

Inhaltsverzeichnis

1.	Ziel	2
2.	Realisierung	3
2.1.	neue Klassen	3
2.1.1.	cApplSpeedbar	3
2.1.2.	cAskForParas	3
2.2.	ergänzte Klassen	4
2.2.1.	vfxForm.cDataForm	4
2.2.2.	vfxObj.cPickfield	6
3.	Implementation	7
3.1.	Container auf cAskForParas	8
3.2.	Spezialfälle	9
4.	Änderungslog	10
4.1.	Version 1.00.002	10
4.2.	Version 1.00.003	10
4.3.	Version 1.00.004	11
5.	Dateiverzeichnis	12

Version	Überarbeitung	Ersteller	Datum
1.00.000	Neues Dokument	F. Maurhofer	21.11.2007
1.00.001	Ergänzung in Kapitel 3	F. Maurhofer	28.11.2007
1.00.002	Änderungen/Ergänzungen (siehe Log) gem. Diskussion mit Burkhard Anglewitz	F. Maurhofer	03.12.2007
1.00.003	div. Ergänzungen	F. Maurhofer	09.12.2007
1.00.004	Pflege in doComposeWhere Doku ergänzt	F. Maurhofer	08.06.2008

1. Ziel

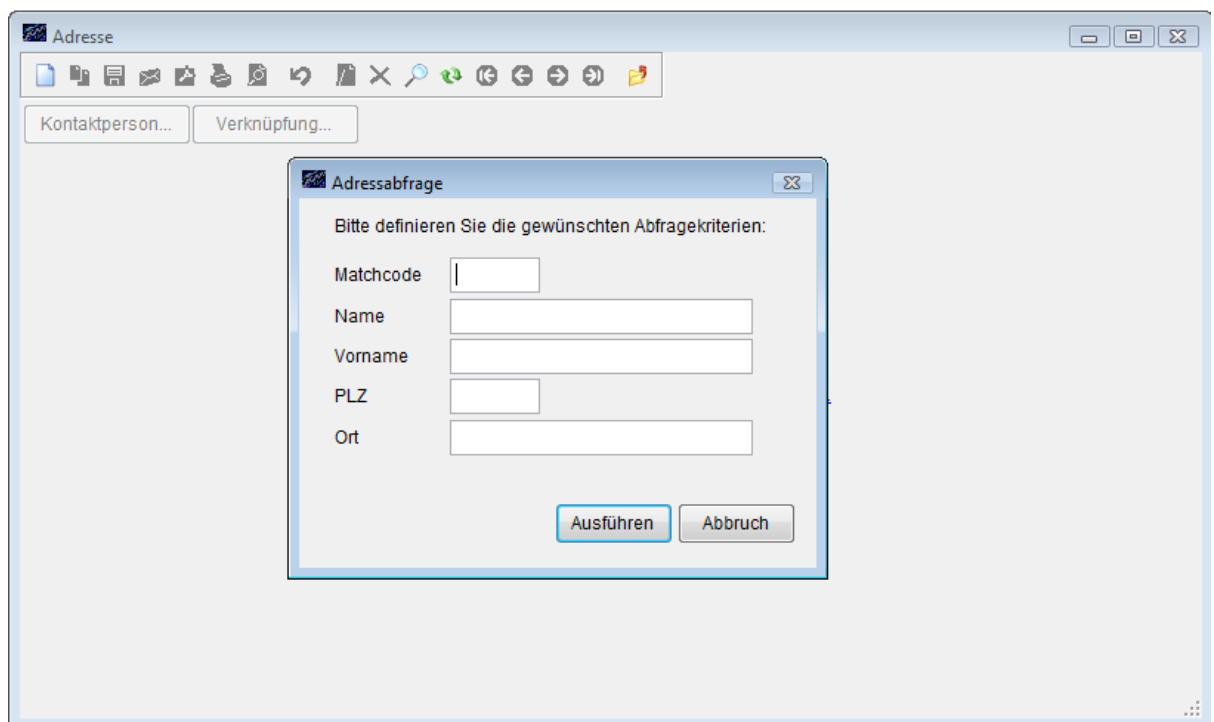
Gesucht wird eine generische Lösung, bei der

- beim Start eines Formulars ein Abfrageform aufpoppt
- die WhereClause des im Datenenvironment vorhandenen Cursoradaptors um die verwendeten Selektionskriterien erweitert wird (keine überflüssigen z.B. like „%“)

Das Abfrageform soll auch beim drücken der F5 Taste (refresh) erscheinen

Das Abfrageform soll nach Bedarf um Kriterienobjekte (z.B. Textboxen) ergänzt werden können, die dann automatisch die Generierung der WhereClause ohne weitere Codeänderungen veranlassen.

