



MOST POPULAR

[ปัญหาเกม Angry Birds สำหรับ PC เล่นไม่ได้ บน Windows XP](#)

หลายท่านคงเคยเล่น หรือไม่รู้ จักชื่อเสียงเรียงนามของเกมนี้นักมี ึ่ง หลังจากที่ได้ติดตั้งมาจากวงgTablet และ Smart Phone ก็ได้ ออกVersion ที่เ้...

[ย้ายภาพไปที่อัลบั้มอื่นในfacebook เรื่องง่ายที่มองข้าม](#)

คนที่ม่ภาพใน facebook เยอะๆก็คงจะต ้องการย้ายภาพไปมา บางทีก็เอาภาพบนกระดานเก็บลงอัลบั้มบางทีก็หาไม่เจอว่า จะย้ายได้ ยังไง การย้ายอัลบั้มไม่ยา...

[สร้าง Database จาก ER Model ด้วย MySQL Workbench](#)

MySQL Workbench ที่ผู้ เขียนใช้ ทำตัวอย่างเป็นversion 5.2.34 CE สำหรับ Windows 32 bit หรือตัวใหม่กว่าก็ได้ ครับ ดาวน์ โหลดได้ ที่นี้<http://www...>

[การสร้างและประยุกต์ใช้ Trigger \(SQL Server\) ตอนที่ 1](#)

Trigger คืออะไร? ท่านที่เคยเขียนโปรแกรม และเคยใช้ Event handle ก็จะมีถึงการทำงานของ Trigger ไม่ยาก Trigger ก็ คือ Code คำสั่ง SQL ที่ถูกสร...

[ทำให้ Windows 32 bit เห็น Ram มากกว่า 4 GB](#)

ปัญหาของวินโดวส์ 32 บิตที่มองเห็นหน่วยความจำ 4GB (หรือมากกว่า) ไม่เต็มความจุ กำลังจะหมดไปด้ วยเทคนิคPAE และวิธีย้ายตำแหน่งอ้างอิง เหตุไฉ...

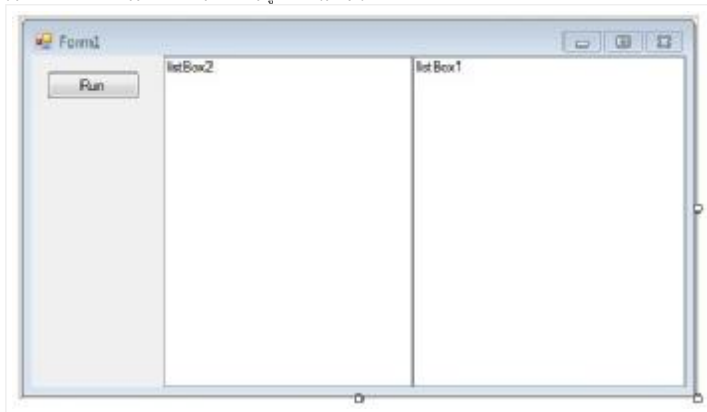
วันพุธที่ 18 พฤษภาคม พ.ศ. 2554

delegate & Invoke ทำ Process จับจ่าย ประมวลผลพร้อ มกันในคราวเดียว

บางครั้งโปรแกรมที่เราต ้องพัฒนาอาจจะมีความจำเป็นในการดึงประสิทธิภาพของprocessor ในการประมวลผลมากที่สุด จำเป็นต ้องใช้ งานInvoke

การใช้ Invoke กับ delegate เป็นตัวอย่างที่ดีที่สุด เพราะง่ายและชัดเจนซึ่งเป็นวิธีที่ผู้ เขียนใช้ ทดสอบ ตัวของ delegate method ที่มีการใช้ BeginInvoke นั้นจะสามารถทำงานพร้อ มกันหลายๆmethod ได้ โดยที่ไม่มีการรอกัน เป็นการทำงาน Multi-Threading อีกแบบหนึ่งที่ใช้ ได้ ดี คำสั่งBeginInvoke จะทำการแยก Thread ออกจาก Common Language Runtime (CLR) และโดนจัดการด้ วยThread pool จึงทำให้ แต่ละdelegate method ทำงานได้ อย่างอิสระต่อกัน และThread ที่เกิดจากการใช้ Invoke สามารถมีได้ ถึง 25 Thread ในการทำงานพร้อ มกันบน Processor ตัวเดียว

