

DSLシリーズ第1回

DSLとは？

by 中博俊

DSLとは？

- Domain Specific Language



- 領域特有言語



- 役割を明確にした言語

勝手に言語分類

言語の種類を乱暴に3種類に分類
してみる。

- フル言語
- スクリプト言語
- DSL

勝手に言語分類

- フル言語

- メモリ操作、ファイル操作、通信、外部プログラム呼び出しなど何でもできる
- おもにコンパイルする

- スクリプト言語

- 何でもはできない。
- おもにインタプリタで実行する

- DSL

- ちょっとしかできない。
- おもにコードジェネレートする。

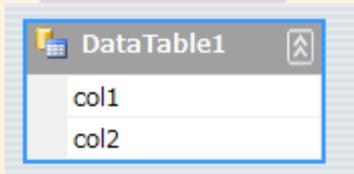
フル言語

- C
- C++
- C#
- Visual Basic
- Java
- Delphi(Pascal)

スクリプト言語

- Java Script
- VB Script
- Shell(csh, ksh, bash等も)
- PowerShell
- Perl
- PHP

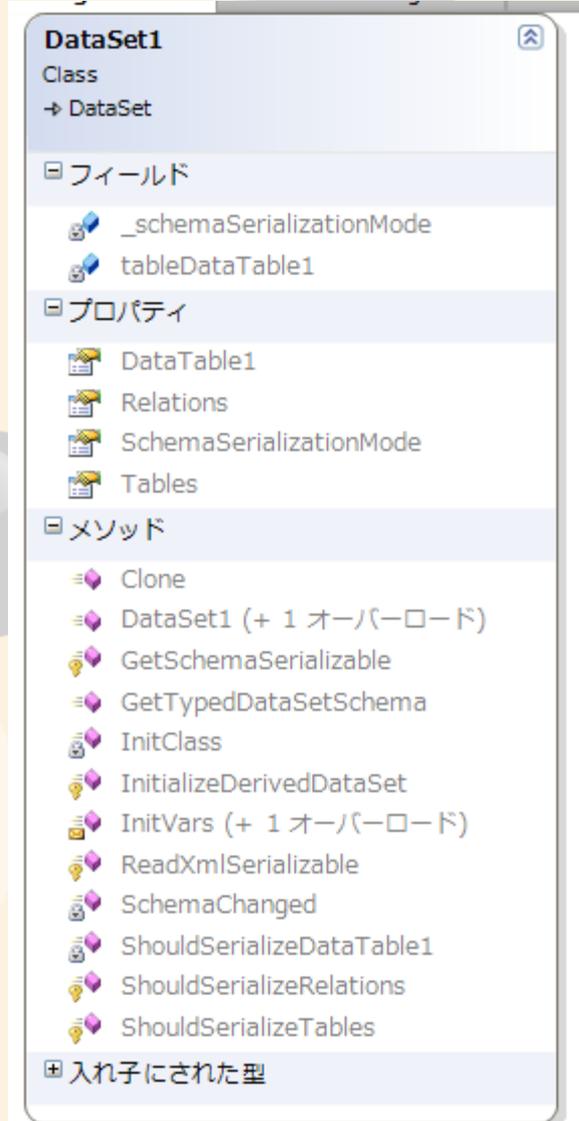
データセットデザイナー



XSD(XMLスキーマ)

```
<?xml version="1.0" encoding="utf-8"?>  
<xs:schema id="DataSet1" targetNamespace="http://tempuri.org/DataSet1.xsd" xmlns:msdn="http://tempuri.org/DataSet1.xsd" xmlns="http://tempuri.org/DataSet1.xsd" xmlns:appinfo="urn:schemas-microsoft-com:xml-msdatasource">  
  <xs:annotation>  
    <xs:appinfo source="urn:schemas-microsoft-com:xml-msdatasource">  
      <DataSource DefaultConnectionIndex="0" FunctionsComponentName="QueriesTableAdapter" Modifier="AutoLayout, AnsiClass, Class, Public, IsDataSource, DataAdapterBase" MsdataProvider="System.Data.SqlClient.ProviderManifest, System.Data.SqlClient" MsdataSchema="System.Data.Schema, System.Data.SqlClient" MsdataSource="System.Data.SqlClient, System.Data.SqlClient" MsdataTable="DataSet1" MsdataView="DataSet1View" MsdataCommand="SELECT * FROM DataSet1" MsdataCommandParameters="<param name='<!-- -->' type='<!-- -->' value='<!-- -->'"/></MsdataCommandParameters></DataSource>  
    </xs:appinfo>  
  </xs:annotation>  
  <xs:element name="DataSet1" msdata:IsDataSet="true" msdata:UseCurrentLocale="true" msprop:Generator_UserDSName="DataSet1" msprop:Generator_DataSourceMethod="DataSet1" msprop:Generator_OriginalName="DataSet1" msprop:Generator_UserTableName="DataSet1" msprop:Generator_OriginalSchema="DataSet1" msprop:Generator_OriginalTable="DataSet1" msprop:Generator_OriginalView="DataSet1View" msprop:Generator_OriginalCommand="SELECT * FROM DataSet1" msprop:Generator_OriginalCommandParameters="<param name='<!-- -->' type='<!-- -->' value='<!-- -->'"/></Msprop:Generator_OriginalCommandParameters></xs:element>  
