



HORIZONT

Smart/JCL - REXX-Interface Erfahrungsbericht

Der einfache Weg zur fehlerfreien JCL

AGENDA

Smart/JCL - REXX-Interface Erfahrungsbericht

Allgemeines

Argumente für den Einsatz von SRI

JCL Änderung mit/ohne SRI-Funktionen

Prüfen von Standards in Batchjobs

Verändern von JCL für einen Integrationstest

Was ist SRI?

SRI steht für **S**martJCL **R**EXX **I**nterface

- SRI ist ein REXX „Function Package“
- SRI ist ein Bestandteil von SmartJCL 2.0, dem JCL-Checker von HORIZONT.
- Spezielle REXX Funktionen ermöglichen lesenden und schreibenden Zugriff auf „tabellarisierte“ JCL
- Ermöglicht die flexible Erweiterungen von SmartJCL, von einfachsten Überprüfungen von Unternehmensstandards bis hin zu komplexen JCL-Generatoren

Argumente für den Einsatz von SRI

SRI bietet dem Anwender:

- hohe Flexibilität
- erheblich geringeren Codieraufwand
- standardisierte und fehlerfreiere JCL
- Zeit- und Kostenersparnis durch Wegfall manueller Tätigkeiten

Argumente für den Einsatz von SRI

- Einhaltung von Unternehmensstandards.
- Massenupdates mit logischen Bedingungen.
- Schneller Aufbau von Testumgebungen.
- Aufbau und Integration von individuellen Fehlermeldungen.

Job-Control Änderung mit REXX mit und ohne SRI-Funktionen

Aufgabenstellung

- Bestimmte Jobs sollen geändert werden.
- Die Jobs sollen Jobclass „Y“ erhalten.
- Falls UNIT-Parameter WORK vorhanden ist, soll dieser entfernt werden.
- Wenn DD-Statement kein UNIT=TAPE hat, soll BLKSIZE Parameter entfernt werden.

Job Control vor der Änderung

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT P391B.CNTL(TEST001) - 01.03 Columns 00001 00070
Command ==>            Scroll ==> CSR
***** Top of Data *****
000001 //TEST001 JOB 'ACCOUNT',
000002 //          CLASS=E,MSGCLASS=T,NOTIFY=8SYSUID
000003 //DEL      EXEC PGM=IDCAMS
000004 //SYSPRINT DD  SYSOUT=*
000005 //SYSIN    DD  *
000006 DEL P391B.XMIT
000007 DEL P391B.XMIT.TAPE
000008 SET MAXCC = 0
000009 //ALLOCC1  EXEC PGM=IEFBR14
000010 //DATE00   DD  UNIT=WORK,
000011 //          DISP=(,CATLG,DELETE),
000012 //          SPACE=(TRK,(100,50),RLSE),
000013 //          RECFM=FB,LRECL=80,BLKSIZE=3120,
000014 //          DSN=P391B.XMIT.WORK
000015 //ALLOCC2  EXEC PGM=IEFBR14
000016 //DATEI1   DD  UNIT=TAPE,DISP=(,CATLG,DELETE),
000017 //          SPACE=(TRK,(100,50),RLSE),
000020 //          RECFM=FB,LRECL=80,BLKSIZE=3120,
MA a
```

Ändern CLASS=Y

UNIT=WORK entfernen.

BLKSIZE entfernen weil
UNIT<>TAPE.

REXX ohne/mit SRI-Funktion

- ohne SRI: ca. 280 REXX-Zeilen
- mit SRI: ca. 25 REXX-Zeilen

Beispiele bzw. REXX-Auszüge siehe nächste Folien

REXX ohne SRI-Funktion

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT          P391B.CLIST(HOTXREXO) - 01.14          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
000119 U_JOB_END:/*----- JOBCARD END ----- */
000120 CALL U_STORE_JOB
000121 DO JOX = 1 TO JO.0                               /* CLASS PARM WAS FOUND */
000122     JO.CLAS = POS(",CLASS",JO.JOX)
000123     IF JO.CLAS = 0 THEN POS(" CLASS",JO.JOX)
000124     IF JO.CLAS /= 0 THEN DO
000125         JO.JOX = OVERLAY("Y",JO.JOX,JO.CLAS + 6,1)
000126         LEAVE
000127     END
000128 END
000129 IF JO.CLAS = 0 THEN DO                               /* CLASS PARM WASN'T FOUND */
000130     IF LENGTH(JO.JOX) < 63 THEN,
000131         JO.JOX = JO.JOX",CLASS=Y"
000132     ELSE DO
000133         JO.JOX = JO.JOX", "
000134         JOX = JOX + 1
000135         JO.JOX = "//          CLASS=Y"
000136     END
000137 JO_KZ = 0
000138 RETURN
```

280 Zeilen.

CLASS vorhanden?
Wo? Achtung wegen
MSGCLASS!!

Einfügen! Wo?
Länge der Zeile!
Formatierung!

REXX ohne SRI-Funktion

- Einlesen div. Programmparameter
- Einlesen der Memberliste mit LMMLIST
- Entscheiden ob Member bearbeitet wird
- Zerlegen der JCL-Zeilen
- Ermitteln der Folgezeilen
- Ersetzen/löschen der Parameters
- ...

