การทดลองตัวอย่างโปรแกรมสำหรับ R&D NID Card DLL Plus

บริษัท อาร์ แอนด์ ดี คอมพิวเตอร์ ซิสเท็ม จำกัด

RDNIDCardDLL_Plus_SourceCode_Usage_R221026.odt

1. ความสำคัญของไฟล์ DLL และ DLD

1.1 ไฟล์ RDNIDLibD.dll คือไฟล์ Library สำหรับใช้ในการติดต่อกับเครื่องอ่านบัตรสมาร์ตคาร์ด และบัตรประชาชน โปรแกรมตัวอย่างทั้งหมดจะมีการเรียกใช้ฟังก์ชันภายในของไฟล์ DLL นี้ เสมอ

1.2 ไฟล์ RDNIDLib.DLD คือไฟล์ที่ใช้ในการตรวจสอบสิทธิ์ในการใช้งาน (License File) ฟังก์ชันใน DLL จะมีการตรวจสอบเครื่องอ่านบัตรสมาร์ตคาร์ดที่ต่ออยู่ว่ามีสิทธิ์ในการใช้งานหรือไม่โดยการตรวจสอบกับไฟล์ DLD นี้ เราสามารถวางไฟล์ DLD อยู่ที่ใดก็ได้ที่โปรแกรมสามารถเข้าถึง แต่โดยปกติ แนะนำให้อยู่ที่เดียวกับไฟล์ DLL เพื่อให้ง่ายต่อการตรวจสอบ

1.3 ดังนั้นผู้พัฒนาโปรแกรมจำเป็นจะต้องนำไฟล์ DLL และ DLD ไปวางไว้ในจุดที่โปรแกรม EXE ทำงาน ทั้งที่ขณะอยู่ระหว่างการพัฒนา (ในโฟลเดอร์ Debug และ Release) และการติดตั้งลงเครื่องของผู้ใช้งาน (ผ่านการสร้างเป็น Package สำหรับการเผยแพร่)

2. ไลบรารี DLL

2.1 กรณีใช้ไลบรารี DLL เป็นแบบ 32 บิต (RDNIDLibD.dll)

- เพื่อให้ไลบรารี RDNIDLibD.dll ที่จะใช้ในการพัฒนาโปรแกรมนี้สามารถใช้งานกับภาษาต่าง ๆ ระบบปฏิบัติการ Windows รุ่นต่าง ๆ และเครื่องคอมพิวเตอร์ที่ใช้หน่วยประมวลผลกลางได้ทั้งแบบ 32 และ 64 บิต ไลบรารี DLL ตัวนี้จึงพัฒนาขึ้นมาบนฐานคำสั่งแบบ 32 บิตทั้งหมด ซึ่งข้อดีคือสามารถใช้กับภาษาคอมพิวเตอร์รุ่นเก่า ที่รองรับเฉพาะคำสั่งแบบ 32 บิตเท่านั้นได้ เช่น VB 6.0 และยังสามารถนำไปใช้กับภาษาคอมพิวเตอร์รุ่นใหม่ ๆ และสามารถใช้กับ Windows ทุกรุ่นไม่ว่าจะเป็น 32 และ 64 บิตได้
- อย่างไรก็ตามในการนำไปใช้กับบางภาษา เช่น VB.NET และ C# ในชุดพัฒนา Visual Studio ผู้พัฒนาจะต้องแจ้งระบุ Platform หรือ Target CPU ให้กับคอมไพเลอร์เพื่อให้คอมไพเลอร์จะได้เลือกการแปลที่รองรับคำสั่งแบบ 32 บิต โดยกำหนด Platform หรือ Target CPU เป็น x86 หากกำหนดเป็นแบบอื่น เช่น x64 หรือ AnyCPU จะใช้งานโปรแกรมบนระบบ Windows 64 บิตไม่ได้ (สำหรับ Visual Studio จะแจ้งรหัสข้อผิดพลาดเป็น Exception from HRESULT:0x8007000B)
- วิธีการตั้ง Platform หรือ Target CPU ขอให้ดูจากรายละเอียดในหัวข้อ **การตั้ง Platform เป็น x86 สำหรับ Visual Studio** ที่อยู่ท้ายเอกสารนี้
- โปรแกรมตัวอย่างที่เป็น VB.NET และ C# ที่ให้ไว้ใน SDK ได้มีการตั้ง Platform เป็นแบบ x86 ไว้ให้แล้ว

