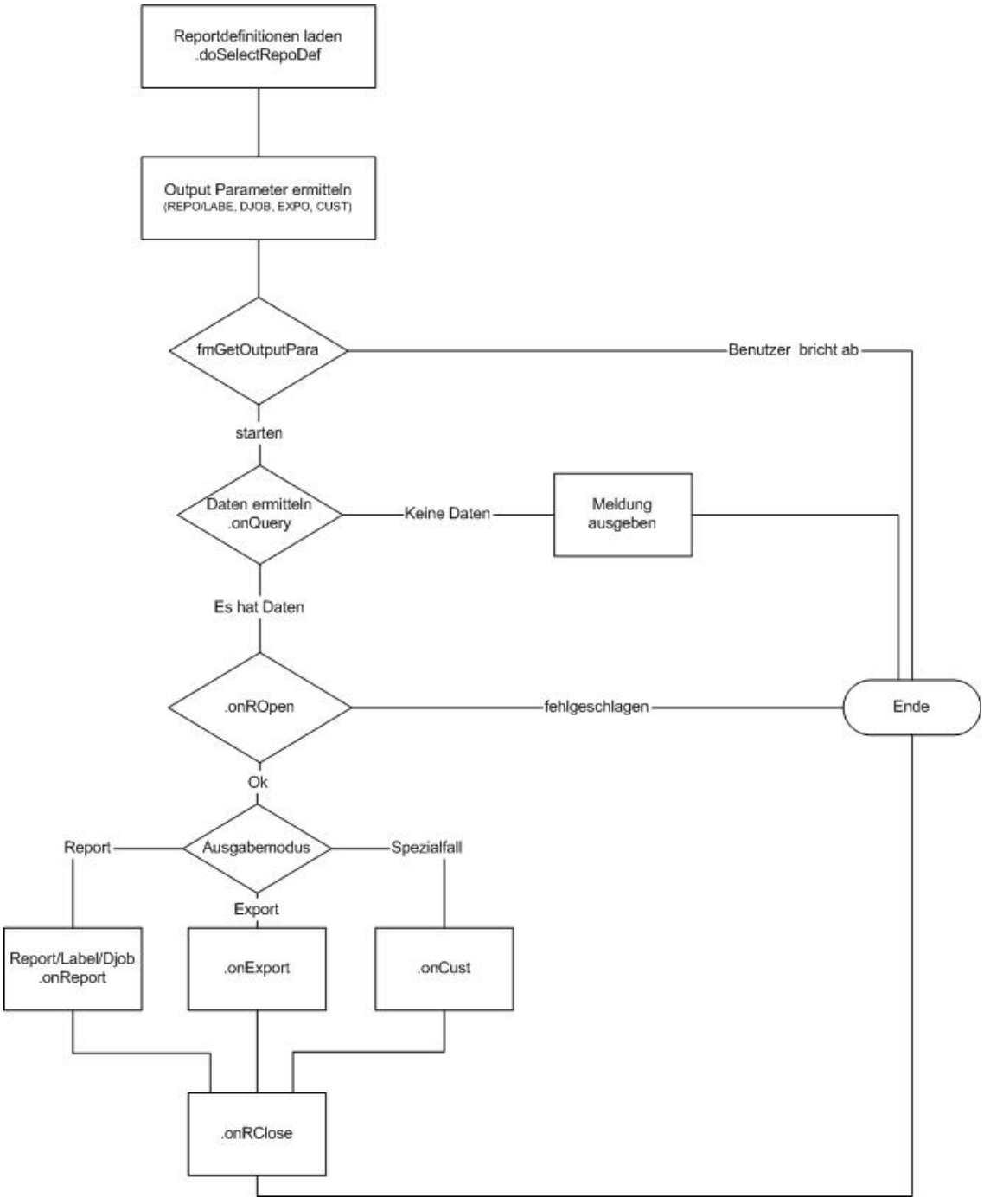
Inhalt

Dieses Dokument enthält die wichtigsten Funktionen der Reportsteuerung, die anlässlich des VFX Usertreffens vom 19. Mai 2006 in Frankfurt vorgestellt worden ist.

Author: Fritz Maurhofer, Maurhofer Informatik AG
murmi@maurhofer-informatik.ch
www.maurhofer-informatik.ch

1. Ablauf onReport



2. Methodencode

2.1. onPrint

OnPrint ist eine Methode der Klasse aRepofunc. Diese wird zur Laufzeit zu einem Container hinzugefügt:

```
this.AddObject("oRepoFunc","aRepoFunc", thisform)
```

