Webkomponente Multistructure

1 Einleitung

Mit der multi-structure- Komponente lassen sich XML-Dialoge auf einer HTML-Seite gliedern. Die Multi-structure Komponente erstellt hierzu eine aufklappbare Hierarchie der einzelnen Dialoge. Eine „hinzufügen“-Funktion fügt neue Dialogtypen in derselben Hierarchieebene dazu.

2 XML-Definitionsaufbau

2.1 Beispiel

```
<multistructure>
	<topmostelement>
		<entity>Surveys</entity>
	</topmostelement>
	<level>
		<xmldialog>/ASP/demo/xmlMulti1.xml</xmldialog>
		<height>200px</height>
		<width>100%</width>
		<name>Oberlevel</name>
		<entity>Questions</entity>
		<relation>
			<thislevel>qst_svy_id</thislevel>
			<toplevel>svy_id</toplevel>
		</relation>
		<level>
			<relation>
				<thislevel>qst_svy_id</thislevel>
				<toplevel>svy_id</toplevel>
			</relation>
			<name>Unterlevel</name>
			<xmldialog>/ASP/demo/xmlMulti2.xml</xmldialog>
			<height>200px</height>
			<width>100%</width>
		</level>
	</level>
</multistructure>
```

2.2 Positionierung Webkomponente

Die Webkomponente darf NICHT innerhalb eines FORM-tags („<form…>“) stehen, sondern muss unbedingt NACH dem FORM-tag positioniert werden. Das Formular muss vom Browser komplett eingelesen sein, damit per Javascript auf die Formelemente zugegriffen werden kann. IE und Netscape verhalten sich verschieden, wobei NS7 sofortigen Zugriff aufs Formular erlaubt. Beispiel:

Falsch:

```
<form name="frmMain">
     <ef:Multistructure id="MultiStructure1"/>
     <button../>
</form>
```

RICHTIG:

```
<form name="frmMain">
     <button../>
</form>
<ef:Multistructure id="MultiStructure1"/>
```

2.3 Hinweis zu Relation bei oberster Level-Ebene

Mittels des relation-tags können Beziehungen zwischen den Levels angegebenen werden, z.B. wird eine Umfrage mehrere Fragen enthalten. Die Fragen wiederum enthalten mehrere Antwortmöglichkeiten. Die unteren Levels lassen sich komplett über das Frontend erzeugen. Die oberste bildet insofern eine Ausnahme, da das übergeordnete Element der obersten Ebene, nicht angelegt werden kann und schon vorhanden sein muß.
Legen Sie in den Eigenschaften `sTopMostElementValue` und `sTopMostElementName` daher den Namen und Wert der übergeordneten Eigenschaft fest. Im Beispiel unserer Umfrage, bei der über die Multi-structure-Komponente die Fragen bearbeitet werden können, wird in `sTopMostElementValue` der Name des Primärschlüssels der Umfrage und als Wert in `sTopMostElementName` der Wert des Primärschlüssels der Umfrage gesetzt.

2.4 Hinweis zum TopMostElement

Das Element `<topmostelement>` gibt das übergeordnete Objekt an. Dieses Objekt kann nicht dynamisch vom Multi-Structure angelegt werden, sondern muss von vornherein vorhanden sein. Legen Sie hier die Entität fest.

2.5 Beispiel eines JavaScript-Aufrufs zum Speichern:

```
function mOnOk() {

      var sDlgData = gsXMLDlgOutput(document.forms["frmMain"], "", true);

      //call the asp-page which stores the dialog-data into
      //the database; you can use the default multistructure-store-asp
      //or write your own:
      var sResult =
gsCallServerMethod("/ASP/system/multistructure/storing/storeProcess.aspx",
sDlgData);
```

```
        if (sResult == "SUCCESS") {
              gReloadMultiStructure("EfMultiStructure1");
        }
        else {
              alert(sResult);
        }

}
```

Wichtig ist, dass nach jedem Speichern die Funktion „gReloadMultiStructure“ aufgerufen wird, damit der Dialog neuaufgebaut wird und damit die Schlüsselwerte eingetragen. Damit wird verhindert, dass neu eingefügte Datensätze immer wieder neu angelegt, statt upgedatet, werden.

