



Udvikling af DOTNET applikationer til MicroStation i C#

Praktiske tips for at komme i gang.

Gunnar Jul Jensen, Cowi

Hvorfor nu det ? Mdl og Vba kan det hele

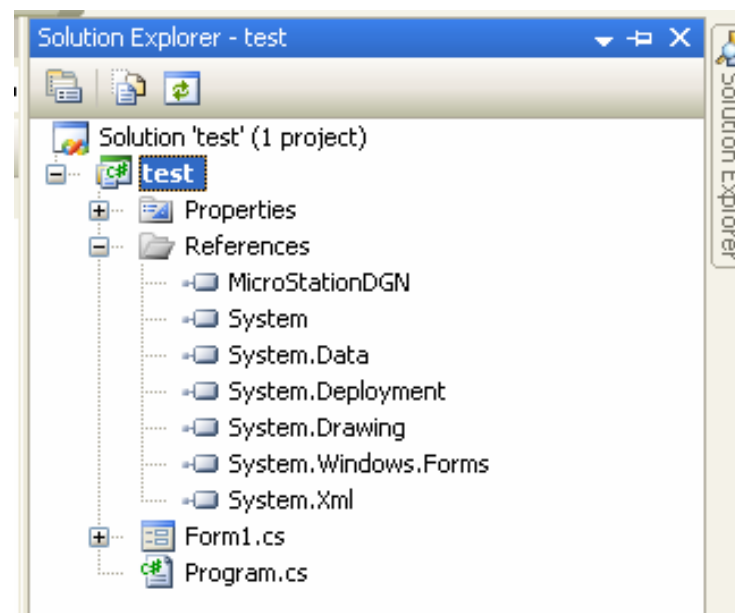
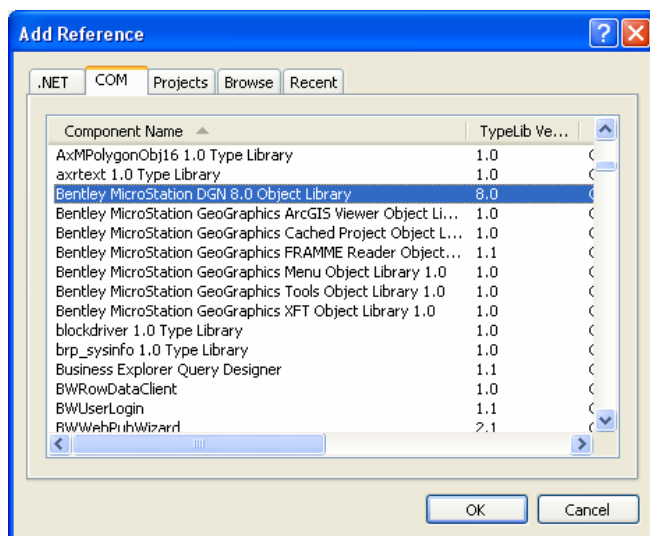


- Fordelene er :
 - udviklingsmiljøet er eksternt i forhold til MicroStation (ikke som vba)
 - Det er objektorienteret.
 - En meget udbredt platform.
 - Mange nye features. En videreudvikling af Studio produkterne

Der kan hentes en (gratis) express version på
msdn.microsoft.com/vstudio/express/visualcsharp/

1. trin MicroStation tilføjes som reference.

- Fra ud udviklingsmiljøet bruges menu Project, add references COM fanen:



Trin 2. Application objekt skal skabes.



```
namespace test {  
    public partial class Form1 : Form {  
        MicroStationDGN.Application m_app;  
        public Form1() {  
            InitializeComponent();  
            m_app = new MicroStationDGN.Application();  
        }  
  
        private void Form1_Load(object sender, EventArgs e) {  
  
        }  
    }  
}
```

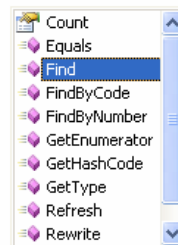
Så er man i gang. Variablen `m_app` er applicationsobjektet som bruges implicit i VBA.

Så er man i gang. Intellisense

```
private void Form1_Load(object sender, EventArgs e) {  
    m_app.  
}
```



```
private void Form1_Load(object sender, EventArgs e) {  
    m_app.ActiveDesignFile.Levels.Find (  
    }  
}
```



Level_Levels.Find(string NamePattern, MicroStationDGN.Level PreviouslyFoundLevel)

Smartere compiler

Fanger logiske programmeringsfejl. Et par simple eksempler

```
private void visFlag(int tal) {
    bool flag;
    if (tal > 1) { flag = true; }
    if (flag == true) {
        MessageBox.Show("Flag er true");
    }
}
```

Error List

1 Error 0 Warnings 0 Messages

Description
1 Use of unassigned local variable 'flag'

```
private void uendeligLokke(int tal) {
    int i=0;
    for(int index=0; index < tal; index) {
        i++;
    }
}
```

Error List

1 Error 0 Warnings 0 Messages

Description
1 Only assignment, call, increment, decrement, and new object expressions can be used as a statement