REXX mit SRI-Funktion

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT      P391B.CLIST(HOTXREXI) - 01.05          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
000002 stmtptr      = SriGetStmt(1)/* GET THE POINTER TO THE FIRST STATEMENT */
000003 DO WHILE stmtptr /= 0                      /* PROCESS EVERY STATEMENT */
000004   stmttype = SRISayStmtType(stmtptr)
000005   IF stmttype = "JOB" THEN DO
000006     parmptr = SRIFindParm(stmtptr, "CLASS")
000007     IF parmptr /= 0 THEN ,
000008       parmptr = SRISetParmValue(stmtptr, parmptr, "Y")
000009     ELSE parmptr = SRIInsParm(stmtptr, "CLASS", "Y")
000010   END
000011   IF stmttype = "DD" THEN DO
000012     parmptr = SRIFindParm(stmtptr, "UNIT")
000013     IF parmptr /= 0 THEN DO
000014       WRK_unit = SRISayParmValue(stmtptr, parmptr)
000015       /* UNIT=TAPE & BLKSIZE */
000016       parmptr1 = SRIFindParm(stmtptr, "BLKSIZE")
000017       IF parmptr1 /= 0 & WRK_unit /= "TAPE" THEN,
000018         parmptr1 = SRIDelParm(stmtptr, parmptr1)
000019       /* UNIT= WORK */
000020       IF WRK_unit = "WORK" THEN parmptr = SRIDelParm(stmtptr, parmptr)
000021     END
END
```

Nur 25 Zeilen.

Bearbeiten CLASS in der Jobkarte.

UNIT

BLKSIZE

REXX mit SRI-Funktion

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT          P391B.CLIST(HOTXREXI) - 01.05          Columns 00001 00072
Command ==>                                     Scroll ==> CSR
000002 stmtptr      = SriGetStmt(1) /* GET THE POINTER TO THE FIRST STATEMENT */
000003 DO WHILE stmtptr /= 0                /* PROCESS EVERY STATEMENT */
000004     stmttype = SRISayStmtType(stmtptr)
000005     IF stmttype = "JOB" THEN DO
000006         parmtr = SRIFindParm(stmtptr, "CLASS")
000007         IF parmtr /= 0 THEN ,
000008             parmtr = SRISetParmValue(stmtptr, parmtr, "Y")
000009         ELSE parmtr = SRIInsParm(stmtptr, "CLASS", "Y")
000010     END
000011     IF stmttype = "DD" THEN DO
000012         parmtr = SRIFindParm(stmtptr, "UNIT")
000013         IF parmtr /= 0 THEN DO
000014             WRK_unit = SRISayParmValue(stmtptr, parmtr)
000015             /* UNIT=TAPE & BLKSIZE */
000016             parmtr1 = SRIFindParm(stmtptr, "BLKSIZE")
000017             IF parmtr1 /= 0 & WRK_unit /= "TAPE" THEN,
000018                 parmtr1 = SRIDelParm(stmtptr, parmtr1)
000019             /* UNIT= WORK */
000020             IF WRK_unit = "WORK" THEN parmtr = SRIDelParm(stmtptr, parmtr)
000021         END
END
```

Alle Statements lesen.
Start bei 1

REXX mit SRI-Funktion

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT          P391B.CLIST(HOTXREXI) - 01.05          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
000002 stmtptr      = SriGetStmt(1)/* GET THE POINTER TO THE FIRST STATEMENT */
000003 DO WHILE stmtptr /= 0                          /* PROCESS EVERY STATEMENT */
000004  stmttype = SRISayStmtType(stmtptr)
000005  IF stmttype = "JOB" THEN DO
000006    parmptr = SRIFindParm(stmtptr, "CLASS")
000007    IF parmptr /= 0 THEN ,
000008      parmptr = SRISetParmValue(stmtptr, parmptr, "Y")
000009  ELSE parmptr = SRIInsParm(stmtptr, "CLASS", "Y")
000010  END
000011  IF stmttype = "DD" THEN DO
000012    parmptr = SRIFindParm(stmtptr, "UNIT")
000013    IF parmptr /= 0 THEN DO
000014      WRK_unit = SRISayParmValue(stmtptr, parmptr)
000015      /* UNIT=TAPE & BLKSIZE */
000016      parmptr1 = SRIFindParm(stmtptr, "BLKSIZE")
000017      IF parmptr1 /= 0 & WRK_unit /= "TAPE" THEN,
000018        parmptr1 = SRIDelParm(stmtptr,parmptr1)
000019      /* UNIT= WORK */
000020      IF WRK_unit = "WORK" THEN parmptr = SRIDelParm(stmtptr, parmptr)
000021  END
```

Ermitteln des Statement-Typ
JOB, DD,CMT, EXEC ...

REXX mit SRI-Funktion

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT          P391B.CLIST(HOTXREXI) - 01.05          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
000002 stmtptr      = SriGetStmt(1)/* GET THE POINTER TO THE FIRST STATEMENT */
000003 DO WHILE stmtptr /= 0                          /* PROCESS EVERY STATEMENT */
000004   stmntype = SRISauStmtType(stmtptr)
000005   IF stmntype = "JOB" THEN DO
000006     parmptr = SRIFindParm(stmtptr, "CLASS")
000007     IF parmptr /= 0 THEN ,
000008       parmptr = SRISetParmValue(stmtptr, parmptr, "Y")
000009     ELSE parmptr = SRIInsParm(stmtptr, "CLASS", "Y")
000010   END
000011   IF stmntype = "DD" THEN DO
000012     parmptr = SRIFindParm(stmtptr, "UNIT")
000013     IF parmptr /= 0 THEN DO
000014       WRK_unit = SRISayParmValue(stmtptr, parmptr)
000015       /* UNIT=TAPE & BLKSIZE */
000016       parmptr1 = SRIFindParm(stmtptr, "BLKSIZE")
000017       IF parmptr1 /= 0 & WRK_unit /= "TAPE" THEN,
000018         parmptr1 = SRIDelParm(stmtptr,parmptr1)
000019       /* UNIT= WORK */
000020       IF WRK_unit = "WORK" THEN parmptr = SRIDelParm(stmtptr, parmptr)
000021     END
END
```

Bei Statement-Typ JOB wird nach CLASS gesucht.