ผู้ เขียนได้ ออกแบบฟอร์มเพื่อทดสอบตามรูปด านล่างนี้



การทำงานของโปรแกรมก็คือ เมื่อกดปุ่มRun จะให้ มีการวนลูปสุ่มชื่อเพื่อน เข้าไปในListBox โดยที่ ListBox ด านขวามือจะให้ "Hello" ส่วนด านซ้ายมือจะให้Bye"

ในส่วน Member ของ Class Form1 ผู้ เขียนจะประกาศdelegate method และรายชื่อของเพื่อนในรูปแบบarray

```
private delegate void showMS(string txt,ListBox l);
```

```
private string[] friends = {"David","Ryan","Adam","Briony","Harry","Sara",
"Peter","Maria","Natacha","Lydia","Nina","Macha",
```

 ค้นหา

C# (30) Delphi (15)
Database/SQL (14) Know how (13) Multi-Threading (9) IT News (6) ASP.NET (5) Concept (5) Solve and Debug (5) Tip (3) Java (1) Java Script (1) Other (1)

TOPIC UPDATE

► 2012 (2)

▼ 2011 (76)

► ตุลาคม (3)

► กันยายน (5)

► สิงหาคม (5)

► กรกฎาคม (16)

► มิถุนายน (22)

▼ พฤษภาคม (18)

[ใช้ Lock ทำงานที่ละ Thread](#)

[สอนน้องทำเครื่องคิดเลขด้ วย Delphi](#)

[เริ่มต้นพัฒนาThread Object ใน Delphi \(2/2\)](#)

[เริ่มต้นพัฒนาThread Object ใน Delphi \(1/2\)](#)

[.Net กับ Transactions PART I](#)

[delegate & Invoke ทำ Process จับจ่าย ประมวลผลพร้อ ม...](#)

[ประวัติ และความเป็นมาของ google](#)

[ทำไม Google จึงเป็นหนึ่งเดียวที่ยังไม่มีใครเทียบข...](#)

[C# Thread กับ ProgressBar แบบชีวๆ](#)

[ความต่างระหว่าง "ทำได้" กับ "ทำเป็น"](#)

[แกะรอยดูการทำงานของ ASP.NET MVC \(MVC 2\) PART II](#)

[แกะรอยดูการทำงานของ ASP.NET MVC \(MVC 2\) PART I](#)

[MVC แนวคิดเก่าประยุกต์ให้ใหม่](#)

[จัดการ C# Multi-Threading GUI เพื่อการใช้ งานจริง...](#)

[จัดการ C# Multi-Threading GUI เพื่อการใช้ งานจริง...](#)

[9 ทักษะไอที ที่ทำให้ คุณไม่ตกงาน](#)

[จัดการ C# Multi-Threading GUI เพื่อการใช้ งานจริง...](#)

[ประยุกต์ใช้ C# Threading แยก Process การทำงานของโปร...](#)

► เมษายน (1)

► มีนาคม (6)

```
"Maggy", "Victoria", "Naomi");
```

ต่อมาคือ method ที่ใช้ กับ delegate ที่ประกาศไว้
method นี้จะมี Parameter 2 ตัวคือ string และ ListBox