Die onPrint Methode wird von den Vorschau/Druck Schaltflächen oder Menüpunkten aufgerufen und hier findet die gesamte Steuerung statt.

```
PARAMETERS tlPreview, tlPrompt, toForm
LOCAL lcRepoDef, lcInClause, llGoOn, lcMsg, lnFlag, loOutPut, ;
      lnOutputTo, loErr, lcSetEscape, lcOnEscape, lnIconVal, ;
       lnCurrSele

llGoOn    = .T.
lcMsg     = ""   && allfällige Meldungen wenn llGoOn = .F.
lnIconVal = 48   && Standard-Icon bei Meldungen
lnCurrSele = SELECT()

* Aufruf für Nachwelt sichern
* damit, z.B. im onCust entsprechend reagiert werden kann
this.lPreview = tlPreview
* tlPrompt ist nur noch aus Kompatibilitätsgründen drin
* es wird jedesmal der Druckprompt angezeigt
* Defaults für Ausgabe
* (wird ggf. später übersteuert)
IF this.lPreview
   this.nOutputTo = 1
ELSE
   this.nOutputTo = 2
ENDIF

PRIVATE loRXReport
* Referenz auf aktuelles Objekt
loRXReport = toForm

* Druckjobs laden
* Cursor mit Formulardefinition(en)
lcRepoDef = "RepoDef" + PADL(reports.nrepoid,4,"0")
IF USED(lcRepoDef)
   * alles Paletti, ist bereits von einem vorherigen
   * Druckjob geladen
ELSE
   * Definitionen laden
   lcInClause = " ("
   IF Reports.cTyp = "DJOB"
      lcInClause = lcInClause + ALLTRIM(Reports.cSelect)
   ELSE
      lcInClause = lcInClause + ALLTRIM(STR(Reports.nRepoID))
   ENDIF
   lcInClause = lcInClause + ") "
   IF this.doSelectrepodef(lcRepoDef, lcInClause, @lcMsg) <= 0
      llGoOn = .F.
   ENDIF
ENDIF
```

```
IF llGoOn
   * wir ermitteln die Output-Parameter
   lnFlag = 0
   lnOutputto = this.nOutputTo
   DO CASE
   CASE INLIST(Reports.cTYP,"REPO","LABE")
        * Es sind alle Optionen möglich
        lnFlag = 0
   CASE Reports.cTyp = "DJOB"
        * direkte Anzeige ist nicht möglich
        lnFlag = 0 + 4 + 8
        * lnFlag = 1 + 4 + 8
        lnOutPutTo = 2
   CASE Reports.cTyp = "EXPO"
        * Anzeige und Export sind möglich
        lnFlag = 2
        lnOutPutto = 3
   CASE Reports.cTyp = "CUST"
        * Anzeige und Druck sind möglich
        lnFlag = 1
   ENDCASE
   TRY
      IF "MAIL"$Reports.mfrx
         * Mailversand = ohne Druckerabfrage
         llGoOn = .T.
      ELSE
         * nun kommt der Benutzer dran...
         loOutPut = CREATEOBJECT("fmGetOutputPara",lcRepoDef,;
                    this.cexportpfad, lnOutPutTo, this.cOutPutType,lnFlag)
         loOutPut.show(1)
         llGoOn = loOutPut.uRetcode
         IF llGoOn
            * nun noch die Werte übernehmen, die nicht in
            * den Druckjob Records sind
            WITH loOutPut
               this.nOutputTo      = .nOutPutTo
               IF this.nOutputTo = 1
                  this.lPreview = .T.
               ENDIF
               this.cExportDatei   = .cExportDatei
               IF !EMPTY(this.cExportDatei)
                  * wir sichern den zuletzt gewählten Pfad
                  this.cExportPfad = JUSTPATH(this.cExportDatei)
               ENDIF
               this.cOutputType    = .cOutputType
               this.lOpenFile      = .lOpenFile
               * Druckart
               this.nPrintRangeModus = .nPrintRangeModus
               * Range
               this.nPageVon = .nPageVon
               this.nPageBis = .nPageBis
            ENDWITH
         ENDIF
         loOutPut.release()
         RELEASE loOutPut
      ENDIF "MAIL"$Reports.mfrx
   CATCH TO loErr
      llGoOn = .F.
      lcMsg = "Fehler beim bestimmen der Ausgabe-Eigenschaften: " + ID_CRLF
+;
```

```
                           ALLTRIM(STR(loErr.ErrorNo)) + "/" +;
                           loErr.Message
   FINALLY
      RELEASE loOutPut
   ENDTRY
ENDIF
SELECT (lnCurrSele)
IF llGoOn
   * wir haben nun die Ausgabeparamter zusammen, es kann also losgehen
   * die Methoden onQuery, onROpen, onRClose, onRepCust und, bei Bedarf,
   * onExportExt müssen im Host-Formular vorhanden sein!
   IF toForm.onQuery()
      IF toForm.onROpen()
         * wir haben Daten und alles ist soweit bereit

         * Preview-Fenster definieren
         IF this.lPreview
            DEFINE WINDOW _preview FROM 0, 0 TO 10, 10 NAME loPreview ;
                   TITLE cap_preview + ": " +
ALLTRIM(NVL(Reports.cBezeich,"")) + ;
                         " (" + ALLTRIM(Reports.cDosName) + ")" ;
                   IN SCREEN CLOSE GROW SYSTEM FLOAT FONT "Arial" , 10
            ZOOM WINDOW _preview MAX
         ENDIF

         * Abbruchmöglichkeit einführen
         lcSetEscape = SET("ESCAPE")
         lcOnEscape = ON("ESCAPE")
         this.lPrintCancel = .F.
         SET ESCAPE ON
         * PRIVATE, damit diese in onAbbruch umgestellt werden kann
         PRIVATE llPrintOn
         llPrintOn = .T.
         ON ESCAPE this.onAbbruch()

         DO CASE
         CASE INLIST(Reports.cTyp,"REPO","LABE","DJOB")
              * Vorschau, Druck oder XFRX
              lcMsg = this.onReport(lcRepoDef)

         CASE Reports.cTyp = "EXPO"
              * Copy to Export
              this.onExport(toForm)

         CASE Reports.cTyp = "CUST"
              * Cust wird an den Container delegiert
              IF PEMSTATUS(toForm,"ONREPCUST",5)
                 toForm.onRepCust()
              ELSE
                 lcMsg = "Diese Berichtsfunktion (CUST) wird im" + ID_CRLF
+;
                         "aktuellen Formular nicht unterstützt!"
              ENDIF
         ENDCASE

         * aufräumen
         SET ESCAPE &lcSetEscape
         ON ESCAPE &lcOnEscape
         * festhalten für onRClose, ob der
         * Druck abgebrochen worden ist oder nicht
```

```
            this.lPrintCancel = !llPrintOn

            toForm.onRClose()

            IF this.lPreview
               RELEASE WINDOW _preview
            ENDIF
         ENDIF
      ELSE
         * es hat keine Daten
         toForm.onRClose()
         lcMsg = "Keine Daten gemäss Ihrer Abfrage"
         lnIconVal = 64
      ENDIF toForm.onQuery()
ENDIF

IF !EMPTY(lcMsg) AND this.lShowMsg
   MESSAGEBOX(lcMsg,0+lnIconVal,"Berichtsausgabe: " + toForm.caption)
ENDIF

IF USED("Reports")
   * definitiv wieder in diesen Bereich zurück
   SELECT Reports
ENDIF
```