REXX mit SRI-Funktion

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT      P391B.CLIST(HOTXREXI) - 01.05          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
000002 stmtptr      = SriGetStmt(1)/* GET THE POINTER TO THE FIRST STATEMENT */
000003 DO WHILE stmtptr /= 0                      /* PROCESS EVERY STATEMENT */
000004   stmttype = SRISayStmtType(stmtptr)
000005   IF stmttype = "JOB" THEN DO
000006     parmptr = SRIFindParm(stmtptr, "CLASS")
000007     IF parmptr /= 0 THEN ,
000008       parmptr = SRISetParmValue(stmtptr, parmptr, "Y")
000009     ELSE parmptr = SRIInsParm(stmtptr, "CLASS", "Y")
000010   END
000011   IF stmttype = "DD" THEN DO
000012     parmptr = SRIFindParm(stmtptr, "UNIT")
000013     IF parmptr /= 0 THEN DO
000014       WRK_unit = SRISayParmValue(stmtptr, parmptr)
000015       /* UNIT=TAPE & BLKSIZE */
000016       parmptr1 = SRIFindParm(stmtptr, "BLKSIZE")
000017       IF parmptr1 /= 0 & WRK_unit /= "TAPE" THEN,
000018         parmptr1 = SRIDelParm(stmtptr,parmptr1)
000019       /* UNIT= WORK */
000020       IF WRK_unit = "WORK" THEN parmptr = SRIDelParm(stmtptr, parmptr)
000021     END
END
```

Wenn CLASS vorhanden,
dann neuen Wert einfügen

Wenn CLASS fehlt, dann
ganzen Parameter neu
einfügen

REXX mit SRI-Funktion

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT      P391B.CLIST(HOTXREXI) - 01.05          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
000002 stmtptr      = SriGetStmt(1)/* GET THE POINTER TO THE FIRST STATEMENT */
000003 DO WHILE stmtptr /= 0                      /* PROCESS EVERY STATEMENT */
000004     stmttype = SRISayStmtType(stmtptr)
000005     IF stmttype = "JOB" THEN DO
000006         parmptr = SRIFindParm(stmtptr, "CLASS")
000007         IF parmptr /= 0 THEN ,
000008             parmptr = SRISetParmValue(stmtptr, parmptr, "Y")
000009         ELSE parmptr = SRIInsParm(stmtptr, "CLASS", "Y")
000010     END
000011     IF stmttype = "DD" THEN DO
000012         parmptr = SRIFindParm(stmtptr, "UNIT")
000013         IF parmptr /= 0 THEN DO
000014             WRK_unit = SRISayParmValue(stmtptr, parmptr)
000015             /* UNIT=TAPE & BLKSIZE */
000016             parmptr1 = SRIFindParm(stmtptr, "BLKSIZE")
000017             IF parmptr1 /= 0 & WRK_unit /= "TAPE" THEN,
000018                 parmptr1 = SRIDelParm(stmtptr,parmptr1)
000019             /* UNIT= WORK */
000020             IF WRK_unit = "WORK" THEN parmptr = SRIDelParm(stmtptr, parmptr)
000021         END
END
```

BLKSIZE entfernen

UNIT=TAPE & BLKSIZE

UNIT=WORK entfernen

UNIT= WORK

REXX mit SRI-Funktion

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT      P391B.CLIST(HOTXREXI) - 01.05          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
000002 stmtptr      = SriGetStmt(1)/* GET THE POINTER TO THE FIRST STATEMENT */
000003 DO WHILE stmtptr /= 0                      /* PROCESS EVERY STATEMENT */
000004   stmttype = SRISayStmtType(stmtptr)
000005   IF stmttype = "JOB" THEN DO
000006     parmtr = SRIFindParm(stmtptr, "CLASS")
000007     IF parmtr /= 0 THEN ,
000008       parmtr = SRISetParmValue(stmtptr, parmtr, "Y")
000009   ELSE parmtr = SRIInsParm(stmtptr, "CLASS", "Y")
000010 END
000011 IF stmttype = "DD" THEN DO
000012   parmtr = SRIFindParm(stmtptr, "UNIT")
000013   IF parmtr /= 0 THEN DO
000014     WRK_unit = SRISayParmValue(stmtptr, parmtr)
000015     /* UNIT=TAPE & BLKSIZE */
000016     parmtr1 = SRIFindParm(stmtptr, "BLKSIZE")
000017     IF parmtr1 /= 0 & WRK_unit /= "TAPE" THEN,
000018       parmtr1 = SRIDelParm(stmtptr, parmtr1)
000019     /* UNIT= WORK */
000020     IF WRK_unit = "WORK" THEN parmtr = SRIDelParm(stmtptr, parmtr)
000021 END
```

wenn DD-Statement, suche nach UNIT-Parameter

REXX mit SRI-Funktion

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT      P391B.CLIST(HOTXREXI) - 01.05          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
000002 stmtptr      = SriGetStmt(1)/* GET THE POINTER TO THE FIRST STATEMENT */
000003 DO WHILE stmtptr /= 0                      /* PROCESS EVERY STATEMENT */
000004     stmttype = SRISayStmtType(stmtptr)
000005     IF stmttype = "JOB" THEN DO
000006         parmptr = SRIFindParm(stmtptr, "CLASS")
000007         IF parmptr /= 0 THEN ,
000008             parmptr = SRISetParmValue(stmtptr, parmptr, "Y")
000009         ELSE parmptr = SRIInsParm(stmtptr, "CLASS", "Y")
000010     END
000011     IF stmttype = "DD" THEN DO
000012         parmptr = SRIFindParm(stmtptr, "UNIT")
000013         IF parmptr /= 0 THEN DO
000014             WRK_unit = SRISayParmValue(stmtptr, parmptr)
000015             /* UNIT=TAPE & BLKSIZE */
000016             parmptr1 = SRIFindParm(stmtptr, "BLKSIZE")
000017             IF parmptr1 /= 0 & WRK_unit /= "TAPE" THEN,
000018                 parmptr1 = SRIDelParm(stmtptr, parmptr1)
000019             /* UNIT= WORK */
000020             IF WRK_unit = "WORK" THEN parmptr = SRIDelParm(stmtptr, parmptr)
000021         END
000021     END
```

Wenn UNIT /= Tape & BLKSIZE
vorhanden, BLKSIZE entfernen.