3 Dialogdaten laden

3.1 Datenformat

Die Datapage zum Laden des Dialoginhalts muss folgendes Format einhalten:

1. Zeile:
OK-||-

2. bis x…. Zeile:

<levelname>|	**<parentid> |**	**<id>|**	**<field-name>|<field-value>|….**	**-||-**
Name des erzeugenden Levels aus XML-Datei	Frei wählbar; keien Sonderzeichen	Frei wählbar; kein Sonderzeichen	Feldname gefolgt vom Feldwert bis zum …	…abschließenden Zeilenumbruch „-||-„

Achtung: die ersten Levelelemente müssen als parentid einen Leerstring übergeben!

3.1.1 Beispiel für Datenformat

```
OK-||-
QUESTIONS|| QUESTIONS_1|qst_id|1|qst_text|Wie heißen Sie?-||-
QUESTIONS||2|qst_id| QUESTIONS_2|qst_text|Wie heißt Ihr Hund?-||-
ANSWERS|QUESTIONS_2|ANSWERS_1|ans_id|1|ans_value|Waldi-||-
ANSWERS|QUESTIONS_2|ANSWERS_2|ans_id|2|ans_value |Purzel-||-
ANSWERS|QUESTIONS_2|ANSWERS_3|ans_id|3|ans_value |Dweeb-||-
QUESTIONS||QUESTIONS_3|qst_id|3|qst_text|Was machen Sie so?-||-
```

3.2 Benutzen von storeDialogData.aspx und condition-tag

Standardmäßig wird die Date /ASP/system/multistructure/storeDialogData.aspx verwendet, um die Inhalte in den Multistructure-Dialog zu laden. Dies ist deshalb möglich, da jedes Level eine Entität besitzt. Beim Laden wird also hierarchisch zunächst das Top-Element geladen und dann alle Unterkinder.

Eine Besonderheit ergibt sich nun, wenn für eine Levelebene, verschiedene Level-Nodes angegeben sind. Die storeDialogData.aspx muß nun entscheiden, welcher dieser Levelnodes verwendet werden soll, da evtl. verschiedene XML-Dialog für jeden dieser Levelnodes hinterlegt sind.

Verwenden Sie in diesem Fall ein **condition**-Tag:

```
<loadinstruction>
      <entityfield>ans_aty_id</entityfield>
      <value>FREETEXT</value>
      <value>GAPTEXT</value>
</loadinstruction>
```

Setzen Sie dieses Tag direkt unter das Level-Element.

Im obigen Beispiel geschieht nun folgendes. Sehen Sie sich zunächst die schematische Darstellung an:

```
<level>
      <name>HighestLevel</name>
      <level>
            <name>SubLevel 1</name>
            <condition>
                  <entityfield>ety_type</entityfield>
                  <value>1</value>
                  <value>2</value>
            </condition>
      </level>
      <level>
            <name>SubLevel 2</name>
            <condition>
                  <entityfield>ety_type</entityfield>
                  <value>3</value>
            </condition>
      </level>
      <level>
            <name>SubLevel 3</name>
            <condition>
                  <entityfield>ety_type</entityfield>
                  <value>4</value>
                  <value>5</value>
            </condition>
      </level>
<level>
```

Zunächst wird das Level **HighestLevel** erstellt. Anschließend werden sämtliche Entitäten der unteren Levels geladen. Es können ja durchaus verschiedene Entitäten in den unteren Levels dranhängen.