2.2 กรณีใช้ไลบรารี DLL เป็นแบบ 64 บิต (RDNIDLibD64.dll)

- โดยปกติควรพัฒนาโปรแกรมเป็นแบบ 32 บิตเท่านั้น เพื่อให้สามารถนำไปใช้งานกับเครื่องคอมพิวเตอร์ได้ทุกเครื่อง

- แต่หากต้องการพัฒนาโปรแกรมให้ทำงานแบบ 64 บิต ให้เปลี่ยนการใช้งานไลบรารี จาก RDNIDLibD.dll เป็น RDNIDLibD64.dll แทน และต้องตั้ง Target CPU เป็นแบบ x64 ด้วย

3. การทดลองตัวอย่างโปรแกรมสำหรับ VB 6.0

3.1 ในระหว่างนำเอาตัวอย่างโปรแกรมไปลองติดตั้งและพัฒนา หากพบปัญหาว่า โปรแกรมหาไฟล์ RDNIDLibD.dll ไม่พบ ให้ Copy ไฟล์ RDNIDLibD.dll จากโฟลเดอร์ต้นฉบับไปวางไว้ในโฟลเดอร์ต่อไปนี้ทีละโฟลเดอร์ (เฉพาะที่มี) แล้วลองดูอีกครั้ง

- \Windows
- \Windows\System
- \Windows\System32
- \Windows\SysWOW64
- c:\Program Files\Microsoft Visual Studio\VB98\
- c:\Program Files (x86)\Microsoft Visual Studio\VB98\

3.2 เมื่อสั่ง Make (คอมไพล์และสร้างไฟล์ EXE) จะต้องมีไฟล์ DLL และ DLD ในโฟลเดอร์เดียวกับไฟล์ EXE ที่คอมไพล์ได้เสมอ จึงจะเรียกใช้งานโปรแกรม EXE นั้นได้

3.3 การสร้าง Package ต้องให้มีการ Copy ไฟล์ DLL และ DLD ไว้ใน Package และกำหนดให้ไปวางลงในโฟลเดอร์เดียวกับไฟล์ EXE ด้วย

3.4 ไฟล์ EXE ที่อยู่ในชุดโปรแกรมตัวอย่าง เป็นไฟล์ที่คอมไพล์มาจากตัวอย่างโปรแกรมที่ให้ไว้ สามารถลองเรียกใช้งานได้ทันที

4. การทดลองตัวอย่างโปรแกรมสำหรับ VB.NET

4.1 ตัวอย่างโปรแกรมนี้ สร้างมาจาก Visual Studio 2019 (VS2019) จึงสามารถโหลดเข้า VS 2019 ได้โดยตรง

4.2 ในการพัฒนาและการใช้งานจะต้องมีการนำไฟล์ DLL และ DLD ไปรวมใน Project ของ VB.NET ก่อน โดยสามารถนำไฟล์ทั้งสองไปวางเองด้วยมือก็ได้ โดยการ Copy ไฟล์ ทั้งสองไปไว้ในโฟลเดอร์เดียวกับไฟล์ EXE ที่คอมไพล์ได้ หรือในโฟลเดอร์ที่จะนำไปใช้งาน หรือจะใช้วิธี Add เข้าไว้ใน Solution ตั้งแต่ขั้นตอนการพัฒนา ขอให้ดูวิธีการในหัวข้อ **การ Add ไฟล์ DLL และ DLD เข้าไปไว้ใน Solution ของ Visual Studio** ที่อยู่ด้านล่าง