Region. Orden i koden og overblik.



```
using System;

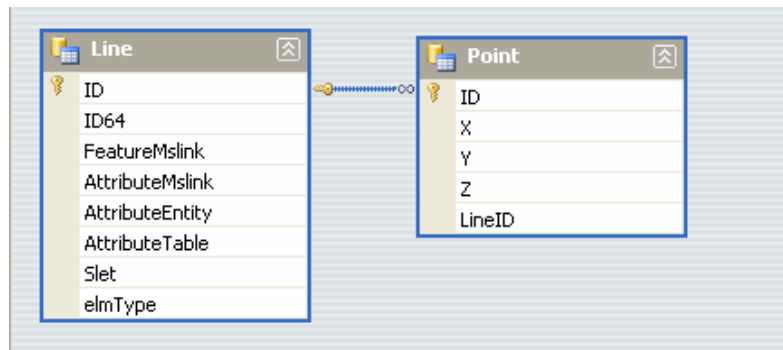
namespace CowiMap.DGN {
    enum ZEROVECTORENUM { ZEROVECTOR, SHORTLINE, PILHOJDE }
    class ZeroVector {
        // datamembers
        // constructor
        // properties
        // private methods

        #region public methods
        public int Start(ScanLevelsEnum scanLevel) {

            CowiMap.DGN.ScanDgnFile scan = new ScanDgnFile();
            m_ids.Clear();
            m_list.Clear();
            m_listIndex = 0;
            scan.setScanClear();
            scan.setScanElementType (MicroStationDGN.MsdElementType.msdElementTypeCellHeader);
            scan.setScanElementType (MicroStationDGN.MsdElementType.msdElementTypeComplexShape);
            scan.setScanElementType (MicroStationDGN.MsdElementType.msdElementTypeComplexString);
            scan.setScanElementType (MicroStationDGN.MsdElementType.msdElementTypeLine);
            scan.setScanElementType (MicroStationDGN.MsdElementType.msdElementTypeLineString);
            scan.setScanElementType (MicroStationDGN.MsdElementType.msdElementTypeShape);
            scan.setScanLevel(scanLevel);
        }
    }
}
```

Dataset En database i memory

- En af mange forbedringer. Avacerede datamodeller uden database. Tabeller, kolonner relationer med defineres grafisk.



The 'Relation' dialog box is shown with the following configuration:

- Name: Line_Point
- Specify the keys that relate tables in your dataset.
 - Parent Table: Line
 - Child Table: Point
- Columns:

Key Columns	Foreign Key Columns
ID	LineID
- Choose what to create:
 - Both Relation and Foreign Key Constraint
 - Foreign Key Constraint Only
 - Relation Only
- Update Rule: None
- Delete Rule: None
- Accept/Reject Rule: None
- Nested Relation

Buttons: OK, Cancel

Hastighed EXE og COM

- Meget vigtig at skelne.
- Sammenligning mellem mdl, vba og almindelige dotnet program (exe fil). Scan stor design fil og tæl antallet af punkter i linestrings.
- MDL: 5 sekunder.
- VBA : 5 sekunder.
- C# exe: 2 minutter !
- C# COM : 10 sekunder.
- Et Exe program ligger i en selvstændig process!
- Et Com modul ligger i samme process.

Hastighed EXE og COM

- Andre opgaver som database operationer og skriv/læs store filer er C# mindst lige så hurtig som MDL og VBA.
- Min erfaring : en lille ulempe. Dotnet verdenen har mange fordele.
- Når en C# aplikation skal behandle større datamængder skal det gøres til et COM objekt.

En dot net applikation som COM objekt.

- Trin 1 . Projekt typen ændres fra Exe til class Modul. Compilering skriver en .dll fil i stedet for en .exe fil. Gøres fra menu : project , properties.
- Trin 2. Tilføj namespace InteropServices og et COMVisible object.

```
using System.Runtime.InteropServices;
namespace CowiMap.DGN {
    [ComVisible(true)]
    public class StartValidate {
        CowiMap.DGN.frmMainValidate m_form = null;
        public StartValidate() {
            if (m_form == null){
                m_form = new CowiMap.DGN.frmMainValidate();
                m_form.Show();
            }
        }
    }
}
```

En dot net applikation som COM objekt.



- Trin 3. Initialiseres fra VBA. I dette eksempel med:
 - Sub Validate()
 - On Error GoTo errHandler
 - Dim f As Object
 - Set f = CreateObject("CowiMap.DGN.StartValidate")
 - errHandler:
 - If Err.Number <> 0 Then
 - MsgBox Err.Description
 - End If
 - End Sub
- Ulempen er at com objekter ikke kan debugges. Derfor udvikle som exe og bruges som COM.

Eksempel på Objekt orienteret kode.

- Generelt problem. Recordset skal lukkes.

```
Dim objRS      As ADODB.Recordset
Set objRS = New ADODB.Recordset
strSql = "select ..."
objRS.Open strSql, objConn
Do While Not objRS.EOF
....
Loop
objRS.Close
Set objRS = Nothing
```

Glemmer man close metoden , får man et memory leak

Det skal man også huske i DOT NET verdenen, men det kan pakkes ind i et objekt, som står for oprydningen og udfører det databehandlingen med en delegate. (En funktion som argument til en funktion)

Brug af delegate



- Funktion erklæres som delegate i en klasse:

```
private delegate void processDataRow(System.Data.DataRow row);
```

- En metode i samme klasse bruger en delegate som argument

```
private void processSql(string sqltxt, processDataRow process) {  
    OleDbCommand selectCommand = new OleDbCommand();  
    OleDbDataAdapter dataAdapter = new OleDbDataAdapter();  
    DataTable table = new DataTable(); selectCommand.Connection = this.m_connection;  
    selectCommand.CommandText = sqltxt;  
    dataAdapter.SelectCommand = selectCommand;  
    dataAdapter.Fill(table);  
    foreach (System.Data.DataRow row in table.Rows) {  
        process(row);  
    }  
    selectCommand.Dispose();  
    table.Dispose();  
    m_connection.Close();  
    return antal;  
}
```

- En bruger af klassen skal nu aldrig mere bekymre sig om at huske at kalde close og dispose.