2.2. doSelectrepodef

Hier werden die Berichtsdefinitionen gemäss den in der Tabelle Reports gewählten Record aufbereitet (in der Regel 1 Bericht/Export, es können aber mehrere Reports, ein sogenannter Printjob, sein):

```
LPARAMETERS tcCursorName, tcInClause, tcMsg
LOCAL lnCurrSele, lnAnzDef, lnRetVal, loErr, lnAnzCursors
lnCurrSele = SELECT()
lnRetVal = 0
lnAnzCursors = 0

TRY
      SELECT Reports.cbezeich, Reports.nrepoid, ;
             Reports.cdosname, Reports.ctyp,
             Reports.cselect, Reports.msbtext, Reports.lsbdokwahl,;
             Reports.matvorl,Reports.cdefprinter, Reports.ndeftray,;
             Reports.ncolor, Reports.ncopies,;
             Repoform.nloffset, Repoform.ntoffset, Repoform.nboffset,;
             Repoform.nporient, Repoform.npapersize, ;
             Repoform.moptions, 0000 AS nOrder;
       FROM ;
           commando!reports ;
          LEFT OUTER JOIN commando!repoform ;
         ON  Reports.cdosname = Repoform.cdosname ;
         AND  Reports.ctyp = Repoform.ctyp ;
       WHERE  Reports.nrepoid IN &tcInClause ;
       INTO CURSOR (tcCursorName) readwrite

      lcOrder  = STRTRAN(STRTRAN(tcInClause,"(",""),")","")
      lnAnzDef = getArgCount(lcOrder,",")
      IF lnAnzDef > 1
         FOR i = 1 TO lnAnzDef
             lnRepoID = INT(VAL(getArg(lcOrder,i,",")))
             LOCATE FOR nRepoID = lnRepoID
             IF FOUND()
                replace nOrder WITH i
             ENDIF
         ENDFOR
         INDEX ON nOrder TAG nOrder
      ENDIF
      IF USED(tcCursorName)
         LOCATE
       lnRetVal = RECCOUNT(tcCursorName)
    ELSE
       lnRetVal = 0
    ENDIF
    * alle Jobs, die noch keinen Default-Printer haben
    * erhalten jetzt die Daten des aktuell gültigen
    SCAN
       IF EMPTY(cDefPrinter)
          REPLACE cDefPrinter WITH SET("Printer",3)
       ENDIF
       IF EMPTY(ndeftray)
          replace nDefTray WITH PRTINFO(7)
       ENDIF
    ENDSCAN
    LOCATE
```