  <xs:complexType base="DataSet1" msprop:Generator_UserTableType="DataSet1" msprop:Generator_OriginalName="DataSet1" msprop:Generator_OriginalSchema="DataSet1" msprop:Generator_OriginalTable="DataSet1" msprop:Generator_OriginalView="DataSet1View" msprop:Generator_OriginalCommand="SELECT * FROM DataSet1" msprop:Generator_OriginalCommandParameters="<param name='<!-- -->' type='<!-- -->' value='<!-- -->'"/></Msprop:Generator_OriginalCommandParameters></xs:complexType>  
    <xs:choice minOccurs="0" maxOccurs="unbounded">  
      <xs:element name="DataTable1" msprop:Generator_UserTableName="DataTable1" msprop:Generator_OriginalName="DataTable1" msprop:Generator_OriginalSchema="DataSet1" msprop:Generator_OriginalTable="DataTable1" msprop:Generator_OriginalView="DataTable1View" msprop:Generator_OriginalCommand="SELECT * FROM DataTable1" msprop:Generator_OriginalCommandParameters="<param name='<!-- -->' type='<!-- -->' value='<!-- -->'"/></Msprop:Generator_OriginalCommandParameters></xs:element>  
    </xs:choice>  
  </xs:complexType>  
  </xs:element>  
</xs:schema>
```

C#やVBのコード

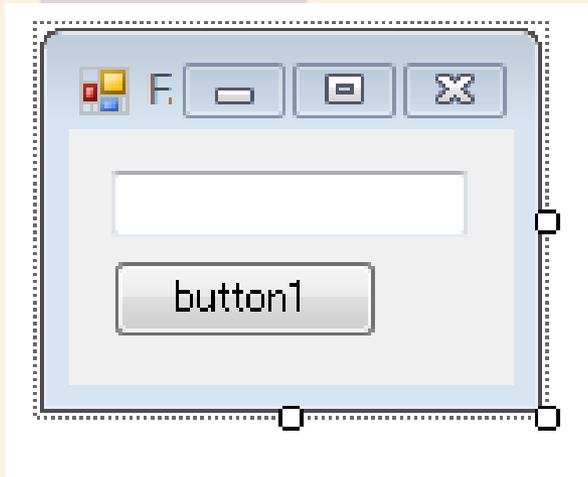


DataSet1
Class
→ DataSet

- フィールド
 - _schemaSerializationMode
 - tableDataTable1
- プロパティ
 - DataTable1
 - Relations
 - SchemaSerializationMode
 - Tables
- メソッド
 - Clone
 - DataSet1 (+ 1 オーバーロード)
 - GetSchemaSerializable
 - GetTypedDataSetSchema
 - InitClass
 - InitializeDerivedDataSet
 - InitVars (+ 1 オーバーロード)
 - ReadXmlSerializable
 - SchemaChanged
 - ShouldSerializeDataTable1
 - ShouldSerializeRelations
 - ShouldSerializeTables

入れ子にされた型

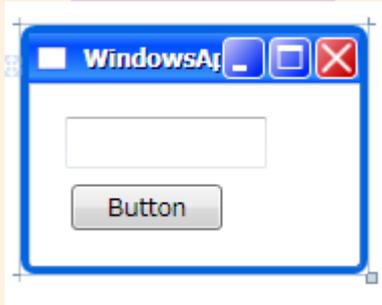
Windows Forms デザイナ



C#やVBのコード

```
private void InitializeComponent()  
{  
    this.textBox1 = new System.Windows.Forms.TextBox();  
    this.button1 = new System.Windows.Forms.Button();  
    this.SuspendLayout();  
    //  
    // textBox1  
    //  
    this.textBox1.Location = new System.Drawing.Point(12, 12);  
    this.textBox1.Name = "textBox1";  
    this.textBox1.Size = new System.Drawing.Size(100, 19);  
    this.textBox1.TabIndex = 0;  
    //  
    // button1  
    //  
    this.button1.Location = new System.Drawing.Point(12, 37);  
    this.button1.Name = "button1";  
    this.button1.Size = new System.Drawing.Size(75, 23);  
    this.button1.TabIndex = 1;  
    this.button1.Text = "button1";  
    this.button1.UseVisualStyleBackColor = true;  
    ..  
}
```