REXX mit SRI-Funktion

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT      P391B.CLIST(HOTXREXI) - 01.05          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
000002 stmtptr      = SriGetStmt(1)/* GET THE POINTER TO THE FIRST STATEMENT */
000003 DO WHILE stmtptr /= 0                      /* PROCESS EVERY STATEMENT */
000004   stmttype = SRISayStmtType(stmtptr)
000005   IF stmttype = "JOB" THEN DO
000006     parmtr = SRIFindParm(stmtptr, "CLASS")
000007     IF parmtr /= 0 THEN ,
000008       parmtr = SRISetParmValue(stmtptr, parmtr, "Y")
000009   ELSE parmtr = SRIInsParm(stmtptr, "CLASS", "Y")
000010 END
000011 IF stmttype = "DD" THEN DO
000012   parmtr = SRIFindParm(stmtptr, "UNIT")
000013   IF parmtr /= 0 THEN DO
000014     WRK_unit = SRISayParmValue(stmtptr, parmtr)
000015     /* UNIT=TAPE & BLKSIZE */
000016     parmtr1 = SRIFindParm(stmtptr, "BLKSIZE")
000017     IF parmtr1 /= 0 & WRK_unit /= "TAPE" THEN,
000018       parmtr1 = SRIDelParm(stmtptr,parmtr1)
000019     /* UNIT= WORK */
000020     IF WRK_unit = "WORK" THEN parmtr = SRIDelParm(stmtptr, parmtr)
000021 END
```

Wenn UNIT=WORK,
Parameter entfernen.

Job Control nach der Änderung

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT      P391B.RESULT(TEST001) - 01.00      Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000001 //TEST001  JOB 'ACCOUNT',
000002 //          CLASS=Y,MSGCLASS=T,NOTIFY=&SYSUID
000003 //DEL      EXEC PGM=IDCAMS
000004 //SYSPRINT DD  SYSOUT=*
000005 //SYSIN     DD  *
000006  DEL P391B.XMIT
000007  DEL P391B.XMIT.TAPE
000008  SET MAXCC = 0
000009 //ALLOC1    EXEC PGM=IEFBR14
000010 //DATE00    DD  DISP=(,CATLG,DELETE),
000011 //          SPACE=(TRK,(100,50),RLSE),
000012 //          RECFM=FB,LRECL=80,
000013 //          DSN=P391B.XMIT.WORK
000014 //ALLOC2    EXEC PGM=IEFBR14
000015 //DATEI1    DD  UNIT=TAPE,DISP=(,CATLG,DELETE),
000016 //          SPACE=(TRK,(100,50),RLSE),
000017 //          RECFM=FB,LRECL=80,BLKSIZE=3120,
000018 //          DSN=P391B.XMIT.TAPE
MA a
```

SmartJCL formatiert automatisch

- Erheblich geringerer Codieraufwand.
Bis zu 90% weniger REXX-Code!
- Dadurch erhebliche Zeitersparnis.
- Leichtere und eindeutigere Identifizierung von Statements und Parametern.
- Schneller zu guten und richtigen Ergebnissen.
- Automatische Formatierung, entspricht deshalb festgelegten Standards.

Haben Sie Fragen ?



info@horizont-it.com

Erfahrungsbericht über Einführung von Smart-JCL REXX-Interface

Das Unternehmen

- Europäische Grossbank
- Ca. 2000 Mitarbeiter
- 25.000 Jobs, davon ca. 15.000 täglich
- Mix von Prozeduren und JCL
- Automatisches Übergabeverfahren:
Entwicklung → Test → Produktion.
- **100 %** standardisierte JCL!

Prüfen von Standards in Batchjobs

Einsatzmöglichkeiten

- Bei dem Transfer eines einzelnen Jobs von der Entwicklungs- zur Integrationstestumgebung.
- Bei einem wöchentlichen Massencheck über alle Job-Control Bibliotheken.

Übersicht der Prüfkriterien

- Syntax-Check der gesamten JCL.
- Die Anzahl der Steps ist beschränkt.
- Existenz einer bestimmten Prozedur als erste EXEC-Angabe.
 - Für die Prozedur muss eine bestimmte Überschreibung vorhanden sein.
- Abhängig vom Sachgebiet findet eine Überprüfung folgender Angaben statt:
 - Haben Job-Account, CLASS und MSGCLASS die richtigen Werte?
 - NOTIFY ist nicht zulässig!

Übersicht der Prüfkriterien

- Hat der Jobname die richtige Länge und Struktur?
- Wie ist die Reihenfolge der JOBLIB-Statements?
(ist je nach Sachgebiet unterschiedlich).
- Sind abweichende Joblib-Dateinamen erlaubt? (Fremdsoftware, Sachgebiets-Bibliotheken)
- Bestimmte SYSOUT-Angaben vorhanden?
z.B. Alle Dumps in eine eigene Klasse.
- Komplette Prüfung der Dateinamen auf Kriterien wie „Sachgebiet im 2. Qualifier“ oder „Teil des Programmnamens im 3. Qualifier“.

Übersicht der Prüfkriterien

- Prüfung sämtlicher Output Angaben.
- Sind die DCB-Angaben komplett vorhanden?
- Ist der RLSE- Parameter vorhanden?
- Richtige UNIT-Angabe bei Bändern ?
- Wird für FTP die Standardprozedur verwendet?
- Sind die FTP-SYSIN vollständig und plausibel?
(Servername, User usw.)
- Entsprechen die Standardprozeduren den erlaubten Prozedurnamen?
(DB2-Zugriffe usw.)
- Sind die Steps aufsteigend nummeriert?
- **Über 650 Prüfungen und Regeln sind möglich**

Code-Beispiel 1

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT      P391B.JCK.SRI.CLIST(SIJLVALR) - 01.58      Columns 00001 00072
Command ==> _____ Scroll ==> CSR_
000282 IF POS("PROC=",DEX_proc_ex) = 1 THEN DO
000283   DEX_txt = "EXEC PROC=<procname> is not allowed on the exec ",
000284           "statement. Use EXEC <procname> instead."
000285   SAY = DEX_txt
000286 END
000287 /* CHECK FOR OVERWRITING STATEMENTS
000288   overstmtptr = SRIGetOverStmt(stmtptr, 0, "FIRST")
000289   DO WHILE overstmtptr /= 0
000290     overstmttype = SRISayOverStmtType(stmtptr, overstmtptr)
000291     overstmtname = SRISayOverStmtName(stmtptr, overstmtptr)
000292
000293     IF overstmttype = "DD" THEN CALL DEX_jcl_dd stmtptr overstmtptr
000294
000295     overstmtptr = SRIGetOverStmt(stmtptr, overstmtptr, "NEXT")
000296   END
000297 RETURN
000298
000299 DEX_JCL_end_prc:/*----- END of Procedur Call -----*/
000300 DEX_proc_lvl = DEX_proc_lvl - 1
000301 IF DEX_proc_lvl = 0 then Dex_prc = "N"