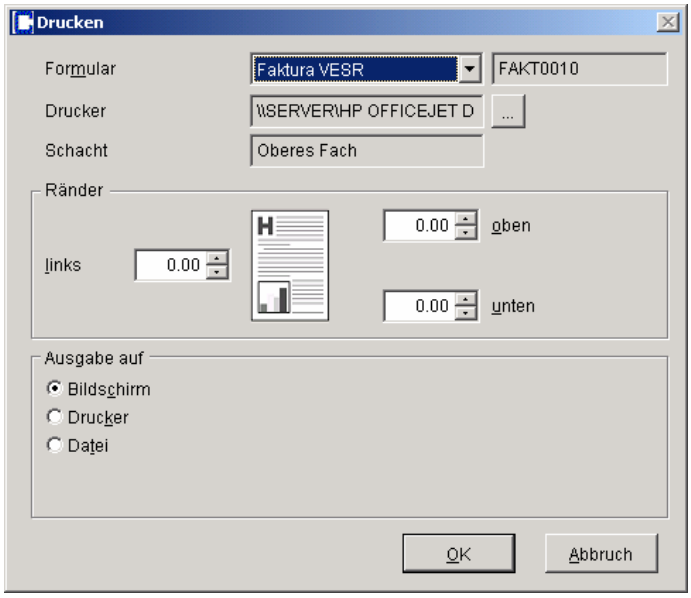
```
CATCH TO loErr
    lnRetVal = -1
    * Variable für Fehlermeldung wurde by Ref. übergeben
    tcMsg = "Reportdefinitionen konnten nicht geladen werden" + ID_CRLF +;
            "bitte Berichts- und Formulardefinitionen überprüfen!" ;
            + ID_CRLF +;
            ALLTRIM(STR(loErr.ErrorNo)) + "/" + loErr.Message

FINALLY
    IF EMPTY(this.aDefCursors[1])
       lnAnzCursors = 0
    ELSE
       lnAnzCursors = ALEN(this.aDefCursors,1)
    ENDIF
    IF lnRetVal > 0
       IF ASCAN(this.aDefCursors,tcCursorName,-1,-1,-1,15) > 0
          * dann gibt's nichts zu tun, Cursor ist bereits in der Liste
       ELSE
          * damit am Schluss korrekt geschlossen wir,
          * Cursor in Liste aufnehmen
          DIMENSION this.aDefCursors[lnAnzCursors + 1]
          this.aDefCursors[lnAnzCursors + 1] = tcCursorName
       ENDIF
    ENDIF
ENDTRY

RETURN lnRetVal
```

2.3. fmGetOutputPara

Diese Klasse dient der Kommunikation mit dem Benutzer. Hier können die Default Einstellungen ggf. übersteuert werden, ebenso welcher Art der Output ist:



2.4. onReport

Steuerung der Druckausgabe von Reports (Drucker oder XFRX-Export).

```
LPARAMETERS tcRepoDef
LOCAL llGoOn, lnCurrRec, lnCurrSele, lcLoopCond, lcLoopStore, ;
      lcWhile, lcRange, loSession, lnResult, lcRetVal, lcPrintModus,;
      lnRepoID, lcRepoOrig, lcRepoExt, ;
      lnDataRec, lcOldPrintDef, lcDefPrinter, lcRepoWhile, llISDJOB

llGoOn       = .T.
lcRetVal     = ""
lcWhile      = ""
lcRepoWhile  = ""
lcRange      = ""
lcPrintModus = ""
llPrintOn    = .T.  && Private Var aus onPrint, wird im ON ESCAPE Fall
umgestellt
* aktuellen Datensatz in Reports festhalten
lnCurrRec    = RECNO("Reports")
lnCurrSele   = SELECT()  && wir sind aktuell im Resultat-Cursor
* wir benötigen eine zusätzliche Kennung, um ggf.
* nach dem ersten Berichtsdurchlauf abbrechen zu können
* (ohne das Abbruch-Property zu setzen
IF Reports.cTyp = "DJOB"
   llISDJOB     = .T.
ELSE
   llISDJOB     = .F.
ENDIF

* Grundeinstellungen
DO CASE
CASE this.nOutputto = 1
     * Vorschau (nur bei einzelnen Berichten
     lcPrintModus = " PREVIEW WINDOW _preview "
CASE this.nOutputto = 2
     * normale Berichte
     IF this.nPrintRangeModus = 2
        lcRange = "RANGE " + ;
                  ALLTRIM(STR(THIS.nPageVon)) + ", " +;
                  ALLTRIM(STR(THIS.nPageBis)) + " "
     ENDIF
     lcPrintModus = " TO PRINT NOCONSOLE "

CASE this.nOutputto = 3
     loSession = EVALUATE("xfrx('xfrx#init')")
     lnResult = loSession.setparams(this.cExportDatei,SYS(2023),;
                !this.lOpenFile,,,,this.cOutputType)
     IF lnResult < 0
        * xfrx-Session kann nicht eröffnet werden
        lcRetVal = "XFRX-Session-Fehler: " + ALLTRIM(STR(lnResult))
        llGoOn = .F.
     ELSE
        * Session ist i.O., Dokumenteigenschaften einstellen
        loSession.setAuthor(ALLTRIM(gu_user_name))
        loSession.setTitle(ALLTRIM(Reports.cBezeich))
        loSession.setSubject(goSystem.execute("ALLTRIM(cKundinf1)"))
     ENDIF
ENDCASE
```

```
IF llGoOn
   * Loop über die Daten
   * While Bedingung festlegen
   * bei einem Druckjob enthält FRT die
   * Schlaufenbedingung für einen einzelen Bericht
   * wird hier mit aufbereitet, damit innerhalb des
   * Loops mit EVAL die jeweils gültige Schleifenbedingung
   * generiert werden kann
   IF Reports.cTyp = "DJOB" AND !EMPTY(Reports.mFRT)
      lcRepoWhile = ALLTRIM(Reports.mFRT)
      * in einem Druckjob gibt es keinen Range
      lcRange = ""
   ENDIF
   IF this.nOutputto = 2
      * beim Report noch das Standardwhile
      IF EMPTY(lcRepoWhile)
         IF EMPTY(lcRange)
            *! nur wenn kein Range definiert worden ist
            *! dann kann es ein While haben
            *! (VFP Verhalten....)
            lcRepoWhile = "' WHILE llPrinton '"
         ENDIF
      ELSE
         lcRepoWhile = "'WHILE ' + " + lcRepoWhile + " +' AND llPrinton '"
      ENDIF
   ENDIF
   SELECT (lnCurrSele)
   * wir stehen am Anfang
   DO WHILE !EOF() AND llGoOn
      * aktuellen Record sichern
      lnDataRec = RECNO()
      * while Bedingung jetzt definitiv für jeden Durchgang aufbauen
      IF !EMPTY(lcRepoWhile)
         lcWhile = EVALUATE(lcRepoWhile)
      ENDIF
      * zu den Bereichtsdefininitionen wechseln
      SELECT (tcRepoDef)
      LOCATE
      DO WHILE !EOF() AND llGoOn
         * solange wir in diesem Loop sind, beginnt
         * jeder Job beim oben Definierten Record
         SELECT (lnCurrSele)
         GOTO lnDataRec
         lnRepoID = EVALUATE(tcRepoDef + ".nRepoID")
         SELECT Reports
         * damit alles bisherige korrekt funktioniert
         * muss auch in Druckjobs der aktuelle Bericht gewählt sein
         LOCATE FOR nRepoID = lnRepoID
         * nun die Verarbeitung
         SELECT (tcRepoDef)
         * Bericht
         lcRepoExt  = IIF(cTyp = "LABE",".lbx",".frx")
         lcRepoOrig = ALLTRIM(cDosName) + lcRepoExt
         lcRepoNeu  = "'" + SYS(2023) + "\" + SYS(2015) + lcRepoExt + "'"
         * aktuellen Drucker sichern
         lcOldPrintDef = SET("Printer",3)
         lcDefPrinter = ALLTRIM(cDefPrinter)
         IF !EMPTY(lcDefPrinter)
            SET PRINTER TO NAME (lcDefPrinter)
```

```
ENDIF
* Report bereitstellen
IF this.onPrepareReport(lcRepoOrig, lcRepoNeu, tcRepoDef,;
                        @lcRetVal)
   * Report verarbeiten
   SELECT (lnCurrSele)
   IF this.nOutPutTo = 3
      * XFRX
      loSession.ProcessReport(EVALUATE(lcRepoNeu),,,,lcWhile)
      *{ 13.06.2005 10:19:39 murmi
      * Druckjobs bei Export
      * zwischen den Reports müssen die Seitenzahlen
      * resettet werden
      loSession.ResetPageNo()
      *} 13.06.2005 10:19:39 murmi
   ELSE
      * Standard Report
      REPORT FORM &lcRepoNeu &lcWhile &lcRange &lcPrintModus
   ENDIF
   * Report löschen
   lcDeleFile = STRTRAN(lcRepoNeu,lcRepoExt,".*")
   ERASE &lcDeleFile
ELSE
   llGoOn = .F.
ENDIF
* Drucker wieder zurück
SET PRINTER TO NAME (lcOldPrintDef)

SELECT (tcRepoDef)
SKIP
* wichtig: der nachfolgende Mechanismus funktioniert bei
* Druckjobs nur wenn Sie exportiert oder gedruckt werden
* (Preview ist eigentlich ausgeschaltet bei DJOB)
IF EOF()
   * alle Defintionen abgearbeitet
   * wir beginnen einen neuen Range
   SELECT (lnCurrSele)
   *{ 13.06.2005 10:56:27 murmi
   * wenn bei Druckjobs:
   * bei XFRX bleibt der Record-Pointer auf dem letzten Satz des
   * Records gem. WHILE-Bedingung, daher hier einen zusätzlichen
   * SKIP
   IF this.nOutPutTo = 3 AND llISDJOB
      SKIP
   ENDIF
   *} 13.06.2005 10:56:27 murmi
   lnDataRec = RECNO()
ELSE
   * noch weitere Durchgänge
   * wieder beim ersten Datenreocrd aufsetzen
   * lnDataRec ist immer noch OK
ENDIF
* allfälligen Escape hier weiterreichen
llGoOn = llPrintOn
IF !llISDJOB
   * wenn's kein Druckjob ist, dann ist nach
   * dem ersten Durchgang Feierabend
   llGoOn = .F.
   * der Abbrech muss zusätzlich gesetzt werden
   * damit der Ausdruck für "Alle" klappt
```

```
         ENDIF
         SELECT (tcRepoDef)
      ENDDO
      * nun sind alle Berichte eines Durchlaufes erfolgt
      * hat es noch Daten im Resultat-Cursor?
      SELECT (lnCurrSele)
   ENDDO WHILE EOF() AND llGoOn
ENDIF llGoOn

IF this.nOutPutTo = 3 AND TYPE("loSession") = "O"
   loSession.finalize()
ENDIF
* präventiv releasen
RELEASE loSession

* wieder auf aktuellen Datensatz in Reports
GO lnCurrRec IN Reports
SELECT (lnCurrSele)

RETURN lcRetVal
```