XAMLデザイナー(WPF)



C#やVBのコード

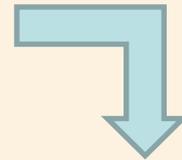
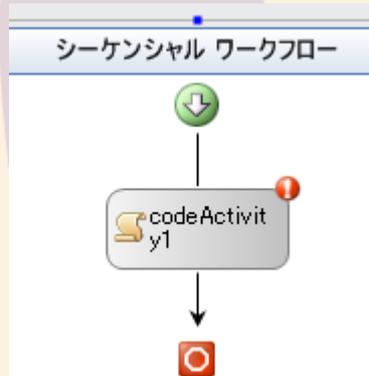
```
void  
System.Windows.Markup.IComponentConnector.Connect(int connectionId, object target) {  
    switch (connectionId)  
    {  
        case 1:  
            this.textBox1 =  
                ((System.Windows.Controls.TextBox)(target));  
            return;  
        case 2:  
            this.button1 =  
                ((System.Windows.Controls.Button)(target));  
            return;  
    }  
    this._contentLoaded = true;  
}
```

XAML

```
<Grid>  
  <TextBox Margin="18,17,0,0"  
    Name="textBox1" Height="26"  
    HorizontalAlignment="Left"  
    VerticalAlignment="Top"  
    Width="100"></TextBox>  
  <Button Height="23" Margin="21,51,0,0"  
    Name="button1" VerticalAlignment="Top"  
    HorizontalAlignment="Left"  
    Width="75">Button</Button>
```

※WPFはほかにも動的解釈などしている

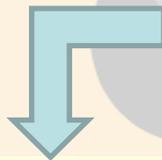
ワークフローデザイナー(WF)



XOML

```
<SequentialWorkflowActivity
  x:Class="WorkflowConsoleApplication2.Workflow2"
  x:Name="Workflow2"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/workflow">
  <CodeActivity x:Name="codeActivity1" ExecuteCode="codeActivity1_ExecuteCode" />
</SequentialWorkflowActivity>
```

MSILに直接
MIXIN ?



```
private void InitializeComponent()
{
    base.CanModifyActivities = true;
    this.codeActivity1 = new CodeActivity();
    this.codeActivity1.Name = "codeActivity1";
    this.codeActivity1.ExecuteCode += new EventHandler(this.codeActivity1_ExecuteCode);
    base.Activities.Add(this.codeActivity1);
    base.Name = "Workflow2";
    base.CanModifyActivities = false;
}
```

SQL

SQL

```
select * from @taba;
```

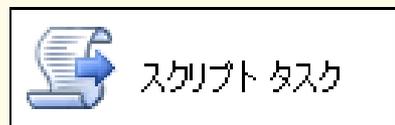
結果ではあるけれど

結果 メッセージ

	col1	col2
1	1	ABC
2	2	DEF
3	3	GHI
4	4	JKL
5	5	MNO
6	6	PQR
7	7	STU
8	8	VWX
9	9	YZ0
10	10	123
11	11	456
12	12	789

SQL Server Integration Service(SSIS)