MÄ a 04/015
```

Prüfen ob JCL-Überschreibungen vorhanden sind.

Code-Beispiel 2

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT P391B.JCK.SRI.CLIST(SIJLVALR) - 01.59 Columns
Command ==>
000405 /* SPACE & RETPD RLSE-Parameter available */
000406 IF F_SUB_PARM_EXIST(stmtptr overstmtptr "SPACE") = "OK" &,
000407 F_SUB_PARM_EXIST(stmtptr overstmtptr "RETPD") = "OK" THEN DO
000408 DEX_spc = F_SUB_PARM(stmtptr overstmtptr "SPACE" "0")
000409 IF POS('RLSE',DEX_spc) = 0 THEN DO
000410 IF DEX_mass = "Y" THEN DO
000411 DEX_txt = "The RLSE-subparameter is required with RETPD."
000412 msg = SRImsg("DEX3755W", stmtptr, DEX_txt)
000413 say "3755 "DEX_appl" "DEX_jobname" "DEX_dd_dsn
000414 END
000415 ELSE msg = SRImsg("DEX3755E", stmtptr, DEX_txt)
000416 END
000417 END /* space found */
000418
000419 /* segment */
000420 IF F_SUB_PARM_EXIST(stmtptr overstmtptr "SEGME" = "OK" THEN DO
000421 is not allowed on statement."
000422 stmtptr, DEX
000423 EN
000424
```

Prüfen ob RLSE- Parameter bei Ausgabedateien vorhanden ist.

000406 IF F_SUB_PARM_EXIST(stmtptr overstmtptr "SPACE") = "OK" &,
000407 F_SUB_PARM_EXIST(stmtptr overstmtptr "RETPD") = "OK" THEN DO
000408 DEX_spc = F_SUB_PARM(stmtptr overstmtptr "SPACE" "0")
000409 IF POS('RLSE',DEX_spc) = 0 THEN DO
000410 IF DEX_mass = "Y" THEN DO
000411 DEX_txt = "The RLSE-subparameter is required with RETPD."
000412 msg = SRImsg("DEX3755W", stmtptr, DEX_txt)
000413 say "3755 "DEX_appl" "DEX_jobname" "DEX_dd_dsn
000414 END
000415 ELSE msg = SRImsg("DEX3755E", stmtptr, DEX_txt)
000416 END

Bei Massenprüfung wird eine Warnung ausgegeben.

Bei Übergabe vom Entwickler wird eine Fehlermeldung ausgegeben.

Selektion über DD-JUPJMLI

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT          P391B.JCK.SRI.CLIST(SRIJVALI) - 01.05          Columns 00001 00072
Command ==> _____ Scroll ==> CSR

000011 //          PARM='/' ,REGION=64M
000012 //STEPLIB DD  DSN=P391B.SMARTJCL.LOAD ,
000013 //          DISP=SHR
000014 //          DD  DSN=SYSH.OPC810.SEQOIMPR
000015 //          DISP=SHR
000016 //JUPJMLI DD  *
000017 MEMBER=AC*
000018 EXCLUDE= ACB9*
000019 MEMBER=BB%%9*
000020 EXCLUDE=BC339*
000021 //APPLIC DD  *
000022 ACRE
000023 //JUPJI001 DD  DSN=DATA.JCL.DEXIA.PROJCL.TEST.JCL ,
000024 //          DISP=SHR
000025 //JUPJI002 DD  DSN=DATA.JCL.DEXIA.JOBLIB ,
000026 //          DISP=SHR
000027 //JUPJI003 DD  DSN=P391B.CNTL ,
000028 //          DISP=SHR
000029 //JUPJI004 DD  DSN=P390A.JUS.HEL.SHR.JOBLIB.HEL ,
000030 //          DISP=SHR

MA a 15/002
```

Mit * generisch
Mit % ersetzend

Exclude zum
ausschließen.

Bis zu 999
Bibliotheken möglich.

Selektion über EXEC Parm

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT          P391B.JCK.SRI.CLIST(SRIJVALX) - 01.01          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
000010 //JCLCHECK EXEC PGM=JCKIPS,COND=(04,LT),REGION=64M,
000011 //          PARM=(' /A*,B*')
000012 //STEPLIB DD DISP=SHR,DSN=P391B.SMARTJCL.LOAD
000013 //* JOBLIBS JUPJI001 - JUPJI999
000014 //JUPJI001 DD DISP=SHR,DSN=DATA.JCL.P391B.JCL.TEST.JCL
000015 //* PROCLIBS JUPPI001 - JUPPI999
000016 //JUPPI001 DD DISP=SHR,DSN=DATA.JCL.P391B.PROCLIB.TEST.JCL
000017 //JCKDTLO DD LRECL=80,SYSOUT=*
000018 //JCKSUMO DD LRECL=178,SYSOUT=*
000019 //JCKMIGO DD LRECL=80,SYSOUT=*
000020 //PWDIN DD DISP=SHR,DSN=P391B.SMARTJCL.SKELS(OPZKPWD)
000021 //JCKMSGI DD DISP=SHR,DSN=P391B.SMARTJCL.PANELS(JCKM)
000022 // DD DISP=SHR,DSN=P391B.SMARTJCL.SAMPLIB(SRIM)
000023 //JCKMSXI DD DISP=SHR,DSN=P391B.SMARTJCL.SKELS(JCKKMSXV)
000024 //SYSIN DD DISP=SHR,DSN=P391B.SMARTJCL.SKELS(JCKKPARV)
000025 //SYSEXEC DD DISP=SHR,DSN=P391B.SMARTJCL.SYSEXEC
000026 //REXEXE DD DISP=SHR,DSN=P391B.SMARTJCL.SYSEXEC(REXXEXEC)
000027 //REXHDR DD DISP=SHR,DSN=P391B.SMARTJCL.CLIST(SRIXHEAD)
000028 //REXFNC DD DISP=SHR,DSN=P391B.SMARTJCL.CLIST(SRIXFNCS)
000029 //REXPRG DD DISP=SHR,DSN=DATA.JCL.P391B.CLIST.NEW(SIJLVALR)

MA a 05/002
```

Auswahl einzelner Jobs oder von/bis

REXX-Code des Anwenders.