4.3 หากจะสร้างโซลูชันขึ้นมาใหม่หมด ต้องไม่ลืมที่จะ Add ไฟล์ DLL, DLD และการกำหนด Platform ให้เป็น x86 ด้วย

4.4 ไฟล์ EXE ที่อยู่ในชุดโปรแกรม เป็นไฟล์ที่คอมไพล์มาจากตัวอย่างโปรแกรมที่ให้ไว้ สามารถลองเรียกใช้งานได้ทันที

5. การทดลองตัวอย่างโปรแกรมสำหรับ C#

5.1 ตัวอย่างโปรแกรมนี้ สร้างมาจาก Visual Studio 2019 (VS2019) จึงสามารถโหลดเข้า VS 2019 ได้โดยตรง และสามารถนำไปใช้กับ Visual Studio รุ่นอื่น ๆ ที่สูงกว่า ได้ด้วย

5.2 จะต้องมีการนำไฟล์ DLL และ DLD ไปไว้ในโฟลเดอร์เดียวกับไฟล์ EXE ที่คอมไพล์ได้ หรือที่จะนำไปใช้งานเสมอ ทั้งในระหว่างการพัฒนาและในการติดตั้งให้กับผู้ใช้งาน ขอให้ดูวิธีการในหัวข้อถัดไป

5.3 โปรแกรมนี้เป็นเพียงตัวอย่าง ผู้พัฒนายังต้องมีการเพิ่มคำสั่งในเรื่องของการป้องกันการคลิกปุ่มอ่านซ้ำซ้อนทั้งที่ยังอ่านบัตรไม่เสร็จ และเรื่องของการดักจับข้อผิดพลาดต่าง ๆ ของโปรแกรม

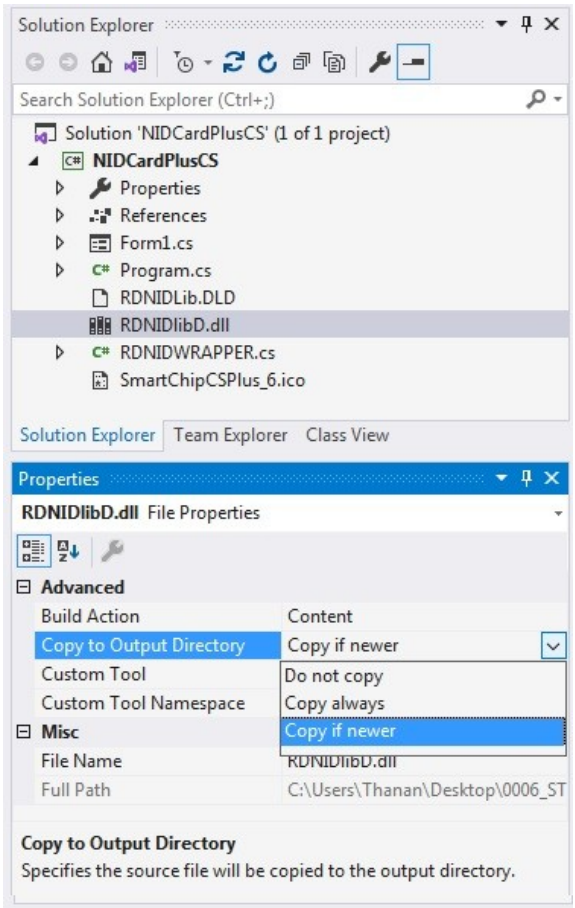
5.4 หากจะสร้างโซลูชันขึ้นมาใหม่หมด ต้องไม่ลืมที่จะ Add ไฟล์ DLL, DLD และการกำหนด Platform ให้เป็น x86 ด้วย

5.5 ไฟล์ EXE ที่อยู่ในชุดโปรแกรม เป็นไฟล์ที่คอมไพล์มาจากตัวอย่างโปรแกรมที่ให้ไว้ สามารถลองเรียกใช้งานได้ทันที

