

# RE: Xml Serialisierung mit .Net 2 und Generics @Frank

---

*Source:*

<http://www.tech-archive.net/Archive/German/Entwicklung/microsoft.public.de.german.entwickler.dotnet.csharp/2006>

---

- *From:* Sascha Dietl <[SaschaDietl@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:SaschaDietl@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Fri, 30 Jun 2006 01:36:01 -0700
- 

sorry ich hätte vielleicht dazu schreiben sollen das das deserialisieren in die Listen nicht funktioniert

"Sascha Dietl" wrote:

Hallo Frank und auch der Rest der NG,

also jetzt hier meine konkrete Problemstellung zur Xml Serialisierung

Ich habe eine Xml Datei, welche so aussieht (statt werten habe ich hier die Datentypen angegeben):

```
<Assignment>
<Control>
<Name>String</Name>
<Type>Integer</Type>
<Filename>String</Filename>
</Control>
<Databases>
<Database>String</Database>
<Database>String</Database>
<Database>String</Database>
</Databases>
<Fields>
<Field>
<GUID>GUID</GUID>
<Cobra>GUID</Cobra>
<David>GUID</David>
</Field>
<Field>
<GUID>GUID</GUID>
<Cobra>GUID</Cobra>
<David>GUID</David>
</Field>
<Field>
<GUID>GUID</GUID>
<Cobra>GUID</Cobra>
```

## RE: Xml Serialisierung mit .Net 2 und Generics @Frank

```
<David>GUID</David>
</Field>
<Field>
<GUID>GUID</GUID>
<Cobra>GUID</Cobra>
<David>GUID</David>
</Field>
</Fields>
</Assignment>
```

und jetzt das wichtigste und wahrscheinlich die Stelle wo mein Fehler liegt das Objekt:

```
[XmlRoot(AttributeName = "Assignment", IsNullable = false), Serializable]
public sealed class Assignment {
    private List<Field> fields;
    private List<Database> databases;
    private Control control;
```

```
[XmlElement(Type = typeof(Control), ElementName = "Control", IsNullable =
false, Form = XmlSchemaForm.Qualified)]
[EditorBrowsable(EditorBrowsableState.Advanced)]
public Control Control {
    get {
        return control;
    }

    set {
        control = value;
    }
}
```

```
[XmlElement(Type = typeof(Field), ElementName = "Fields", IsNullable
= false, Form = XmlSchemaForm.Qualified)]
[EditorBrowsable(EditorBrowsableState.Advanced)]
public List<Field> Fields {
    get {
        return fields;
    }

    set {
        fields = value;
    }
}
```

```
[XmlElement(Type = typeof(Database), ElementName = "Databases",
IsNullable = false, Form = XmlSchemaForm.Qualified)]
[EditorBrowsable(EditorBrowsableState.Advanced)]
public List<Database> Databases {
    get {
```

```

return databases;
}

set {
databases = value;
}
}

public Assignment() {
//leere Listen definieren
Fields = new List<Field>();
Databases = new List<Database>();
}

}

```

an dieser Stelle lasse ich Control weg denn das Funktioniert einwandfrei,  
also hier die beiden Item Klassen:

```

[Serializable]
[DebuggerStepThrough]
[DesignerCategory("Code")]
[XmlType(AnonymousType = true)]
public sealed class Database {
private string name;

[XmlElement(ElementName = "Database", IsNullable = false, Form =
XmlSchemaForm.Qualified, DataType = "string")]
[EditorBrowsable(EditorBrowsableState.Advanced)]
public string Name {
get {
return name;
}

set {
if (value != "") {
name = value;
}
}
}

public Database() {
Name = string.Empty;
}
}

[Serializable]
[DebuggerStepThrough]
[DesignerCategory("Code")]
[XmlType(AnonymousType = true)]
public sealed class Field {

```

```
private string guid;

[XmlElement(ElementName = "GUID", IsNullable = false, Form =
XmlSchemaForm.Qualified, DataType = "string")]
[EditorBrowsable(EditorBrowsableState.Advanced)]
public string GUID {
get {
return guid;
}

set {
guid = value;
}
}

private string david;

[XmlElement(ElementName = "David", IsNullable = false, Form =
XmlSchemaForm.Qualified, DataType = "string")]
[EditorBrowsable(EditorBrowsableState.Advanced)]
public string David {
get {
return david;
}

set {
david = value;
}
}

private string cobra;

[XmlElement(ElementName = "Cobra", IsNullable = false, Form =
XmlSchemaForm.Qualified, DataType = "string")]
[EditorBrowsable(EditorBrowsableState.Advanced)]
public string Cobra {
get {
return cobra;
}

set {
cobra = value;
}
}

public Field() {
GUID = string.Empty;
David = string.Empty;
Cobra = string.Empty;
}
}
```

RE: Xml Serialisierung mit .Net 2 und Generics @Frank

Jetzt bin ich natürlich für jede Hilfe dankbar.

Grüße  
Sascha