Ein Level wird allerdings nur dann dargestellt, wenn die **condition** erfüllt wird. Im Fall von **SubLevel 1** wird dieses nur dann angezeigt, falls das angegebene Entitätenfeld entweder den Wert **1** oder **2** besitzt. Damit ließe sich z.B. einstellen, dass abhängig von einem bestimmten Feldwert, verschiedene XML-Dialoge verwendet werden.

4 Alle Tags (Übersicht)

/multistructure	Stammelement
/multistructure/topmostelement	Das über alles übergeordnete Element
/multistructure/topmostelement/entity	Die am meisten übergeordente Entiät
/multistructure/level	Level-element
/multistructure/level/name	Eindeutiger Name des Levels; eindeutig über alle Hierarchien
/multistructure/level/xmldialog	Dateiname des XML-Dialogs z.B. /ASP/Project/1/xmldialog.xml
/multistructure/level/height	Höhe in der Darstellung in Pixeln
/multistructure/level/width	Breite in der Dartellung in Pixeln
/multistructure/level/entity	Die Entität, die hier dargestellt wird
/multistructure/level/newbutton	Beschriftung des Einfügen-Buttons
/multistructure/level/relation	Beziehung zum übergeordneten Element, z.B. cst_grp_id
/multistructure/level/relation/thislevel	Feldname des jetzigen, unteren Levels
/multistructure/level/relation/toplevel	Feldname des übergeordneten Levels, z.B. grp_id
/multistructure/level/condition	Bedingung, wann der Level geladen werden soll (optional)
/multistructure/level/condition/entityfield	Name des Feldes, das überprüft wird
/multistructure/level/condition/value	Wert, den das Feld haben muss; es können mehrere Values angegeben warden
/multistructure/level/level…	Beliebige Verschachtelung von Levels (Achtung: Name des Levels muss eindeutig bleiben)
/multistructure/level/optiondialog	Einstellungen für einen Optionsdialog
/multistructure/level/optiondialog/text	Beliebiger Anzeigetext im Optionsdialog; steht im übergeordneten Level, von dem aus der Auswahldialog angezeigt wird
/multistructure/level/optiondialog/caption	Bezeichnung des aktuellen Levels im Option-Dialog
/multistructure/level/optiondialog/newbutton	Bei übergeordneten Element muss hier der String eingegeben werden, der auf dem New-Button erscheint
/multistructure/topmostelement/optiondialog	Auch für ein Top-Most-Element lässt sich ein Option-Dialog definieren

5 Interne Programmierkonzepte

Zugegeben: die Programmierung dieser Multistructure wird wohl alleine durch studieren des Programmcodes nicht ganz einfach fallen.

Grund: die Komponente ist optimiert geschrieben. Es wird an mehreren Stellen dasselbe gerendert, um zweifache Aufrufe zu vermeiden.

5.1 Grundsätzlich gilt

- Jedes eingefügte Level bekommt eine eindeutige ID. Diese sieht immer so aus „LV_x_y_z_...“

  Daraus lässt sich dann folgender Hierarchiebaum beispielseweise erzeugen:

  LV_1_
  LV_1_1_
  LV_1_1_1_
  LV_1_1_2_
  LV_1_1_3_
  LV_1_2_
  LV_1_2_1_
  LV_2_
  LV_2_1_
  …
- Ein Level besteht aus:
  - XML-Dialog
  - Neu-Schaltfläche zum Einfügen eines Levels auf gleicher Ebene
  - Neu-Schalftfläche zum Einfügen eines Levels auf unterer Ebene
- Eine Problematik ist das „Anfangen“ des Editierens eines Levels. Am Anfang befindet sich nur die Schalftfläche „Neu“ ganz oben im Dialog. Klickt man auf diese, so wird das erste Level der obersten Ebene gerendert und eingefügt. Über die Schaltfläche „Neu“, lassen sich dann weitere Levels auf gleicher Ebene einfügen.
  Klappt man das soeben eingefügte Level über die drei Punkte „…“ auf, so findet man die Schaltfläche zum Einfügen von Unterlevels.