Ergebnis DD-SYSPRINT - Anfang

```
Sitzung A - [24 x 80]
Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY P391BVAL JOB09922 DSID 106 LINE 0 COLUMNS 02- 81
COMMAND INPUT ==> _ SCROLL ==> CSR
***** TOP OF DATA *****
*****
JCKIPS - CHECK JCL
HORIZONT GMBH, MUNICH, 2004, ALL RIGHTS RESERVED
COMPILED WITH IBM PLI LE V1R1 AT 11 JUN 04 18.01.10
*****
LICENCE KEY CHECK
CPU-ID : 0A0000B11247
LICENCE KEY(S) : BDYDFDYFHK
PRODUCT-ID : JCK
LICENCE KEY CHECK PASSED

JCK0050I START READING MEMBER LIST FROM JUPJMLI
JUP0058I END READING MEMBER LIST FROM JUPJMLI: MEMBERS 1, EXCLUDE 0
JUP0000I 001,AC1ECD2C, START 20040617110531255
JUP0000I 001,AC1ECD2C, STOP 20040617110533558 DURATION: 00:00:02:303
JUP0000I 001,AC1ECE2D, START 20040617110533558
JUP0000I 001,AC1ECE2D, STOP 20040617110539077 DURATION: 00:00:05:519
JUP0000I 001,AC1ECH5B, START 20040617110539077
JUP0000I 001,AC1ECH5B, STOP 20040617110545153 DURATION: 00:00:06:076

MA a 04/021
```

Liste der verarbeiteten Member.

Start- und Endezeit sowie Dauer.

Ergebnis DD-SYSPRINT - Ende

```
Sitzung A - [24 x 80]
Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY P391BVAL JOB09922  DSID   106 LINE 47      COLUMNS 02- 81
COMMAND INPUT ==>
MEMBER INPUT FROM JCKJI001 TO JCKI999:
MEMBERS READ:                               14
MEMBERS WITH INFORMATIONS:                   0
MEMBERS WITH WARNINGS:                       0
MEMBERS WITH ERRORS:                          1
NUMBER OF PROCESSED JCL STATEMENTS:
DD      :           3944
EXEC    :           487
IF      :             0
INCLUDE :             0
JCLLIB  :             0
JES2    :             1
JES3    :             0
JOB     :            14
OUTPUT  :             0
PROCCALL :          470
SET     :             0
TOTAL TIME: 00:03:48:136
*****
JCKIPS ENDED AT 11:09:19:559, RC=0008
MA a                                     04/021
```

1 Member von 14, mit Fehler.

Statistik

RC=8

Ergebnis DD-JCKSUMO - links

```
Sitzung A - [24 x 80]
t Options Help
-----
JOB09922  DSID   104 LINE 0          COLUMN 00  81
SCRN
*****
***** TOP OF DATA *****
MSG.N  MESSAGE TEXT
-----
DEX3255W  745 Dsn EX.USERLIB is not recommended in the joblib concatena
DEX3614W  817 There is a DCB parameter in the overwriting or in the pr
DEX3255W  745 Dsn EX.USERLIB is not recommended in the joblib concatena
DEX3614W  817 There is a DCB parameter in the overwriting or in the pr
DEX3255W  745 Dsn EX.USERLIB is not recommended in the joblib concatena
DEX3614W  817 There is a DCB parameter in the overwriting or in the pr
DEX3255W  745 Dsn EX.USERLIB is not recommended in the joblib concatena
DEX3614W  817 There is a DCB parameter in the overwriting or in the pr
DEX3255W  745 Dsn EX.USERLIB is not recommended in the joblib concatena
DEX3614W  817 There is a DCB parameter in the overwriting or in the pr
DEX3001E 1678 The use of /*XEQ is not allowed.
DEX3255W  745 Dsn EX.USERLIB is not recommended in the joblib concatena
DEX3614W  817 There is a DCB parameter in the overwriting or in the pr
DEX3101E 1213 First character of stepname does not equal the 6th charac
JCK0220E STEP DOES NOT CONTAIN REQUIRED DD NAME SYSUT2
JCK0220E STEP DOES NOT CONTAIN REQUIRED DD NAME SIN
DEX3101E 1213 First character of stepname does not equal the 6th charac
MA a 04/021
```

DEX ... Vom Kunden definierte Meldung

... W = Warning
.... E = Error

Zeilennummer der Meldung im REXX-Code

Kurztext mit Variablen

JCK ... SmartJCL-Meldung

Ergebnis DD-JCKDTLO

```
Sitzung A - [24 x 80]
Display _Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY P391BVAL JOB00644  DSID   103 LINE 16      COLUMNS 02- 81
COMMAND INPUT ==>                                SCROLL ==> CSR
000013 //N000 EXEC SCRDPY,COND=(0,NE)
000014 //S001.DD010 DD DSN=EX.A090030.A00000,
000015 //          DISP=SHR
000016 /**
000017 /**
000018 /**
.HAAB DEX3101E 1213 First character of stepname does not equal the 6th
      character of the jobname. Should be 0.
.HAAB DEX3614W 817 There is a DCB parameter in the overwriting or in the
      procedure. Code the subparms separately.
000019 //D185B EXEC EX999200,COND=(0,EQ,D185A),
000020 //          PARM='/RK1CM08IF2          -YMMDD0500'      (D=(+001,0))
000021 /**
.HAAC JCK0027E PROCEDURE EX999200 NOT FOUND
.HAAC JCK0105E STEP D185A NOT FOUND
.HAAC DEX3614E 1053 Procname EX999200 is not allowed. Use the correct procname
      or contact DHD (tel 27000) to add it to the list.
.HAAC DEX3614W 1074 The procname does not begin with the application code.
.HAAC DEX3101E 1213 First character of stepname does not equal the 6th
      character of the jobname. Should be 0.
```

Kunden-Fehlermeldung

Ausgabe JCL mit Fehlern und Auflösungen.