2.5. onPrepareReport

Erstellen einer Kopie des Originalreports und patchen der Kopi emit sämtliche gewünschte Einstellungen (Drucker, Ränder) . In onPrint wird dann diese Kopie verwendet.

```
LPARAMETERS tcRepoOrig, tcRepoNeu, tcRepoDef, tcMsg
* Para's Passed: <ExpC1>  Name des Original-Reports
*                         (über Search-Path erreichbar)
*                <ExpC2>  Pfad/Name für temporäre Berichtsdatei
*                <ExpC3>  Alias der Berichtsdefinition (aktueller
*                         Record ist zu verw.)
*                <ExpC4>  Var (by Reference) für allfällig Fehlermeldung
LOCAL lnCurrSele, lcExp, llRetVal, lnLOffset, lnTOffset, lnBOffset,;
      nDefTray, lnMagicNumber, ;
      lnFRULeftOffset, lnFRUTopOffset, lnFRUBotOffset, loErr, lnSeleNeu

lnCurrSele = SELECT()
lcExp = ""
llRetVal = .T.
* Faktor für die Umrechnung des Offsets
lnMagicNumber = 3937
* Select-Bereich für den neuen Report
lnSeleNeu = 0

TRY
   * Einstellungen gem. Definition
   SELECT (tcRepoDef)
   lnLOffset   = nLOffset
   lnTOffset   = nTOffset
   lnBOffset   = nBOffset
   lcOptions   = mOptions
   IF EMPTY(nDefTray)
      * "vernünftigen" Default
      lnDefTray = 15
   ELSE
      lnDefTray = nDefTray
   ENDIF
   * Expr bereitstellen
   lcExp = "ORIENTATION="   + ALLTRIM(STR(nPOrient))    + ID_CRLF +;
           "PAPERSIZE="     + ALLTRIM(STR(nPaperSize)) + ID_CRLF +;
           "DEFAULTSOURCE=" + ALLTRIM(STR(lnDefTray))  + ID_CRLF +;
           "COLOR="         + ALLTRIM(STR(nColor))     + ID_CRLF +;
           "COPIES="        + ALLTRIM(STR(nCopies))
   IF !EMPTY(mOptions)
      lcExp = lcExp + ID_CRLF + mOptions
   ENDIF
   * temporären Bericht bereitstellen
   * wegen allfälligen Leerzeichen in Pfad alles mit ;
   * Makro-Subst!
   SELECT 0
   USE (tcRepoOrig)
   COPY all to &tcRepoNeu
   USE &tcRepoNeu EXCLUSIVE
   lnSeleNeu = SELECT()
   LOCATE
   * Definitions-Felder
   REPLACE tag WITH "", ;
           tag2 WITH "", ;
```

```
            expr WITH lcExp

   * Linker Rand
   IF !EMPTY(lnLOffset)
      lnFRULeftOffset = ROUND((lnLOffset * lnMagicNumber),0)
      REPLACE hPos WITH hPos + lnFRULeftOffset
   ENDIF
   * oberer Rand
   IF !EMPTY(lnTOffset)
      lnFRUTopOffset  = ROUND((lnTOffset * lnMagicNumber),0)
      * Pager Header
      LOCATE FOR objtype = 9 AND objcode = 1
      IF FOUND()
         REPLACE height WITH height + lnFRUTopOffset
      ENDIF
      * restliche Report-Positionen gem. oberem Rand anpassen
      SCAN FOR INLIST(objtype, 5, 6, 7, 8, 17)
           REPLACE vpos WITH vpos + lnFRUTopOffset
      ENDSCAN
   ENDIF
   * unterer Rand
   IF !EMPTY(lnBOffset)
      lnFRUBotOffset  = ROUND((lnBOffset * lnMagicNumber),0)
      * Pagefooter anpassen
      LOCATE FOR objtype = 9 AND objcode = 7
      IF FOUND()
         REPLACE height WITH height + lnFRUBotOffset
      ENDIF
   ENDIF
   USE
   FLUSH

CATCH TO loErr
      llRetVal = .F.
      tcMsg = "Fehler beim Bereitstellen des Berichtes: " +;
              ALLTRIM(reports.cDosName) + ID_CRLF +;
              ALLTRIM(STR(loErr.ErrorNo)) + "/" +;
              loErr.Message

FINALLY

   * allenfalls offene Berichtsdatei schliessen
   IF lnSeleNeu > 0
      SELECT (lnSeleNeu)
      USE
   ENDIF

ENDTRY

SELECT (lnCurrSele)

RETURN llRetVal
```