  Anmerkung: stehen auf einer Hierarchieebene, verschiedene Levels zu Verfügung, so kann der Benutzer über einen zwischengeschalteten Optionsdialog auswählen, welches dieser Level eingefügt werden soll. Im Level-Tag werden über das Element <optiondialog/> Angaben hierfür gemacht. Sogar auf oberster Ebene, also bei der allerersten „Neu“-Schaltfläche, erscheint bereits dieser Optionsdialog. Hierzu muss im Element `/multistructure/topmostelement` das Element OptionsDialog angegeben werden.

5.2 Benötigt werden folgende Komponente, um den multistructure darzustellen

- **Assembly efMultistructureHelper**
  - Rendern des XML-Dialogs (gsRenderSpecificLevel)
  - Rendern der möglichen Optionswerte im Zwischenauswahldialog (gsRenderOptionValuesForNewButton)
- **/ASP/js/efMultistructure.js**
  - Ermöglicht Javascript-Interaktion;
  - ruft Server-Methoden
  - enthält Init-Funktion zum Initialisieren des Multistructure in HTML-Seite
- **/ASP/system/multistructure/ondemandrenderer/newbuttonOptions/newButtonOptions.aspx**
  - Rendert einen Ergebnisstring, der die Möglichen Optionswerte für den Zwischendialog enthält, falls auf die Schalftfläche "Neu…" gedrückt wird
- **/ASP/system/multistructure/ondemandrenderer/renderLevel/renderLevel.aspx**
  - Ruft Funktion in Assembly efMultiStructureHelper auf, um ein komplettes Level zu rendern
  - Zu einem kompletten Level gehört der gerenderte XML-Dialog sowie die Neuschaltfläche zum Einfügen von Levels auf gleicher Ebene, sowie der darunterliegenden Ebene
- **/ASP/system/multistructure/storeDialogData/storeDialogData.aspx**
  - Wird verwendet, um die Daten des Multistructure-Inhalts zu laden
- **/ASP/system/multistructure/storeProcess/storeProcess.aspx**
  - Diese ASPX-Seite speichert die Daten des Multistructures
  - Es wird mit Hilfe von Transaktionen atomar gespeichert (schlägt eine Speicherung eines Levels fehl, so findet ein Rollback statt)
- **Webkomponente efMultiStructure**
  - Die Webkomponente ruft die Init-Javascript-Funktion auf

5.3 Ablauf Multistructure

5.3.1 Initialisierung des Multi-Structure

Es wird davon ausgegangen, dass die Multistructure-Komponente in einen Dialog eingebaut wird:

1. Aufruf von MultistructRenderer.gsRenderInit
2. Rendern des Aufrufs der Javascript-Funktion `msGetMultiInitHtml`
3. Aufruf von `msGetMultiInitHtml`
   a. Setzen von lokalen Javascript-Variablen
   b. Erzeugen der hidden-fields für die TopMost-Elemente
   c. Einfügen des allerersten New-Buttons; die Beschriftung dieses New-Buttons wurde als Parameter an die Funktion übergeben
   d. Anschließend wird `gReloadMultiStructure` aufgerufen, damit der Multi-Structure von der Datapage Daten lädt

5.3.2 Funktion mAddNewLevel

Diese Funktion ist für den Aufbau des Multistructure im HTML sehr ausschlaggebend.

mAddNewLevel wird beispielsweise aufgerufen, wenn der Multistructure seine Daten aus der Datapage reinlädt, sowie, wenn auf die Schaltfläche „Neu“ gedrückt wird.

mAddNewLevel erwartet die Parameter:

- sParentId
- sLevelName

Es fügt einen Level in der selben Hierarchieebene an, wie in sLevelName angegeben. Der Typ des Levels der eingefügt wird, ist sLevelName.

