Inhaltsverzeichnis

1. cDataForm.doScatterData....2
2. More onMore 3

1. cDataForm.doScatterData

Besonders in Parent/Child Situationen kann es vorkommen, dass irgendwelche verknüpfte Daten aus dem Parent benötigt werden. Natürlich könnte man den Cursoradapter im Child erweitern oder einen zusätzlichen Cursoradapter für Parentdaten instantiieren. In beiden Fällen kann das ganze unhandlich werden, bzw. ist zum Teil wegen fehlender Fremdschlüssel nicht in allen Fällen möglich. Ausserdem müssen wir einen weiteren Roundtrip zum Server machen. Wieso auch, sind doch die Daten im Parentform immer aktuell vorhanden. Bauen wir uns also eine kleine Brücke:

Eine neue Methode in cDataForm	

doScatterData	Diese Methode erhält als Parameter, der by Reference übergeben werden muss, ein Objekt und füllt darin alle Felder aus dem aktuellen Alias ab:
	
	`LPARAMETERS toVarObject`
	
	`SCATTER MEMO NAME toVarObject`

Im Childform, z.B. in onPostInsert kann nun diese Methode wie folgt aufgerufen werden:

```
LOCAL loPara as Object

loPara = CREATEOBJECT("Empty")
this.oParentform.doscatterdata(@loPara)
```

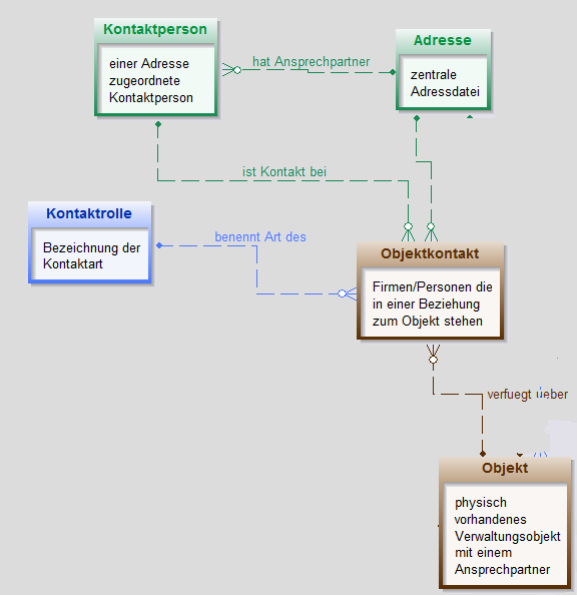
Die Felder des aktuellen Parent Records stehen nun im Objekt loPara zur Verfügung, z.B.:

```
replace <Feld>   WITH loPara.<Feld>, ;
        etc.
```

Selbstverständlich kann ich hier nun z.B. noch mit `PEMSTATUS(loPara,"meinFeld",5)` jede mir genehme Prüfung machen, um Eventualitäten vorzubeugen.

2. More onMore

Mit der onMore Funktion können wir perfekt Childforms aufrufen, in denen im Workalias der Primarykey des Parent, der Foreignkey des Childs ist. In einer etwas komplexeren Situation geht das nicht mehr so einfach:



Die Situation ist die, das wir bei den Objektdaten auf einer Page auch die Objektkontakte anzeigen wollen und, sofern es welche hat, soll der oder die Benutzerin auch auf Knopfdruck die aktuelle Adresse im entsprechenden Formular öffnen können. Nun haben wir folgende Probleme:

- Der Fixfieldvalue ist beim Child nicht der Fremd- sondern der Primärschlüssel.
- Im aufgerufenen Adressformular können wir also keinen Insert zulassen, weil sonst der neu generierte Primärschlüssel mit dem übergebenen Fixfieldvalue überschrieben würde (Konflikt!).
- Abgesehen davon würden wir uns nur zusätzliche Mühen bei der Anzeige der Kontaktadressen aufhalsen.

Wir müssen also gewisse Einstellungen vornehmen, die das Standard onMore nicht kann.