6. การ Add ไฟล์ DLL และ DLD เข้าไปไว้ใน Solution ของ Visual Studio

6.1 ขั้นตอนการนำเอาไฟล์ DLL และ DLD ไปรวมไว้ใน Solution ตั้งแต่ขั้นตอนการพัฒนา วิธีการมีดังนี้

- เปิด Solution ของโปรแกรมขึ้นมา
- ที่เมนู คลิก Project > Add Existing Item... > Object of type = All Files (*.*) เลือกไฟล์ DLL และ DLD ที่อยู่ใน SDK (หากมีหลายไฟล์ ให้เลือกไฟล์ที่มีวันที่ที่ใหม่กว่า) เสร็จแล้วจะเห็นไฟล์ DLL และ DLD ปรากฏใน Solution Explorer (ช่องด้านบนขวามือ)



- คลิกที่ไฟล์ RDNIDLibD.dll หรือ RDNIDLibD64.dll (สำหรับโปรแกรมแบบ 64 บิต) ใน Solution Explorer จะมีข้อมูล Properties ของ RDNIDLibD.dll แสดงในช่อง Properties ด้านล่าง > คลิกที่

Build Action แล้วเลือกเป็น Content > คลิกที่ Copy to Output Directory แล้วเลือกเป็น Copy if newer

- คลิกที่ไฟล์ RDNIDLib.DLD ใน Solution Explorer จะมีข้อมูล Properties ของ RDNIDLib.DLD แสดงในช่อง Properties ด้านล่าง > คลิกที่ Build Action แล้วเลือกเป็น Content > คลิกที่ Copy to Output Directory แล้วเลือกเป็น Copy if newer

หมายเหตุ

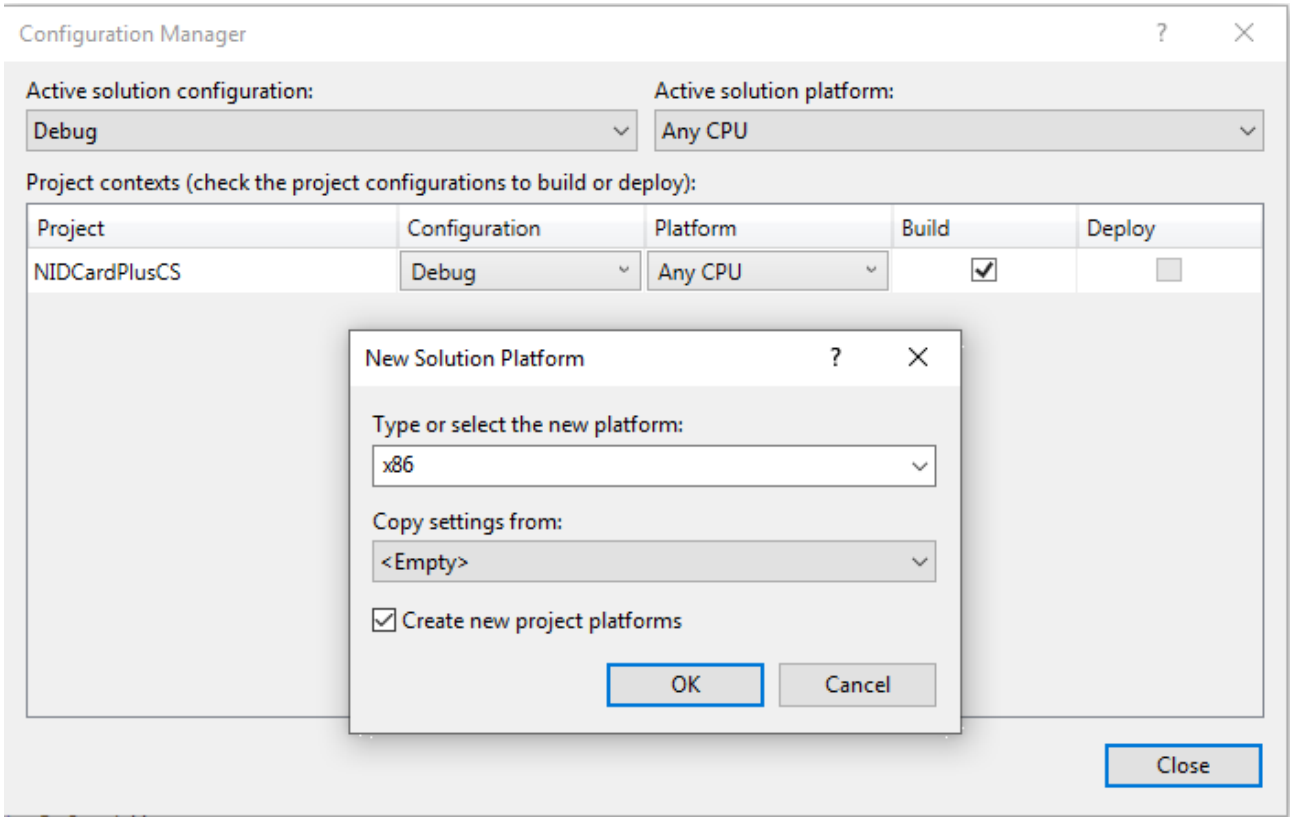
- ให้สังเกตว่าการทำข้างต้นเป็นการ Add Existing Item ไม่ใช่การ Add Reference
- โปรแกรมตัวอย่างของ VB.Net และ C# ที่อยู่ในชุดพัฒนา ได้มีการรวมไฟล์ทั้งสองไว้ใน Solution เรียบร้อยแล้ว