Beispiel (beim öffnen eines Forms):



The screenshot shows a window titled 'Adresse' with a toolbar and two buttons: 'Kontaktperson...' and 'Verknüpfung...'. A modal dialog box titled 'Adressabfrage' is open, containing the text 'Bitte definieren Sie die gewünschten Abfragekriterien:'. Below this text are five input fields labeled 'Matchcode', 'Name', 'Vorname', 'PLZ', and 'Ort'. At the bottom of the dialog are two buttons: 'Ausführen' and 'Abbruch'.

Bitte beachten: die Code Snippets sind z.Zt. für den SQL-Server optimiert.

Weitere Anwendung ab Version 1.00.003: das identische Abfrageformular kann nun auch bei Pickfields benützt werden.

2. Realisierung

Soweit möglich wurde das Verhalten und die Funktion der cAskViewArgPgf Klasse nachgebaut (die ja für Remote Views angewendet werden kann).

2.1. neue Klassen

In der Appl.vcx

2.1.1. cAppSpeedbar

Basierend auf cSpeedbar. Geändert wurde nur die Methode cmdRequery.click (gleicher Aufruf wie die entsprechende Menüposition).

→ Wird ggf. überflüssig, wenn die cSpeedbar Klasse entsprechend geändert wird

2.1.2. cAskForParas

Basierend auf cModalForm.

Eigenschaften	
cAskForPrefix	Präfix für die Caption des AskForParas Forms (wird in die Caption des Originalforms ergänzt. Achtung: ist zu gegebener Zeit noch sprachabhängig einzurichten.
cGridSource	Erhält die RecordSource des Grid in oMasterform zugewiesen
cSQLWhere	Enthält die zusammengesetzte WhereClause
cSQLWhereVerbose	Enthält die zusammengesetzte WhereClause für Anzeigezwecke (z.B. im Titel des Forms)
ICalledByPickfield	.T. = das Formular wurde von einem Pickfield aufgerufen .F. = Formularaufruf ab einem Dataform
IFollowMasterform	.T. = Form Position folgt dem Masterform (Default) .F. = Position gem. Formulareinstellung bzw. VFX-Defaults
IGetAllData	.T. = bei leerer WhereClause werden alle Daten geholt .F. = es erfolgt keine Abfrage (Default)
nDatabaseServerType	Hier wird aufgrund des Connection Managers festgehalten, mit welcher Datenbankart wir es aktuell zu tun haben. Damit können im doComposeWhere SQL-spezifische Unterschiede abgehandelt werden. Konkret wird dies z.Zt. für den Aufbau von DatumsZeit- und logischen Werten benutzt. 1 = FoxPro 2 = SQL Server 3 = Oracle 4 = IBM DB2
oMasterForm	Referenz auf das aufrufende Formular

Methoden	
doComposeWhere	Setzt die Where Clause zusammen Externer Funktionsaufruf: fmSQLDate (Datum/Zeitwerte gem. SQL-Server formatieren). Logische Werte werden z.Zt. mit 0 und 1 (für SQL-Server) , bzw. .T. und .F. (VFP), umgesetzt.
getGrid	Referenz auf Grid-Objekt im Masterform ermitteln
getUserWhereCondition	Diese Methode wird im doComposeWhere nach der Generierung der Where Clause aufgerufen. Hier können nun sämtliche Spezialfälle einer spezifischen Implementierung des Forms abgehandelt und damit die Where Clause ergänzt werden.
Requery	Daten im InitialSelectedAlias im oMasterForm gem. neuer cSQLWhere ermitteln: Grid sichern Requery der Daten Caption des oMasterForm's gem. cSQLWhereVerbose anpassen Grid restoren
restoreGridData	Grid in oMasterForm restoren
saveGridDate	Grid in oMasterForm sichern
setoMasterFormCaption	Caption in oMasterForm setzen

2.2. ergänzte Klassen

2.2.1. vfxForm.cDataForm

Eigenschaften	
cAskForForm	Enthält den Namen des Parameter Forms, das im Refresh Modus aufzurufen ist.
cSQLWhere	Enthält die zusammengesetzte WhereClause
cSQLWhereVerbose	Enthält die zusammengesetzte WhereClause für Anzeigezwecke (z.B. im Titel des Forms)