Verpassen wir also unserem cDataForm weitere Methoden:

RequeryCalledOnInit	Diese Methode speichert und restored die Grid-Einstellungen, verpasst dem Cursor Adapter eine neue Where Clause und lädt die Daten neu:
	
	`LPARAMETERS tcWhereClause`
	`* die Whereclause wird im aufrufenden Form`
	`* zusammengestellt, hier muss mind. ein Leerstring ankommen!`
	
	`LOCAL loCARef, loGrid, lcGridSosurce`
	
	`loCARef = GETCURSORADAPTER(this.cWorkalias)`
	`loGrid = this.getGrid()`
	`lcGridSosurce = ""`
	
	`IF TYPE("loGrid") = "O"`
	`WITH logrid`
	`.savestatus()`
	
	`* save RecordSource`
	`lcGridSosurce = .RECORDSOURCE`
	
	`* destroy RecordSource`
	`* preserves Columns`
	`.RECORDSOURCE = ""`
	`ENDWITH`
	`ENDIF`
	
	`* umstellen des CA`
	`this.lWorkonparameterizedca = .T.`
	`loCARef.cWhereClause = tcWhereClause`
	`loCARef.NoData = .F.`
	`loCARef.CursorFill()`
	`this.requery(.F.,.T.)`
	`IF RECCOUNT() = 0`
	`this.onEmptyform()`
	`ENDIF`
	`this.nOldRecno = 0`
	
	`IF TYPE("loGrid") = "O"`
	`WITH logrid`
	`* restore Record Source`
	`.RECORDSOURCE = lcGridSosurce`
	
	`* set Control Sources of the Columns`
	`.restorestatus()`
	`ENDWITH`
	`ENDIF`

GetGrid	Sucht und findet ein allenfalls auf dem Form vorhandenes Grid. Anmerkung: es handelt sich hier um die gleiche Methode wie in der cAskForParas Klasse (siehe sep. Dokumentation). Als Optimierung müsste man die Methode im cDataForm auch aus dem cAskForParas Form aufrufen.

In den beteiligten Formularen kann nun wie folgt verfahren werden:

Im Parentform wird ganz normal die onMore Methode (bzw. der Builder dazu eingesetzt).

Im Childform sind folgende Ergänzungen notwendig:

Init	Wir müssen a) verhindern, dass allenfalls ein cAskForParas Formular aufgerufen wird und nach dem DODEFAULT() müssen wir die Daten noch „händisch“ laden:
	
	`DO CASE`
	`CASE INLIST(this.cCalledBy,"FRMOBJEKT","FRMADRVERKN",`
	`"PICKDIALOG")`
	`  this.cAskForForm  = ""`
	`ENDCASE`
	
	`lInitOk =dodefault(tcArg)`
	
	`**********************************************`
	`** Insert your extra initialization code here`
	`IF lInitOk`
	`  DO CASE`
	`  CASE INLIST(this.cCalledBy,"FRMOBJEKT","FRMADRVERKN",`
	`"PICKDIALOG")`
	`  thisform.RequeryCalledOnInit(this.cCalledBy)`
	
	`  ENDCASE`
	`ENDIF`

RequeryCalledOnInit	Hier bauen wir unsere Where-Clause zusammen und nehmen allfällige weitere Einstellungen vor. Anschliessend wird mit DODEFAULT() das Laden der Daten gestartet:

```
LPARAMETERS tcCalledBy
LOCAL lcWhereClause, llGoOn, lnFormStatus

lcWhereClause = ""
llGoOn        = .F.
lnFormStatus  = thisform.nFormStatus

DO CASE
CASE INLIST(this.cCalledBy,"FRMOBJEKT","FRMADRVERKN")
     PRIVATE tnANummer
     tnANummer = INT(VAL(this.cfixfieldvalue))
     lcWhereClause = "Adresse.nANummer = ?tnANummer"
     llGoOn = .T.
     * in diesem Modus kann nicht erfasst
     * oder kopiert werden
     * (FixfieldValue ergäbe einen identischen Key)
     this.lCanInsert = .F.
     this.lCanCopy = .F.
CASE this.cCalledBy = "PICKDIALOG"
     PRIVATE tnANummer
     IF TYPE("thisform.opickfield") = "O"
        tnANummer = INT(VAL( ;
                    thisform.opickfield.txtField.value))
     ELSE
        tnANummer = 0
     ENDIF
     IF tnANummer > 0
        * dann gleich in den Mutationsmodus
        lnFormStatus = 1
     ENDIF
     lcWhereClause = "Adresse.nANummer  =  ?tnANummer"
     llGoOn = .T.
ENDCASE

IF llGoOn
   DODEFAULT(lcWhereClause)
   IF !thisform.lEmpty AND ;
      thisform.nFormStatus <> lnFormStatus
      thisform.nFormStatus = lnFormStatus
   ENDIF
ENDIF
```

Mit dieser Methode lassen sich nicht nur spezielle Parent/Child Situationen beherrschen, sondern auch der Aufruf des Formulars aus dem Pickdialog (wenn hier beim Cursor Adapter die Daten zuerst nachgeladen werden müssen). Das direkte Umschalten in den Editmodus erfolgte, da nur so konsistente Resultate erzielt werden konnten.