3. Integration von XFRX

XFRX ist ein Report Konverter der Firma Eqeus,, mit dem FoxPro Reports in verschiedene Formate (u.a. PDF, Excel, Word, Html, TIFF) exportiert und dies in erstaunlicher Qualität und Geschwindigkeit. Auch ist die Kompatibilität erstaunlich. Bisher habe ich keinen Bericht in meiner Sammlung gefunden, der sich nicht korrekt exportieren liesse.

In Commando wird die Version für VFP 8 Reports verwendet:

- Der Konverter ist in die XFRX.app kompiliert
- Die Supportdateien werden mit dem app ins Applikationsverzeichnis kopiert.

Dies hat den Vorteil, dass eine neue Version von XFRX problemlos verteilt werden kann, ohne dass die Hauptapplikation verändert werden muss. Natürlich sind dann nicht automatisch neue Features drin, aber mindestens werden so die gefixten Bugs berücksichtigt.

Die Verwendung von XFRX ist denkbar einfach und es braucht nur einige wenige Zeilen Code. Hier ein Minimalbeispiel für PDF Output

```
* initialisieren
loSession = EVALUATE("xfrx('xfrx#init')")
lnResult = loSession.setparams("Test.pdf",,,,,,"PDF")
* Report verarbeiten
loSession.ProcessReport(EVALUATE("MeinReport")
* Dokument schliessen
loSession.finalize()
```

Die ausführliche Variante finden Sie im Code der Methode onReport.

XFRX gibt es natürlich auch in der Variante als Report Listener für VFP9. Ausserdem steht ein Viewer zur Verfügung, der alle Features von XFRX auch bei der Anzeige zur Verfügung stellt. Ausführliche Informationen sind auf der Homepage von Eqeus zu finden:

www.eqeus.com

4. Anhang

Details der für die Reportsteuerung verwendeten Tabellen:

Reports.
Repoform
PrintInf

Commando
Auftragsbearbeitung und Warenwirtschaft
Attributsverzeichnis

Entität: reports
Tabelle: REPORTS (1252)
Definition: Verzeichnis der Berichte
Beschreibung: alle im System verfügbaren Berichte (dazu gehören auch Etiketten und Export) sind hier verzeichnet

Attribut	Definition	Feldname	Typ	Länge	Dec.	PK	Form./Mask	Anschrift (Caption)
RepoID	automatisches Nummernfeld - Reports	nRepoID	I	4	0	*		ID
CDOSNAME	physischer Name des Berichtes oder Dateierweiterung bei Export	CDOSNAME	C	20	0			DOS-Name
CBEZEICH	Name des Eintrages	CBEZEICH	C	30	0			Bezeichnung
CTYP	REPO Report LABE Label EXPO Datenexport CUST Spezialanwendung (gem. Form.OnCust)	CTYP	C	4	0			TYP
CWINDSHOW	Filter für Anzeige gemäss Form	CWINDSHOW	C	4	0			Kontext
LEDITABLE	.T. = Bericht darf (Berechtigung vorausgesetzt) editiert werden	LEDITABLE	L	1	0			Editierbar
MFRX	Export = Exporttyp (leer = VFP-Tabelle) Report = ggf. FRX des Originalreports	MFRX	M	4	0			FRX
MFRT	Export = Feldliste Report = ggf. FRT des Originalberichtes	MFRT	M	4	0			FRT
CSELECT	Stammlisten = Name der Entität (Alias)	CSELECT	C	30	0			Kontext 2
NMODUS	frei verwendbarer Modus (gem. Form.onQuery)	NMODUS	N	2	0			Modus
NSORT	frei verwendbarer Sortm. Form.onQuery)	NSORT	N	2	0			Sort
SBText	Dokument, welches den Word Serienbrief Text enthält	mSBText	M	4	0			Seriebrief Dok.
SBDokWahl	.T. = das Word Serienbrief Dokument kann gewählt oder geändert werden .F. = keine Änderung möglich	lSBDokWahl	L	1	0			SB Dok. änderbar
ATVorl	Name der Vorlage, die den Autotext für Fusszeilen etc. enthält	mATVorl	M	4	0			Autotextvorlage
RepDesc	Beschreibung des Berichtes (User-Info, ev. als Tool-Tip etc.)	mRepDesc	M	4	0			Beschreibung