SSISプロジェクトデザイナー

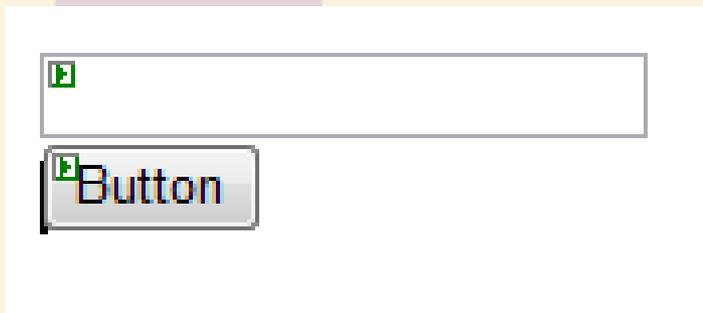


SSISパッケージ

```
<?xml version="1.0"?><DTS:Executable xmlns:DTS="www.microsoft.com/SqlServer/Dts" DTS:Execu
"></DTS:Property><DTS:Property DTS:Name="Creator Name"> localnaka-PC\localnaka</DTS:Property
/6/2007 12:01:54 AM</DTS:Property><DTS:Property DTS:Name="PackageType">5</DTS:Property><DT
perty DTS:Name="PackagePriorityClass">0</DTS:Property><DTS:Property DTS:Name="VersionMajor
S:Property DTS:Name="VersionGUID">{E84B4677-2D85-4CF1-B369-80B8B29916FC}</DTS:Property><DT
ame="SaveCheckpoints">0</DTS:Property><DTS:Property DTS:Name="CheckpointUsage">0</DTS:Prop
<DTS:PackageVariable><DTS:Property DTS:Name="PackageVariableValue" DTS:DataType="8">&lt;Pa
as.microsoft.com/analysiservices/2003/engine/2" xmlns:ddl2_2="http://schemas.microsoft.com
Diagram&gt;&lt;dw:BoundingTop&gt;2143&lt;/dw:BoundingTop&gt;&lt;dw:Layout&gt;&lt;dds&gt
&lt;diagram fontlsid="{0BE35203-8F91-11CE-9DE3-00AA0048B851}" mouseiconlsid="{0BE35204
se.Layout.GraphLayout" version="7" nextobject="7" scale="100" pagebreakanchorx="0" pagebre
1000" zoom="100" x="19394" y="11642" bgcolor="15334399" defaultpersistence="2" PrintPage
epointer="0" snaptogr id="0" autotypeannotation="1" showscrollbars="0" viewpagebreaks="0" c
&lt;/font&gt;↓
&lt;ddsxmlobjectstreamwrapper binary="010100009001905f01000c4d5320554920476f74686963
&lt;/font&gt;↓
&lt;mouseicon&gt;↓
&lt;ddsxmlobjectstreamwrapper binary="6c74000000000000" /&gt;↓
&lt;/mouseicon&gt;↓
&lt;/diagram&gt;↓
&lt;layoutmanager&gt;↓
&lt;ddsxmlobj /&gt;↓
&lt;/layoutmanager&gt;↓
&lt;ddscontrol controlproxid="DdsShapes.DdsObjectManagedBridge.1" tooltip="データ フロー
"0" nomove="0" nodefaultattachpoints="0" autodrag="1" usedefaultidshape="1" selectable="1
```

ASP.NET

ASP.NETデザイナー



aspx

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<title>無題のページ</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox><br />
<asp:Button ID="Button1" runat="server" Text="Button" />&nbsp;&nbsp;&nbsp;</div>
</form>
</body>
</html>
```

C#やVBのコード

```
Object(new LiteralControl("<br />"));
r validator = this.__BuildControlNameTextRequiredFieldValidator();
Object(validator);
dator validator2 = this.__BuildControlNameTextLengthValidator();
Object(validator2);
Object(new LiteralControl("<br />"));
__BuildControlHandleTextBox();
Object(box2);
Object(new LiteralControl("<br />"));
dator validator3 = this.__BuildControlHandleTextLengthValidator();
Object(validator3);
Object(new LiteralControl("<br />"));
der extender2 = this.__BuildControlHandleTextBoxWatermark();
Object(extender2);
Object(new LiteralControl("<br />"));
__BuildControlMailText();
Object(box3);
der extender3 = this.__BuildControlMailTextWatermark();
Object(extender3);
Object(new LiteralControl("<br />"));
r validator4 = this.__BuildControlMailTextRequiredFieldValidator();
Object(validator4);
dator validator5 = this.__BuildControlMailTextFormatValidator();
Object(validator5);
dator validator6 = this.__BuildControlMailTextLengthValidator();
Object(validator6);
Object(new LiteralControl("<br />"));
__BuildControlPrintDocumentCheckBox();
```

DSL

- データセットデザイナー
- SQL
- Windows Form デザイナ
- XAMLデザイナー(WPF)
- ワークフローデザイナー(WF)
- SQL Server Integration Service(SSIS)
- ASP.NET

あなたもDSLを作っていませんか？

あなたもDSLを作った経験がきっとある

Excel

class名	ExchangeRate		
String	Rate		
String	Country		

```
public class ExchangeRate {
    private String _Rate;
    public String getRate() {
        return this._Rate;
    }
    public void setRate(String value) {
        this._Rate = value;
    }
    private String _Country;
    public String getCountry() {
        return this._Country;
    }
    public void setCountry(String value) {
        this._Country = value;
    }
}
```

あなたもDSLを作っていませんか？

あなたもDSLを作った経験がきっとある

Excel

	テーブル名	店舗		
PK	カラム名	型	サイズ	インデックス
1	店舗ID	int		
	店舗名	nvarchar	40	1



```
CREATE TABLE [dbo].[BLOGData](
    [dat] [datetime] NULL,
    [dattime] [datetime] NULL,
    [yyyymm] [char](7) NULL,
    [csUriStem] [varchar](255) NULL,
    [csUserAgent] [varchar](255)
    NULL,
    [csReferer] [varchar](255) NULL,
    [username] [varchar](30) NULL,
    [AggWebFlag] [int] NULL,
    [PageNo] [int] NULL
) ON [PRIMARY]
```

さああなたもDSLを
作っていきましょう。

午後からSSIS、次回以降にWFを研
究します。