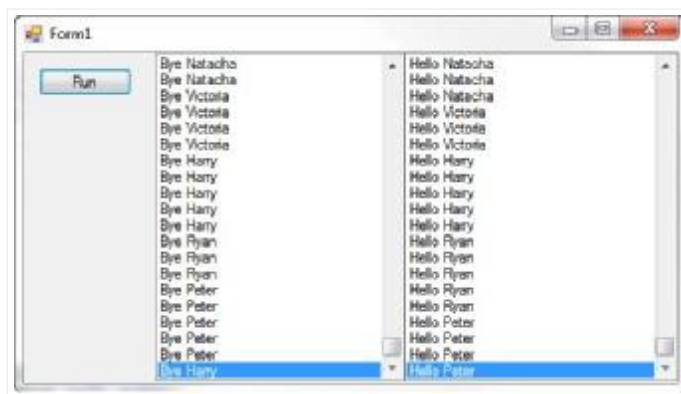
```
private void mess(string str, ListBox lb)
{
    lb.Items.Clear(); //ล้างListBox
    int i=0;
    while (i < 2000)
    {
        //สุ่มชื่อเพื่อนขึ้นมา
        string fName = friends[new Random().Next(0, friends.Length)];
        lb.Items.Add(str + " " + fName);
        lb.SelectedIndex = i;
        i++;
    }
}
```

จากนั้นก็มาถึงคำสั่งที่ใช้ กับปุ่ม Run (button1)
โดยเมื่อมีการกดปุ่มแล้ว จะมีการใส่ค่าลงใน ListBox ทั้ง 2 ตัวพร้อมกัน

```
private void button1_Click(object sender, EventArgs e)
{
    //มีการตรวจสอบ Thread Cross
    Form1.CheckForIllegalCrossThreadCalls = false;
    //สร้าง delegate method
    showMS sms = new showMS(mess);

    //ประกาศ IAsyncResult เพื่อรับค่าสถานะจาก delegate ที่มีการ invoke
    IAsyncResult helloRes = sms.BeginInvoke("Hello", this.listBox1, null, null);
    IAsyncResult byeRes = sms.BeginInvoke("Bye", this.listBox2, null, null);
}
```

เมื่อลองรันโปรแกรมและกดปุ่ม Run ก็จะมีการใส่ค่าใน ListBox ทั้ง 2 ตัวพร้อมกัน



ผู้เขียนจะสร้าง Method เพิ่มเดิมเพื่อให้ทำงานเมื่อจากมี Callback กลับมาจากการ Invoke

- Method helloCallback : เมื่อมีการ Callback ให้ แสดงข้อความ Completed"
- Method byeCallback : เมื่อมีการ Callback ให้ ทำงานซ้ำอีกรอบ และ Callback ไปที่ helloCallback

```
private void helloCallback(IAsyncResult res)
{
    MessageBox.Show("Completed");
}

private void byeCallback(IAsyncResult res)
{
    showMS sms = new showMS(mess);
    IAsyncResult byeRes = sms.BeginInvoke("Bye", this.listBox2, helloCallback, null);
}
```

และแก้ไข Code ในปุ่ม Run (button1) ให้ Invoke มี Parameter ที่เรียกใช้ Callback method

```
private void button1_Click(object sender, EventArgs e)
{
    Form1.CheckForIllegalCrossThreadCalls = false;
    showMS sms = new showMS(mess);
```



SHARE TO ...

[Share this on Facebook](#)

[Tweet this](#)

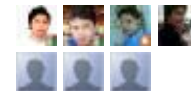
[View stats](#)

[\(NEW\) Appointment gadget >>](#)

FOLLOWING

Google Friend Connect

สมาชิก (7)



เป็นสมาชิกแล้ว หรือขอชื่อเข้าใช้

AUTHOR



อ่านแล้ว:

ปทุมธานี, Thailand

ถึงผมจะไม่ได้ ำเรียนทางด้านคอมพิวเตอร์ แต่เป็นเพราะใจรัก และ

มุ่งมั่นไม่เคยยอมแพ้ อะไรๆ ก็รัก คอมพิวเตอร์ แต่อยู่กับคอมพิวเตอร์มานานเป็น Senior Programmer อยู่ที่บริษัทแห่งหนึ่ง ไม่มี SA ขาแรกเลยทำ सकกะเบียดเรือบ ตั้งแต่องค์กรมาใช้งานได้ ส่วนอีกข้างหนึ่งสองทั้ง แก่ มีปัญหาให้ Programmer มีข้อสงสัย หรือแนะนำติดต่อได้ เลยครับที่ arrays1981@gmail.com