Legende Datentyp: C=Zeichen, D=Datum, T=Datum/Zeit, L=Logisch, N=Numerisch, I=Ganzzahl, B=Double Precision, Y=Währung, M=Memo, G=Objekt

Attributsverzeichnis

Attribut	Definition	Feldname	Typ	Länge	Dec.	PK	Form./Mask	Anschrift (Caption)
DefPrinter	Default Drucker, auf dem der Report auszugeben ist (leer = es wird der aktuelle Standarddrucker verwendet). Diese Einstellung kann zur Laufzeit übersteuert werden. Name gemäss SET("PRINTER",3), verwendbar für SET PRINTER TO NAME	cDefPrinter	C	50	0			Drucker
DefTray	Default Drucker Schacht (gemäss PRTINFO(7))	nDefTray	I	4	0			Schacht
Color	Modus bei Farbdruckern 1 = Monochrome, 2 = Farbdruck, 3 = n/a (wird nicht bei allen Druckern unterstützt)	nColor	I	4	0			Farbmodus
Copies	Anzahl Kopien	nCopies	I	4	0			Kopien

Entität: RepoForm

Tabelle: RepoForm (1252)

Definition: Ausgabeparameter für Reports

Beschreibung: Jedes FRX/LBX hat hier einen Eintrag in dem die Parameter für den Druck festgelegt werden können. Anmerkungen zu den Rändern (Links, Oben, Unten): die hier eingetragenen Werte werden den im Bericht definierten Randwerte hinzugezählt. Ein positiver Wert erhöht den Rand, ein negativer verringert ihn (Achtung: Man kann damit den Druckbeginn in den unsichtbaren Bereich verschieben, resp. bei einem negativen Wert kann die Fusszeile plötzlich im Seitenkopf gedruckt werden). Diese Werte werden beim Druck als Default vorgeschlagen.
Hinweis zum Berichtsdesign: Berichte mit Rand 0 definieren und hier die notwendigen Ränder einstellen.

Attribut	Definition	Feldname	Typ	Länge	Dec.	PK	Form./Mask	Anschrift (Caption)
CDOSNAME	physischer Name des Berichtes oder Dateierweiterung bei Export	CDOSNAME	C	20	0	*		DOS-Name
CTYP	REPO Report LABE Label EXPO Datenexport CUST Spezialanwendung (gem. Form.OnCust)	CTYP	C	4	0	*		TYP
FormName	Name, Bezeichnung des Formulares	cFormName	C	30	0			Name
LOffset	Linker Rand (in cm, abhängig vom Drucker)	nLOffset	N	5	2		99.99	Linker Rand
TOffset	Oberer Rand (in cm, abhängig vom Drucker)	nTOffset	N	5	2		99.99	Oberer Rand
BOffset	unterer Rand (in cm, abhängig vom Drucker) Anmerkung: der untere Rand wird für einen Report nur korrekt berechnet, wenn im Seitenfuss nur Objekte sind, die sich nach oben ausrichten (technisch: der Seitenfuss wird um diesen Offset verändert)	nBOffset	N	5	2		99.99	Unterer Rand
POrient	Ausruchtung des Papiers: 0 = Portrait, 1 = Landscape	nPOrient	I	4	0			Ausrichtung
PaperSize	Papierformat gemäss PRTINFO(2), Default = 9 (A4)	nPaperSize	I	4	0			Format
Options	Der Inhalt dieses Feldes wird beim Report dem Feld EXP angehängt, die hier definierten Werte überschreiben also alle vorgängig ermittelten. Sämtliche im Feld EXP gültigen Werte sind hier erlaubt (1 Eintrag pro Zeile).	mOptions	M	4	0			Druckoptionen

Entität: PrintInf

Tabelle: PrintInf (1252)

Definition: Erklärungen zu Drucker Konstanten

Beschreibung: Druckerinformationen zu Schacht, Papiergrösse etc. liegen nur in numerischer Form vor. Diese Tabelle enthält den dazugehörenden Klartext.

Attribut	Definition	Feldname	Typ	Länge	Dec.	PK	Form./Mask	Anschrift (Caption)
Modus	Modus gemäss PrtInfo() 1 = Ausrichtung 2 = Papierformat 7 = Druckerschacht 8 = Druckqualität 9 = Farbe 10 = Duplexmodus 12 = TrueType Font handling 13 = Collate gemäss PrtInfo()	nModus	I	4	0			Modus
Setting	Wert gemäss PRTINFO(nModus)	nSetting	I	4	0			Einstellung
PrtSet	Name der Einstellung gemäss nModus und nSetting	cPrtSet	C	40	0			Bezeichnung