Methoden	
Init	<p>Sofern die Eigenschaft cAskForForm einen Wert enthält, wollen wir das betreffende Abfrageformular beim Programmstart anzeigen. Daher ist hier folgender Code einzutragen:</p> <pre> LPARAMETERS tcarg LOCAL llRetVal llRetVal = .T. IF DODEFAULT(tcArg) IF PEMSTATUS(thisform,"cAskForForm",5) IF NOT EMPTY(thisform.cAskForForm) * wir wollen das Form sehen, wenn * die Abfrage kommt IF NOT thisform.Visible thisform.Visible = .T. ENDIF DO FORM (thisform.cAskForForm) WITH thisform ENDIF ENDIF ELSE llRetVal = .F. ENDIF RETURN llRetVal </pre>
onRefresh	<p>Menü Refresh und die Refresh Schaltfläche im Speedbar (cAppSpeedBar) rufen diese Methode auf. Folgender Code ist daher einzutragen:</p> <pre> LPARAMETER tlloaddata, tlrefreshonly * modifiziertes Refresh() LOCAL lcAskForForm lcAskForForm = "" IF PEMSTATUS(thisform,"cAskForForm",5) lcAskForForm = thisform.cAskForForm ENDIF IF TYPE("ThisForm.lCanRequery") == "L" ; AND !ISNULL(ThisForm.lCanRequery) IF PEMSTATUS(ThisForm, "Requery" ,5) IF EMPTY(lcAskForForm) ThisForm.REQUERY(.F.,.T.) ELSE DO FORM (thisform.cAskForForm) WITH thisform ENDIF ENDIF ELSE * wenn's gar nichts ist, dann halt das bisherige DODEFAULT(tlloaddata, tlrefreshonly) ENDIF </pre>

2.2.2. vfxObj.cPickfield

Eigenschaften	
cAskForForm	Enthält den Namen des Parameter Forms, das vor dem dem Start des Pickforms aufgerufen wird. Leer = Verhalten wie bisher (Default)
lClearWithNull	.T. = die Clear Methode leert das txtField.Value mit .NULL. (siehe auch Clear-Methode)
Events	
cmdPick.click	<p>Neuer Code:</p> <pre> LOCAL loCAREf, lcWhere lcWhere = "" IF NOT EMPTY(this.parent.cAskforForm) DO FORM (this.parent.cAskforForm) WITH thisform, this.Parent TO lcWhere IF NOT EMPTY(lcWhere) loCAREf = GETCURSORADAPTER(this.Parent.ctablename) loCAREf.cPickWhereClause = lcWhere ENDIF DODEFAULT() IF TYPE("loCAREf") = "O" loCAREf.cPickWhereClause = "" ENDIF ELSE DODEFAULT() ENDIF </pre>
Methoden	
Clear	<p>In gewissen Situationen ist es notwendig, den Wert in txtField mit .NULL. zu initialisieren, statt dem VFP-Standardwert (mit GetEmptyVariableValue). Zusammen mit der Eigenschaft lClearWithNull wird dem hier Rechnung getragen:</p> <pre> DODEFAULT() IF this.lClearWithNull this.txtField.Value = .NULL. ENDIF </pre>

3. *Implementation*

Basierend auf der Klasse cAskForParas wird ein Formular erstellt, auf dem pro möglichem Abfrageparameter ein Eingabeobjekt vorhanden ist. Hier am Beispiel des Formulars AskAdresse:

Bei jeder Textbox (oder anderem Eingabeobjekt wie Checkbox, Radiobutton) werden folgende Eigenschaften mit Werten versehen:

Tag	Name des Kriteriums der für cSQLWhereVerbose, also die Titelanzeige im Formular verwendet wird (entspricht in der Regel der Caption des korrespondierenden Labels)
cViewParameter	Name des Datenbankfeldes, das mit diesem Kriterium abgefragt wird. Achtung: Da dieser Wert in der WhereClause verwendet wird, ist der Originalname der Tabelle und nicht etwa der vom CursorAdaptor benützte Alias Name zu verwenden. Also z.B. Adresse.cName und nicht caAdresse.cName
ExtraBuffer	Zu verwendender Operand. Zulässig ist jeder im SQL-Statement benützbare Operand also like, >, < etc. Die Operaden = und == sind wie folgt einzutragen: "=" oder "==".
Formular DataSession	Wir benützen die Datasession des oMasterForm, daher: 1 – Default Data Session

Die Methode doComposeWhere scant nun alle Objekte und verarbeitet diejenigen, bei denen im cViewParameter ein Wert enthalten ist. Sofern die Eigenschaft IGetAllData den Wert .T. hat, werden bei einem „leeren“ Abfrageformular (= keine Kriterien definiert) nun alle Daten geholt.