sParentId enthält die ID (s. Grundsätzlich gilt) des übergeordneten Levels, z.B. „LV_1_4_5_“.

mAddNewLevel ruft standardmäßig die Aspx-Datei renderLevel.aspx auf, die wiederum die Methode **gsRenderSpecificLevel** aus dem Multistructure-Renderer aufruft.
Diese Methode liefert einen Semikolon-separierten String, der momentan (31.05.2004) folgende Werte enthält:

- **sNamePraefix**
  entspricht sNewLevelId, z.B. „LV_1_4_2_“
- **sLevelName**
  Name des Levels aus Element <name> (Eindeutig im MultiStructu-Xml)
- **sLevelHeight**
  die Höhe des Dialogs in Pixeln
- **sLevelWidth**
  die Breite des Dialogs in Pixeln
- **sNewButtonCaption**
  Die Beschriftung des Buttons, um ein neues Level auf gleicher Ebene hinzuzufügen
- **sOptionDialogText**
  Text der im Zwischendialog angezeigt wird, falls auf dieser Levelebene mehrere Levels vorhanden sind, und der Benutzer eine Auswahl treffen kann
- **sNextSubLevelName**
  Das nächste untergeordnete Element spielt hier deshalb eine Rolle, da zu einer Leveleinheit, wie bereits oben erwähnt, der XML-Dialog, der New-Button für die gleiche Hierarchieebene, sowie der New-Button für die untergeordnete Hierarchieebene gehört
  Für die Variablen **sNextSubLevelName**, **sNextSubLevelNewButtonCaption** und **sOptionsOfNextSubElement** ergibt sich eine Besonderheit, mit der

Einführung des zwischengeschalteten Optionendialogs und wird daher weiter unten detailliert erläutert

- **sNextSubLevelNewButtonCaption**
  s. sNextSubLevelName
- **sOptionsOfNextSubElement**
  s. sNextSubLevelName
- Immer als Abschluß: **gerendertes HTML des XML-Dialogs** (hier können ja auch Semikolons enthalten sein, deshalb wird dieses HTML immer ans Ende gehängt)

5.3.2.1 Beschreibung von sNextSubLevelName, NextSubLevelNewButtonCaption und sOptionsOfNextSubElement:

Diesen Parameter kommt eine besondere Bedeutung seit der Einführung des zwischengeschalteten Optionsdialogs zu.

Nehmen wir beispielsweise an, dass dem Level A genau ein Level B untergeordnet werden kann.
In diesem Fall würde in **sNextSubLevelName** der Wert „**Level B“** stehen, **sNextSubLevelNewButtonCaption** würde die entsprechende, benutzerdefinierte Beschriftung enthalten, z.B. „Level B einfügen“. **sOptionsOfNextSubElement** wäre dagegen leer. Es werden keine Auswahloptionen benötigt, da genau ein Leveltyp hinzugefügt werden kann.

Wäre es dagegen so, dass unter Level A, die Levels Level B, Level C, Level D …. hinuzgefügt werden können, dann ist **sNextSubLevelName** leer, da nicht genau festgestellt werden kann, welches Level exakt hinzugefügt werden soll. Der Benutzer muß eine Auswahl treffen.
sNextSubLevelNewButtonCaption enthält den benutzerdefinierten Text der Schaltfläche.
sOptionsOfNextSubElement enthält die Auswahlwerte für den zwischengeschalteten Auswahldialog. Die Methode **gsRenderOptionValuesForNewButton** erzeugt den Inhalt, der über die Aspx-Seite **newbuttonOptions.aspx** geladen wird. Dieser String enthält in sich zwei Semikolons.
sOptionsOfNextSubElement enthält folgende Felder:

