

Dynamisches Maskendesign

Mittwoch, 19. März 2014

Track 1 / Session 8

11:00 – 12:30 Uhr

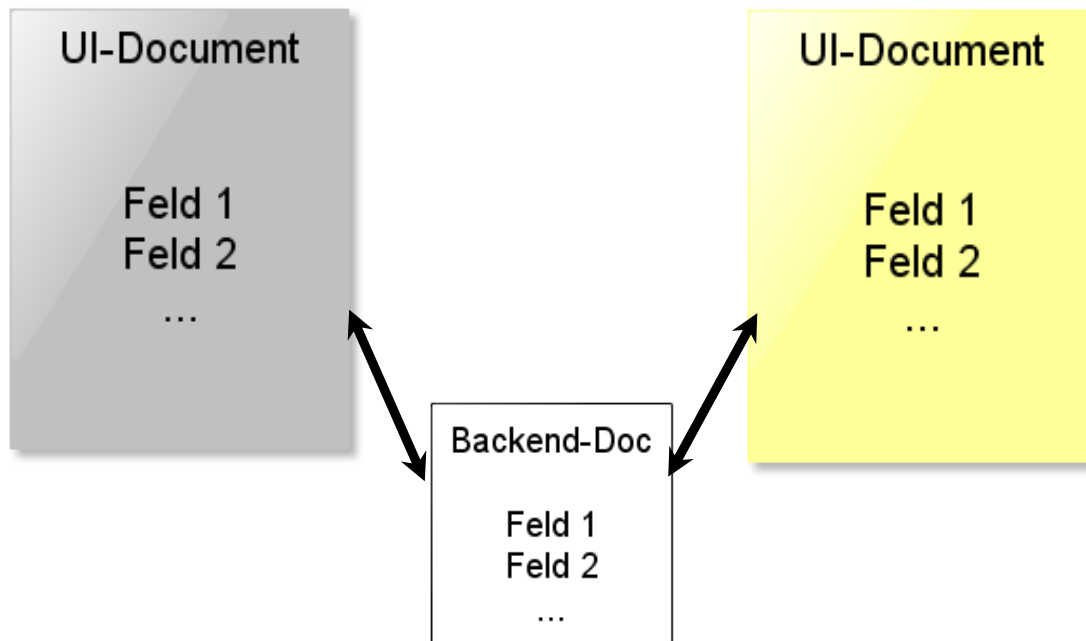
Bert Häßler



**NOTES & DOMINO
ENTWICKLERCAMP**

- Masken und Teilmasken
 - Austauschen im UI
- Designelemente manipulieren
 - DXL-Export/Import
- Aktionen mit Serverrechten ausführen
 - Call Agent.RunOnServer()
 - Class ScriptOnServerExecuter
- „personalisierte“ - Designelemente

Demo



```
' Script zum Erneuern eines UI-Documents
```

```
Set doc = uidoc.Document
```

```
doc.Form = "XXX" ' <- ganz wichtig,
```

```
' da dieses Feld bei ungespeicherten Dokumenten noch nicht existiert
```

```
' 2. UI-Instanz öffnen:
```

```
Set new_uidoc= Workspace.EditDocument(True, doc)
```

```
doc.SaveOptions = "0" ' Speichern? - Frage unterbinden
```

```
Call uidoc.Close ' Instanz 1 schließen
```

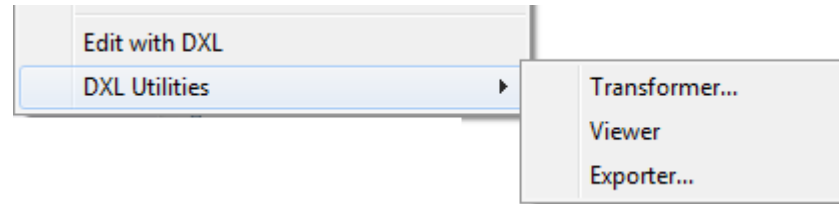
```
Call new_uidoc.Document.RemoveItem("SaveOptions")
```

```
Call new_uidoc.Reload ' sonst wirkt die Zeile darüber nicht...
```

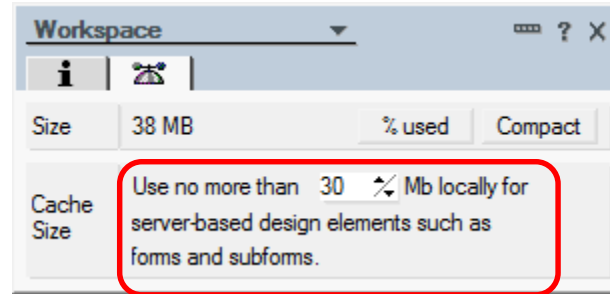
```
Call new_uidoc.Refresh ' falls berechnete Felder existieren
```