Im Masterform (also dem effektiven Datenformular, das auf einer Subklasse des cDataForm basieren kann) sind folgende Einstellungen nötig:

cAskForForm	Name des Parameterforms eintragen (im Falle dieses Beispiels AskAdresse)
DataEnvironment CursorAdaptor.NoData	Auf .T. stellen, damit im Load keine Daten geladen werden, sondern erst, wenn im Parameterform mindestens ein Kriterium definiert ist. Anmerkung: Alle Daten können geladen werden, wenn bei einem „like“ Kriterium ein %-Zeichen eingetragen wird.
Speedbar	Hier ist die Klasse von cSpeedbar auf cApplSpeedbar zu ändern (wegen Refresh Verhalten)

Bei der Verwendung im Pickfield ist folgendes zu beachten:

- In der Datenumgebung des Formulars mit dem Pickfield ist der Cursoradaptor gemäss Pickfield.cTablename aufzunehmen. Eigenschaft NoDate = .T.
- cAskForForm erhält den Namen des Parameterforms (AskAdresse gem. obigem Beispiel)
- IHideCode = .T. (nur in dieser Form getestet)

Mit dem Click auf die Schaltfläche oder mit F9 wird die Auswahl gestartet und vor dem Pickform wird zuerst das Parameterform angezeigt. Sofern mindestens ein Abfragekriterium definiert wurde, wird der Eigenschaft cPickWhereClause des Cursoradaptors gemäss cTableName die ermittelte Where-Bedingung zugewiesen. Der Rest wird vom Pickfield, resp. dem Cursoradapter automatisch erledigt.

Unterschied zur Anwendung im Dataform: werden beim Dataformeinsatz keine Daten gefunden, wird eine Meldung ausgegeben. Im Falle des Pickfields wird das leere Pickform angezeigt.

3.1. Container auf cAskForParas

Die Verwendung von Container Objekten (z.B. Pickfield) ist auf dem Abfrageformular ebenfalls möglich. Dabei ist folgendes zu beachten:

Auf dem Containerobjekt werden die Werte analog der übrigen Objekte eingetragen, also bei den Eigenschaften

- Tag
- cViewParameter
- ExtraBuffer

Innerhalb des Containers erhält das Objekt mit dem zu verwendenden Value in der Eigenschaft cViewParameter den gleichen Wert wie bei der Container Eigenschaft. Im Falle eines Pickfields erhält also das txtField.cViewParameter den Wert gemäss Container.

3.2. Spezialfälle

Sofern das Abfrageformular Objekte oder Kombinationen enthält, die mit den vorhandenen Mitteln nicht in eine korrekte Where Clause aufgelöst werden können, kann die Methode **getUserWhereCondition** verwendet werden. Sie wird unmittelbar nach erstellen des Wertes für cSQLWhere und cSQLWhereVerbose aufgerufen. Hier können nun diese Werte nach Wunsch ergänzt oder manipuliert werden. Beispiel anhand eines Datumsbereiches:

Der Datumsbereich ist ein Container (Standardklasse, in dem z.B. die Abhängigkeiten des von/bis Datums geprüft werden), der seinerseits wiederum Container enthält (Datums-Textfelder). Das ganze wird nun wie folgt implementiert:

Der Container und seine Elemente enthalten keinerlei Angaben in Tag, cViewparameter und Extrabuffer. In der Methode getUserWhereCondition wird nun der Code aufbereitet, der dann automatisch an die Where-Clause angehängt wird:

```

LOCAL ldDateVon, ldDateBis, lcDateWhere, lcDateVerbose

ldDateVon      = ThisForm.cntdAufDatum.txtRXDatVon.txtDate.Value
ldDateBis      = ThisForm.cntdAufDatum.txtRXDatBis.txtDate.Value
lcDateWhere    = ""
lcDateVerbose  = ""

DO CASE
CASE NOT EMPTY(NVL(ldDateVon,{})) AND NOT EMPTY(NVL(ldDateBis,{}))
  * Bereich
  lcDateWhere   = "Faktura.dRgDatum BETWEEN " +
                  fmSQLDate(ldDateVon) + " AND " +
                  fmSQLDate(ldDateBis)
  lcDateVerbose = "Datum: " + DTOC(ldDateVon) + " - " + DTOC(ldDateBis)
CASE NOT EMPTY(NVL(ldDateVon,{}))
  * ab
  lcDateWhere   = "Faktura.dRgDatum >= " + fmSQLDate(ldDateVon)
  lcDateVerbose = "Datum: ab " + DTOC(ldDateVon)
CASE NOT EMPTY(NVL(ldDateBis,{}))
  * bis
  lcDateWhere   = "Faktura.dRgDatum <= " + fmSQLDate(ldDateBis)
  lcDateVerbose = "Datum: bis " + DTOC(ldDateBis)
ENDCASE

IF NOT EMPTY(lcDateWhere)
  IF NOT EMPTY(this.cSQLWhere)
    this.cSQLWhere = this.cSQLWhere + " AND "
  ENDIF
  this.cSQLWhere = this.cSQLWhere + lcDateWhere
  this.cSQLWhereVerbose = this.cSQLWhereVerbose + " " + lcDateVerbose
ENDIF

```

4. **Änderungslog**

4.1. **Version 1.00.002**

cAskForparas

Requery

WhereClause wird nun zurückgestellt, wenn keine Daten gefunden worden sind.

IGetAllData neues Property

.T. = bei leerer WhereClause werden nun alle Daten geholt
.F. (Default) bisheriges Verhalten: es erfolgt keine Abfrage

IFollowMasterform neues Property

.T. = Form Position folgt dem Masterform (Default)
.F. = Position gem. Formulareinstellung bzw. VFX-Defaults

Init

Angepasst an neues Verhalten, insbesondere wird kann nun die Formularposition und die Form Caption automatisch an das Masterform gekoppelt werden.

coMasterFormCaption

Neue Eigenschaft, deren Wert im Init ermittelt wird (das Resultat wird nun an zwei Orten verwendet)

cAskForPrefix neues Property

Präfix für die Caption des AskForParas Forms (wird in die Caption des Originalforms ergänzt.

Achtung: ist zu gegebener Zeit noch sprachabhängig einzurichten.

getGrid

Liste der Klassennamen erweitert

SetoMasterFormCaption

Umgebaut gemäss neuen Gegebenheiten

4.2. **Version 1.00.003**

cAskForparas

In den Methoden **Init**, **Requery** und **SetoMasterFormCaption** wurden Anpassungen wegen des Einsatzes bei Pickfields vorgenommen.

Neue Methode **getUserWhereCondition**

Diese Methode wird im doComposeWhere nach Fertigstellung der Where Clause aufgerufen. In einer spezifischen Ableitung des Formulars können hier nun sämtliche Spezialfälle abgehandelt und die Where Clause ergänzt werden.

nDatabaseServerType neue Eigenschaft

Hier wird aufgrund des Connection Managers festgehalten, mit welcher Datenbankart wir es aktuell zu tun haben. Damit können im doComposeWhere SQL-spezifische Unterschiede abgehandelt werden. Konkret wird dies z.Zt. für den Aufbau von DatumsZeit-Werten benützt.

Container Objekte (z.B. Pickfields) werden nun ebenfalls automatisch abgehandelt. Bitte die Hinweise in Kapitel 3.1.

vfxObj.cPickfield

cAskforForm neue Eigenschaft für Parameterform
(Default = leer)

cmdPick.Click

Aufruf des Parameterforms gemäss cAskforForm und übergabe der WhereClause an den CursorAdapter

4.3. Version 1.00.004

cAskForparas

Methode doComposeWhere

- Leere Pickfieldvalues berücksichtigen
- Logische Felder gem. VFP und SQL-Server unterscheiden

5. *Dateiverzeichnis*

Zu diesem Thema werden folgende Dateien mitgeliefert:

Lib	Neue Klassen: Appl_trans cApplspeedbar und cAskForParas Ergänzte Klassen (ersetzen, bzw. ergänzen die Originale in vfxForm, resp. vfxObj): vfxForm_trans cDataform vfxObj_trans cPickfield
Programm	fmSQLDate.prg und fmVFPSQLDate.prg
Text	Dieses Dokument (PDF)