- **sOptionDialogText**
  Der benutzerdefinierte Text, der im Optionendialog angezeigt wird. Normalerweise steht hier Text wie z.B. „Bitte wählen Sie einen der folgenden Einträge aus:“ drinnen
- **sOptionDialogValues**
  Dies ist eine Liste der Werte, die zurückgeliefert werden, wenn eine Option ausgewählt wird. Es handelt sich hier um eine Liste, die durch die Zeichenfolge „-||-„ getrennt ist.
  Hier stehen die Namen der Levels drinnen, in unserem Beispiel wäre das also: **“LEVELA-||-LEVELB-||-LEVELC-||-…….“**
- **sOptionDialogCaptions**
  Hier werden die Beschriftungen der Optionen übergeben. Auch diese sind mit der Zeichenfolge „-||-„ abgetrennt.

5.3.3 mAddNewButton

Diese Funktion fügt die Schaltfläche „Neu“ hinzu. Sie fügt die Neuschaltfläche für in folgenden Fällen hinzu:

- Allererster Neu-Button
- Neu-Button zum Einfügen von Levels auf **gleicher** Ebene
- Neu-Button zum Einfügen von **untergeordneten** Levels

mAddNewButton erwartet folgende Parameter:

- **sParentId**
  Id des Levels, zu der der New-Button gehört, z.B. „LV_1_2_“
- **sLevelName**
  Name des Levels, das mit dieser Schaltfläche erzeugt wird
- **sNewButtonCaption**
  Beschriftung der Schaltfläche
- **bForLevelsOnSameHierarchy**
  Wenn True, dann erzeugt der New-Button Levels auf gleicher Ebene. Wenn False, dann erzeugt der New-Button untergeordnete Levels.
- **bFetchOptionValuesFromAsp**
  Wenn **True**, wird die Aspx-Seite **newButtonOptions.aspx** aufgerufen, um die Werte für die Auswahlwerte zu erhalten

  Falls sie **False** angeben, so werden die Optionswerte aus den Parametern **sOptionDialogText, sOptionDialogValues und sOptionDialogCaptions** verwendet.

  Damit kann ein Server-Roundtrip zum Aufruf von newButtonOptions.aspx vermieden werden

- **sOptionDialogText, sOptionDialogValues u. sOptionDialogCaptions**
  s. Beschreibung von sNextSubLevelName, NextSubLevelNewButtonCaption und sOptionsOfNextSubElement:

Beim Klick auf die Schaltfläche neu können folgende Dinge passieren:

1. Liegt in der Hierarchieebene nur ein Leveltyp vor, so wird dieses sofort eingefügt
2. gibt es verschiedene Leveltypen in der Hierarchieebene, in der die Schaltfläche gedrückt wird, so wird ein Auswahldialog zwischengeschaltet, in dem der Benutzer auswählen kann, welcher Leveltyp eingefügt werden soll

Die Funktion **msGetFuncStringToAddNewLevel** liefert den Funktionsstring zurück, was beim OnClick-Event der Schaltfläche ausgeführt wird.

5.3.4 msGetFuncStringToAddNewLevel

Diese Funktion liefert, wie in mAddNewButton angesprochen, das onClickEvent zurück, wenn auf eine Neu-Schaltfläche gedrückt wird.

Sie erwartet folgende Parameter:

- **sParentId**
  Level-ID (z.B. "LV_1_2_"), für das die Neu-Funktion ausgeführt wird
- **sActualLevelName**
  Name des Levels, das eingefügt wird
- **sOptionText**
  Falls ein Optionstext vorliegt, so wird dieser hier angezeigt
- **sOptionValues**
  Falls OptionValues vorliegen, so werden diese hier angezeigt
- **sOptionCaptions**
  Falls OptionCaptions vorliegen, so werden diese hier angezeigt

Wann zeigt die Funktion nun den Auswahldialog an, und wann wird einfach ein Level einfügt?

Falls der Parameter **sOptionValues ungleich „"** ist, so wird der Auswahldialog angezeigt. Ist **sOptionValues null oder „"**, dann wird das angegebene Level sofort eingefügt.