- Hide-When-Bereiche:
 - Sinnvoll bei einzelnen Feldern
 - Speicherung unnützer (leerer) Felder
 - Unübersichtlich bei komplexeren Zusammenhängen
- Teilmasken:
 - Separierung von Spezialfällen
 - Corporate Design
 - Eigene Maskenereignisse (erst Maske, dann TM)
 - Langsamer (?)
 - Performancetrick: Dummycode

- Tools\DXL Utilities
- Edit with DXL
- Class NotesDXLExporter/NotesDXLImporter
 - NotesStreams
- Suchen, Bearbeiten
 - Parser DOM/SAX
 - Stringfunktionen



- Standardfunktionalität des Notes-Clients
 - Cache.ndk



- Designänderung erst nach erneutem Öffnen der DB wirksam



- Der Trick:

Anhang im Designelement !!!

- Call Agent.RunOnServer()
 - Signatur des Agenten bestimmt das Zugriffsrecht
 - Auch Leser können die Aktion auslösen
 - Parameterübergabe möglich
 - Parameterdokument muss in DB gespeichert sein

```
Class ScriptOnServerExecuter
```

```
Function ExecuteScript(code, doc)
```

```
    ' temporäres Parameterdokument anlegen und speichern
```

```
    ...
```

```
    Call agent.RunOnServer(id)
```

```
    ...
```

```
End Function
```

```
Sub delete
```

```
    ' Parameterdokument löschen
```

```
    ...
```

```
    Call agent.RunOnServer(id)
```

```
    ...
```

```
End Sub
```

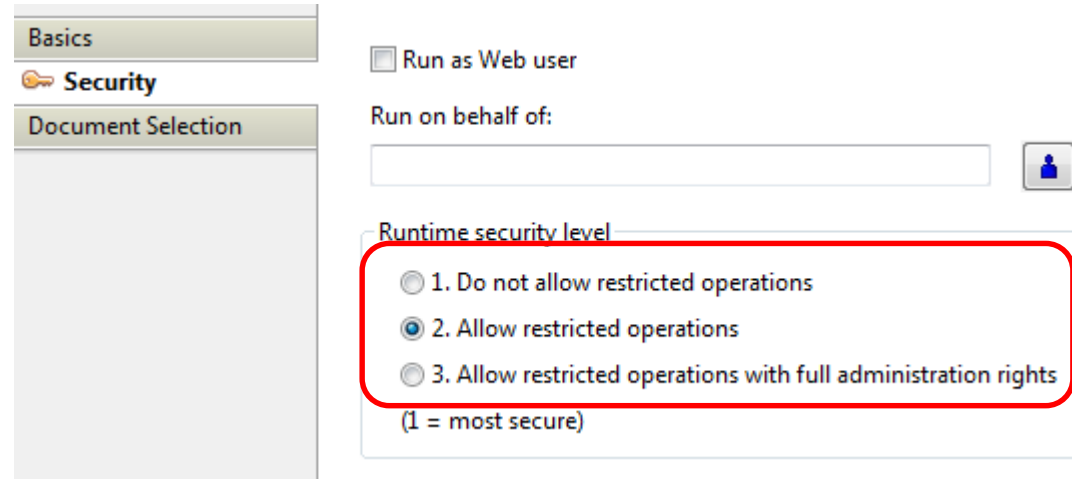
```
End Class
```

```
Sub Initialize
  ' Parameterdokument übernehmen
  Set tdoc = db.GetDocumentByID(ses.currentagent.ParameterDocID)


  Select Case tdoc.GetItemValue("$Action") (0)
  Case "Execute":
    code = tdoc.GetItemValue("$Code") (0)
    Execute code ' <---
  Case "Remove":
    Call tdoc.Remove(True)
  End Select

End Sub
```

■ Security Tab



The screenshot shows the configuration interface for the Security tab. On the left, there is a sidebar with three tabs: 'Basics', 'Security' (which is active and highlighted with a key icon), and 'Document Selection'. The main area on the right contains the following settings:

- Run as Web user
- Run on behalf of: [Empty text box] 
- Runtime security level** (highlighted with a red box):
 - 1. Do not allow restricted operations
 - 2. Allow restricted operations
 - 3. Allow restricted operations with full administration rights
- (1 = most secure)

- DXL-Import von Design-Notes ist keine „restricted operation“
- Signieren aber schon!

- Nutzergesteuerte Designänderungen kritisch bei parallelen Nutzerzugriffen
- Unterschiedliche Benennung
 - z.B. Person_BertHaessler.subform
- Einbindung als berechnete Teilmaske

- Kontakt:
 - Bert Häßler
 - bert.haessler@leonso.de
 - www.xing.com/profile/Bert_Haessler
 - de.linkedin.com/pub/bert-haessler/63/383/b5

- Alle Unterlagen zu unseren Vorträgen beim EC 2014 in Kürze unter:
 - www.leonso.de/ec14