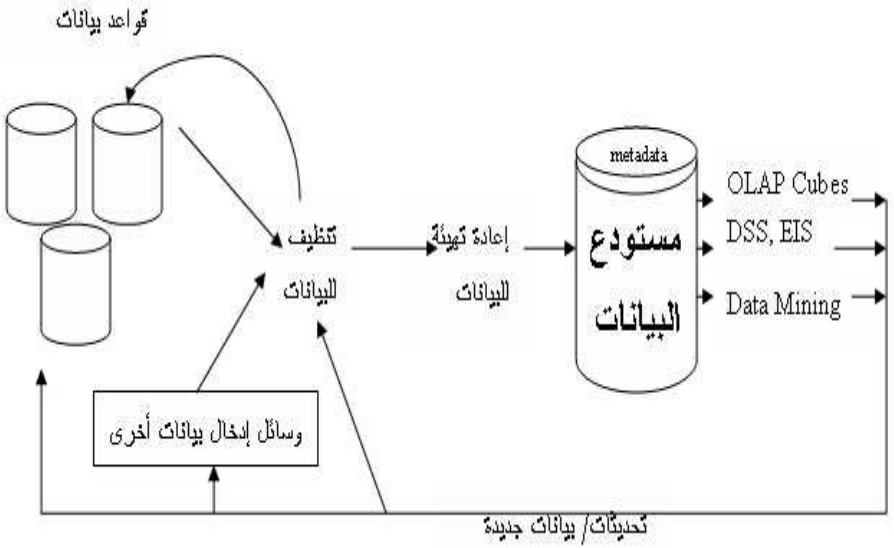


Star Schema



Snowflake Schema

بناء وتشغيل مستودعات البيانات:
العمليات التي تتم لإنجاز هذا العمل موضحة في الشكل التالي:



يتم استيراد البيانات من قواعد البيانات التعاملية / operational transactional databases في كل فترة معينة ويتم تنظيفها وملاءمتها للتخزين في المستودع (إزالة الـnulls، ...إلخ) ومن ثم تخزن في المستودع، وبالتالي يمكن الوصول إليها إما بتكوين مكعبات بيانات منها، أو بكتابة برامج اتخاذ قرارات DSS/EIS أو باستعمال أو كتابة برامج تنقيب عن البيانات. وفي حال حصول أي تعديل عليها أو ضافة لها يتم الحفظ إما في قاعدة البيانات الأصلية أو يتم الإدخال مرة أخرى للمستودع بعد الإخضاع للتنظيف والملاءمة.

قضايا التي تجب مراعاتها عند بناء مستودعات البيانات:
1-استخلاص البيانات يتم من عدة مصادر قد تكون غير متجانسة.
2-تهيئة البيانات لضمان تلاؤمها (Consistency) داخل مستودع البيانات.
3-تنظيف البيانات لضمان شرعيتها(Validity) ، ويتم ذلك في قاعدة البيانات التي تم أخذ البيانات منها.
4-مراقبة وضبط حجم مستودع البيانات خلال وبعد تحميله بالبيانات.
5-كل كم فترة يجب تحديث البيانات فيه؟
6-ماهو الزمن اللازم لبنائه؟ وماهي جدواه الاقتصادية بالنسبة للمؤسسة التي استعملته؟
7-هل نحتاج لأن يكون موزعا (Distributed) أم مركزيا(Centralized) ؟

الوظائف داخل مستودع البيانات:

1-Roll-up:

حيث يتم تلخيص البيانات في شكل عمومي متزايد (أسبوعياً إلى ربعياً إلى سنوياً.)

2-Drill-Down:

حيث تتم زيادة مستويات تفصيل البيانات، بعكس ال.Roll-up

3-Pivot / Rotation:

سبق شرحها أعلاه.

4-Slice and Dice:

تنفيذ عمليات الإسقاط على الأبعاد.

5-Sorting:الترتيب

ترتيب البيانات بقيمة قابلة للترتيب.

6-الاختيار:Selection

اختيار البيانات بقيمة أو مدى قيم.

7-الصفات المحسوبة:Computed Attributed

وهي قيم لصفات يتم حسابها بعمليات على القيم المخزنة والمشتقة.

الفرق ما بين مستودعات البيانات والمرئيات :views

مع أنهما يبدوان متشابهين لأول وهلة، إلا أن بينهما الفروقات التالية:

1-مستودعات البيانات تتواجد كتخزين دائم ولا تتشكل عند الطلب كالمرئيات.

2-مستودعات البيانات ليست دائماً علائقية، بل متعددة الأبعاد.

3-مستودعات البيانات يمكن فهرستها لتحسن الأداء، بينما لايمكن فهرسة المرئيات مستقلةً عن جداولها القاعدية.

4-مستودعات البيانات تستطيع عمل الوظائف المذكورة أعلاه، بينما المرئيات لاتستطيع عملها كلها.

5-مستودعات البيانات تعطي تخزينا ضخما لبيانات زمنية بصورة أكبر من تلك المحتواة في قاعدة البيانات، بينما المرئيات هي خلاصة قاعدة البيانات.

الأدوات :

مكتبات دوالROLAP Relational OLAP ، دوال MOLAP Multidimensional OLAP، امتدادات من SQL وطرق ربط متقدمة (Advanced Join Methods) ومسح ذكي (Intelligent Scanning) وخدمات تحليلanalysis service ، ويتم تحسين أداء هذه الأدوات باستخدام المعالجة المتوازية.

الصعوبات التي تواجه تطبيق مستودعات البيانات :

1/الإنشاء يستغرق بعض الزمن، ابتداءً من وضع الخطط حوله وحتى الانتهاء من تطبيقه.