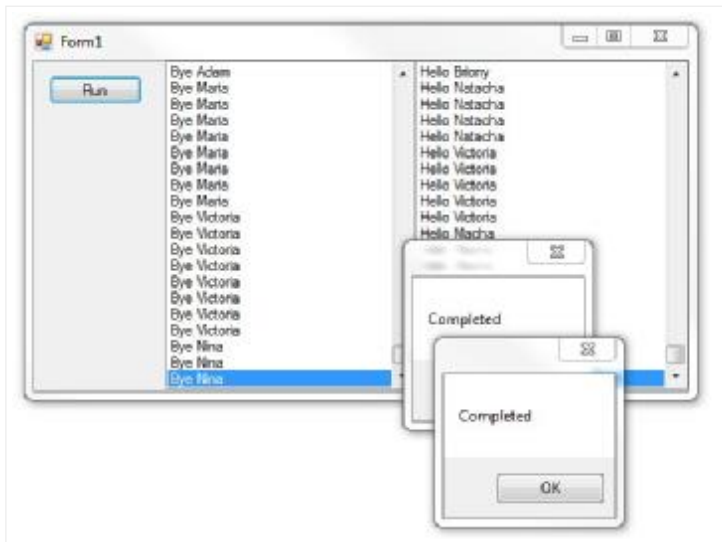
ดูโปรไฟล์ทั้งหมดของฉัน

```

IAsyncResult helloRes = sms.BeginInvoke("Hello", this.listBox1, helloCallback, null);
IAsyncResult byeRes = sms.BeginInvoke("Bye", this.listBox2, byeCallback, null);
}

```

จากนั้นลองทดสอบอีกครั้งจะพบว่ามีการทำงานทั้งหมด 3 รอบ คือ ใส่ขั้ว มัลติใน ListBox คือ Hello 1 รอบ และ Bye 1 รอบ และหลังจาก ListBox ฝั่งซ้ายทำงานเสร็จสิ้นลงจะมีกา Callback จะทำงานต่ออีก 1 รอบก่อนที่จะจบ โดยทำงานไม่รอกัน



หมายเหตุ

```

IAsyncResult helloRes = sms.BeginInvoke("Hello", this.listBox1, helloCallback, null);
IAsyncResult byeRes = sms.BeginInvoke("Bye", this.listBox2, byeCallback, null);
)

```

ค่าสถานะที่ได้ รับจากการInvoke มีรายละเอียดคือ

- | AsyncState : สถานะการทำงานที่ต่างกันออกไป
- | AsyncWaitHandle : รอ และปกป้องให้ การทำงานที่ต่างกันสำเร็จ
- | CompleteSynchronously : บ่งบอกว่าการทำงานที่ต่างกันนั้นสำเร็จ
- | IsCompleted : บ่งบอกว่าการทำงานทั้งหมดเสร็จสิ้นแล้ว

เขียนโดย [อ้วนเสี้ยว](#) ที่ 10:46  +2 [แนะนำใน Google](#)

ป้ายกำกับ: C#, Multi-Threading

ปฏิบัติกรรยา ถูกใจ (0) อยากรให้ชยยกรความ (0) 'ไม่ชอบ (0)

❗️ไม่มีความคิดเห็น

แสดงความคิดเห็น

แสดงความคิดเห็นในฐานะ: Somphop Saejia (Google) ออกจากหน้า

เผยแพร่

แสดงตัวอย่าง

สมัครรับข้ อมูลทางอีเมล

ลิงก์ไปยังบทความนี้

สร้างลิงก์

บทความใหม่กว่า

หน้าแรก

บทความที่กล่าว

สมัครสมาชิก: ส่งความคิดเห็น (Atom)

| ถูกใจ |

207 คนถูกใจสิ่งนี้เป็นคนแรกจากเพื่อน
ของคุณ

เทมเพลต Simple. ขับเคลื่อนโดย [Blogger](#).