SmartJCL-Fehlermeldung

04/021

Kunden Message-File

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT          P391B.CLIST(SRIUMSG) - 01.02          Columns 00001 00072
Command ==>  _____ Scroll ==> CSR

***** ***** Top of Data *****
000001 *****
000002 ** CUSTOMER MESSAGES FOR REXX INTERFACE
000003 *****
000004 DEX3001E 08 The use of /*XEQ is not allowed.
000005 DEX3001W 04 The use of /*XEQ is not allowed.
000006 DEX3002E 08 Accounting info is temporarily still required due to a JES2
000007 DEX3002W 04 Accounting info is temporarily still required due to a JES2
000008 DEX3003E 08 Keyword ADDRSPC is not allowed.
000009 DEX3003W 04 Keyword ADDRSPC is not allowed.
000010 DEX3004E 08 Keyword BYTES is not allowed.
000011 DEX3004W 04 Keyword BYTES is not allowed.
000012 DEX3005E 08 Keyword CARDS is not allowed.
000013 DEX3005W 04 Keyword CARDS is not allowed.
000014 DEX3006E 08 Keyword COND is not allowed.
000015 DEX3006W 04 Keyword COND is not allowed.
000016 DEX3007E 08 Keyword GROUP is not allowed.
000017 DEX3007W 04 Keyword GROUP is not allowed.
000018 DEX3008E 08 Keyword LINES is not allowed.
000019 DEX3008W 04 &VAR1 _____

M& a
```

... W = Warning
.... E = Error

Return-Code

Anzeigetext auch
mit Variablen

Haben Sie Fragen ?



info@horizont-it.com

Verändern von Job-Control für einen Integrationstest

Übersicht der Änderungskriterien

- Änderung von folgenden Parametern:
 - Job Name (1 Zeichen an der 3. Stelle)
 - Job-Account
 - CLASS
 - MSGCLASS
 - Alte Joblibs löschen und Neue einfügen
 - Ersetzen bestimmter Pagedef/Formdef sowie bestimmter Verteilerschlüssel

Übersicht der Änderungskriterien

- SYSOUT in BETA93
- Alle Datasets erhalten Test-HLQ
- Bestimmte Prozeduren werden durch Test-prozeduren ersetzt.
- Bestimmte Programme werden mit IEFBR14 ersetzt (z.B. FTP)
- FTP-Steuerkarten werden verändert. (andere Server, User)
- **Über 300 Änderungen sind möglich**

Code-Beispiel 1 (Instream-Data)

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT      DATA.JCL.DEXIA.CLIST.NEW(SIJLCONV) - 01.46      Columns 00001 00072
Command ==> _____ Scroll ==> CSR
000126    dmy = SRIInsOrigLn(stmtptr, 0, New_linetxt, 'FIRST');
000127 END
000128 RETURN
000129
000130 DEX_JCL_SYSIN:/*----- SYSIN Statement ( JCL )-----*/
000131 ARG stmtptr lnptr
000132 DEX_job_sector = SUBSTR(DEX_jobname,1,2)
000133 DEX_sysin_rec  = ""
000134 DEX_sysin_orig = ""
000135 DEX teller     = 1
000136 DEX_sysin_orig = SRISayLn(lnptr)
000137 DEX_sysin_rec  = DEX_sysin_orig
000138 /* ALGEMENE AANPASSING SYSIN: PRD WORDT GTU -----*/
000139 SELECT
000140     WHEN POS('ISPSTART',DEX_sysin_rec) /= 0     THEN NOP
000141     WHEN POS('DEST=ABCPRD',DEX_sysin_rec) /= 0 THEN NOP
000142     WHEN POS('N=VAKPRD2_',DEX_sysin_rec) /= 0 THEN NOP
000143     WHEN POS('REPRDQT',DEX_sysin_rec) /= 0     THEN NOP
000144     OTHERWISE DO
000145         DEX_pos_prd = POS('PRD',DEX_sysin_rec)

```

Bei größeren Aktionen SYSIN Zeile als Variable speichern.