7. การตั้ง Platform เป็น x86 สำหรับการพัฒนาโปรแกรมแบบ 32 บิต ใน Visual Studio

7.1 เพื่อให้โปรแกรมที่พัฒนาขึ้นมาจาก Microsoft Visual Studio สามารถใช้งานได้กับทั้ง Windows 32 บิต และ 64 บิต ผู้พัฒนาจำเป็นต้องกำหนด Platform ในการคอมไพล์ให้ถูกต้อง วิธีการคือ ใช้ Configuration Manager ในการเลือก Platform โดยเลือกเป็น x86

7.2 การตั้ง Platform สำหรับ Visual Studio

- ที่แถบเมนูด้านบน คลิกเปิดเมนูของ Build จะเห็น Configuration Manager อยู่ ให้เรียกใช้งาน Configuration Manager
    - หากไม่พบเมนู Build ให้เลือก Tools>Settings>Expert Settings
    - หากไม่พบบรรทัดเมนู Configuration Manager ให้ทำดังนี้
        - คลิกที่เมนู Tools > Options
        - เฉพาะ Visual Studio บางรุ่น ให้เลือกใส่เครื่องหมายถูกในช่อง Show all settings (อยู่มุมล่างซ้ายของหน้าต่าง Options)
        - คลิกที่ Projects and Solutions และ General
        - เลือก Show advanced build configurations แล้วคลิก OK
        - กลับไปลองคลิกเปิดเมนูของ Build อีกที จะเห็น Configuration Manager อยู่บรรทัดล่าง ให้เรียกใช้งาน Configuration Manager
- เมื่อเรียกแล้ว จะมีหน้าต่าง Configuration Manager ขึ้นมา
- ที่ช่อง Active solution platform คลิกที่ปุ่มลูกศรลง แล้วเลือก x86 แล้วคลิก Close
    - หากไม่พบ x86 ให้สร้างขึ้นมาใหม่เอง โดยเลือก <New...>
    - จะมีหน้าต่าง New Solution Platform แสดงขึ้นมา ที่ช่อง Type or select the new platform ให้เลือกเป็น x86 แล้วคลิก OK



7.3 หลังจากตั้ง Platform เป็น x86 แล้ว ไฟล์ .EXE ที่คอมไพล์ได้ จะอยู่ในโฟลเดอร์ bin\x86