2/إدارة المستودع صعبة نظرا لكبر لحجمه وتعقيده وتتطلب تدريبا أكثر للقائمين عليه، وخصوصا من ناحية مراعاة جودة البيانات.(Data quality)

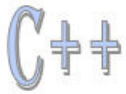
3/تقدير احتياجات مستخدميه قبل إنشائه.

4/ظهور منابع جديدة للبيانات بعد الانتهاء من إنشائه يزيد من صعوبة إدارته.

المصادر:

-R. Almasri and S. Navarthe, Fundamentals of Database Systems, Addison-Wesely, 3rd Ed., p. 845-855

-S. Nolan, T. Huguelet and others, Microsoft SQL Server 7.0 Data Warehousing Training Kit, Microsoft Press, 2000



مؤشرات الدوال

كاتب الموضوع : **Super_emad55**

بسم الله الرحمن الرحيم

أود أن اطرح موضوع ممتع و مهم و مخفي عن كثير من المبرمجين و هو "المؤشرات إلي دوال" أو ما يسمى "Function Pointers" لن أطيل عليكم بالمقدمة، لنبدأ بالتوضيحات,

لنفرض أن لدينا الدوال الاتية بداخل برنامج ما:

```
float Plus (float a, float b) { return a+b; }
float Minus (float a, float b) { return a-b; }
float Multiply (float a, float b) { return a*b; }
float Divide (float a, float b) { return a/b; }
```

فاذا اردنا استخدامها في برنامج معين، سوف نقوم بالاتي:
نكتب الداله الاتيه، لنستدعيها من الmain

```
float Switch(float a, float b, char opCode)
{
    float result;
    switch(opCode) // execute operation
    {
    case '+' : result = Plus (a, b); break;
    case '-' : result = Minus (a, b); break;
    case '*' : result = Multiply(a, b); break;
    case '/' : result = Divide (a, b); break;
    }
    return result;
}
```

ليكون ال main لدينا كالاتي:

```
void main()
{
    cout << Switch(10, 20, '+') << endl;
    cout << Switch(10, 20, '*') << endl;

}
```

للبرنامج اعلاه الامر واضح جدا.
الان هل ياترى نستطيع ان نبدل '+' او '*' باسم الداله نفسه؟؟!!
يعني هل نستطيع ان نقول بالاتي:

```
Instead of
  Switch(10, 20, '+')
Can we do
Switch(10, 20, Plus)
```

لحسن الحظ و لقوة لغة ال c++ نعم نستطيع، فكما ان اسم المصفوفه هو مؤشر لاول عنصر فيها، اسم الداله هو مؤشر لها !! معقول!!

نعم اخي، و كل ما تحتاجه هو ان تخبر ال compiler من أي نوع الداله هي.

الكثير ينتظر الامثله لكي يتضح الوضع.
حسنا هنا مثال واضح جدا:

لنفرض ان الداله التي نستخدمها هي

```
float Plus (float a, float b) { return a+b; }
```

فتكون الدالة المؤشره لداله بهذا الشكل

```
typedef float (* PF) (float a, float b)
```

typedef: دائما بهذا الشكل.
Float: نوع الداله الاصليه
(* PF): اسم المؤشر
(float a, float b): كما هو في الداله الاصليه

و الأن المفاجأه الكبرى، انظروا ماذا يمكننا ان نفعل الان

```
void main()
{
   PF my_function;
   my_function = Divide;
   cout << my_function(10, 20) << endl;
   cout << (* my_function)(10, 20) << endl;
}
```

قد تقول انا فهمت هذه

```
cout << my_function(10, 20) << endl;
```

ولكن ماهذا

```
cout << (* my_function)(10, 20) << endl;
```

اخي الكثير يحب ان يعمل برنامجه معقد و يصعب على الاخرين فهمه "هذا مرض بس اش نسوي"
هاتان الجملتان يقومان بنفس الوظيفه تماما.

```
void main()
{
   cout << New_Switch(10, 20, Plus)<< endl;
   cout << New_Switch(10, 20, Divide) << endl;
}
```

يجب ان نقوم بكتابت الداله الاتيه

```
float New_Switch(float a, float b, PF function)
{
```

```
    return function(a, b);
}
```

لنعرض موضوع اخر للتوضيح
لنفرض ان لدينا هتان الدالتان لطباعة كلمه مثلا، و لتكن بالشكل الاتي:

```
void print1(char * string)
{
    cout << "[" << string << "]" << endl;
}

void print(char * string)
{
    cout << string << endl;
}

void main()
{
     print("one");
    print("two");
}
```

فنقوم بتكوين الداله الاتيه كمؤشر لدوالنا

```
void (* PF) (char * string) typedef
```

و نضعها قبل الmain

فيصبح برنامجنا بالشكل هذا

```
#include <iostream>
using namespace std;


void print1(char * string)
{
    cout << "[" << string << "]" << endl;
}

void print(char * string)
{
    cout << string << endl;
}

typedef void (* PF) (char * string);

void main()
{
    PF printfunction;
    printfunction = print;
    printfunction("one");

    printfunction = print1;
    printfunction("one");
}
```

بالاضافه الى هذا يمكننا ان نقوم بعمل مصفوفه من المؤشرات الى الدوال، ف ال main تكون بهذا الشكل

```
    for(int i = 0; i<4; i++)
    cout << doit[i](5, 10) << endl;
}
```

```
void main()
{
    PF doit[] = {Plus, Minus, Multiply, Divide};
```