Code-Beispiel 1 -Fortsetzung

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT          P391B.JCK.SRI.CLIST(SIJLCONV) - 01.15          Columns 00001 00072
Command ==>  _                                           Scroll ==> CSP
000625
000626 /* -SERVER = FIN - ----- */
000627 DEX_pos_str = POS('SERVER=FIN -',DEX_sysin_rec)
000628 IF DEX_pos_str /= 0 THEN DO
000629     DEX_sysin_rec = DEX_chg_sysin('SERVER=FIN -','SERVER=D42 -',,
000630     DEX_sysin_rec,stmtptr,DEX_teller)
000631 END
000632
000633 /* AANPASSEN IDMS-DATABASES -----*/
000634 SELECT
000635     WHEN POS(' IDMSD ',DEX_sysin_rec) /= 0 THEN DO
000636         DEX_sysin_rec = DEX_chg_sysin(' IDMSD ',' IDMSU ',,
000637         DEX_sysin_rec,stmtptr,DEX_teller)
000638     END
000639     WHEN POS(' IDMSE ',DEX_sysin_rec) /= 0 THEN DO
000640         DEX_sysin_rec = DEX_chg_sysin(' IDMSE ',' IDMSU ',,
000641         DEX_sysin_rec,stmtptr,DEX_teller)
000642     END
000643     WHEN POS(' IDMSI ',DEX_sysin_rec) /= 0 THEN DO
000644         DEX_sysin_rec = DEX_chg_sysin(' IDMSI ',' IDMSU ',,
MA a                                                                 04/015
```

Verändern von Serverangabe in Steuerkarten.

Nächste Änderung.

Code-Beispiel 2 (Joblib)

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT          DATA.JCL.DEXIA.CLIST.NEW(SIJLCONV) - 01.46      Columns 00001 00007
Command ==>
000742      WHEN DEX_dsn_value = "PRD.CM.LOADLIB"                THEN DO
000743          DEX_nw_dsn = setNode(DEX_dsn_value,"GTU",1)
000744          dmy = F_SUB_parm_Write(DEX_cmt_avl stmtptr "DSN" DEX_nw_dsn)
000745
000746          stmtptr = SRIInsStmt(stmtptr, "", "DD", "AFTER")
000747          parmptr = SRIInsParm(stmtptr, "DSN", "UAT."DEX_ins_sector".LOADLIB")
000748          parmptr = SRIInsParm(stmtptr, "DISP", "SHR")
000749          statement_nr = statement_nr + 1
000750
000751          stmtptr = SRIInsStmt(stmtptr, "", "DD", "AFTER")
000752          parmptr = SRIInsParm(stmtptr, "DSN", "UAT.CM.LOADLIB")
000753          parmptr = SRIInsParm(stmtptr, "DISP", "SHR")
000754          statement_nr = statement_nr + 1
000755
000756          stmtptr = SRIInsStmt(stmtptr, "", "DD", "AFTER")
000757          parmptr = SRIInsParm(stmtptr, "DSN", "PRD."DEX_ins_sector".LOADLIB")
000758          parmptr = SRIInsParm(stmtptr, "DISP", "SHR")
000759          statement_nr = statement_nr + 1
000760
000761          stmtptr = SRIInsStmt(stmtptr, "", "DD", "AFTER")
```

Ändern 1. Qualifier in Joblib-DSN

Danach weitere Joblib-DSN einfügen

Ergebnis Beispiel 2

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT          DATA.JCL.DEXIA.RESULT(AC1ECD2C) - 01.00          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** Top of Data *****
000001 //A21ECD2C JOB ('C0000  ','1  '),
000002 //          GKB-CCB-YYDDD,REGION=0002M,
000003 //          MSGLEVEL=(1,1),MSGCLASS=6,CLASS=A,USER=ACEC
000004 //*
000005 //JOBLIB DD DISP=SHR,
000006 //          DSN=GTU.CM.LOADLIB
000007 //          DD DISP=SHR,
000008 //          DSN=UAT.AC.LOADLIB
000009 //          DD DISP=SHR,
000010 //          DSN=UAT.CM.LOADLIB
000011 //          DD DISP=SHR,
000012 //          DSN=PRD.AC.LOADLIB
000013 //          DD DISP=SHR,
000014 //          DSN=PRD.CM.LOADLIB
000015 //          DD DISP=SHR,
000016 //          DSN=EX.USERLIB
000017 //*****
000018 //* CHAINE D'ARCHIVAGE EN IAFC (DISQUE OPTIQUE) DE LA LISTE *
000019 //*          TRANSFERT DE SOLDE - OVERDRACHT SALDO          *
```

Ändern 1. Qualifier in Joblib-DSN

Joblibs eingefügt.

JCL zum Starten im Batch

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT          P391B.JCK.SRI.CLIST(SRIJUACX) - 01.00          Columns 00001 00072
Command ==>                                         Scroll ==> CSR

000016 //JCLREFER EXEC PGM=JCKIRFMT,
000017 //          PARM='/' ,REGION=64M
000018 //STEPLIB DD DISP=SHR,DSN=P391B.SMARTJCL.LOAD
000019 //JUPJMLI DD *
000020 MEMBER=AC*
000021 //JUPJI001 DD DISP=SHR,DSN=DATA.JCL.DEXIA.PROJCL.TEST.JCL
000022 //PWDIN DD DISP=SHR,DSN=P391B.SMARTJCL.SKELS(OPZKPWD)
000023 //JCKMSGI DD DISP=SHR,DSN=P391B.SMARTJCL.PANELS(JCKM)
000024 // DD DISP=SHR,DSN=DATA.JCL.DEXIA.CLIST.NEW(SRIUSMSG)
000025 // DD DISP=SHR,DSN=P391B.SMARTJCL.SAMPLIB(SRIM)
000026 //SYSIN DD DISP=SHR,DSN=P391B.SMARTJCL.SKELS(JCKKRFMT)
000027 //JUPREDI DD DISP=SHR,DSN=P391B.SMARTJCL.SKELS(JCKKRFMT)
000028 //JUPJCLO DD DISP=(,CATLG,DELETE),DSN=DATA.JCL.DEXIA.WORK1,
000029 //          SPACE=(TRK,(10,10)),LRECL=80,RECFM=FB
000030 //SYSEXEC DD DISP=SHR,DSN=P391B.SMARTJCL.SYSEXEC
000031 //REXPRG DD DISP=SHR,DSN=DATA.JCL.DEXIA.CLIST.NEW(SIJLUACV)
000032 //REXEXE DD DISP=SHR,DSN=P391B.SMARTJCL.SYSEXEC(REXXEXEC)
000033 //REXHDR DD DISP=SHR,DSN=P391B.SMARTJCL.CLIST(SRIXHEAD)
000034 //REXFNC DD DISP=SHR,DSN=P391B.SMARTJCL.CLIST(SRIXFNCS)
000035 //SYSPRINT DD SYSOUT=*

MA a 05/015
```

Selektion über JUPJMLI oder EXEC-Parm

Ausgabe im IEBUPDTE-Format

Kunden REXX-Code

Ergebnis DD-SYSPRINT

```
Sitzung A - [24 x 80]
Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY P391BACV JOB00640  DSID   107 LINE 8          COLUMNS 01- 80
COMMAND INPUT ==>          SCROLL ==> CSR
  LICENCE KEY(S)           : BDHDFDZFWK
  PRODUCT-ID               : JRF
LICENCE KEY CHECK PASSED

JCK0050I START READING MEMBER LIST FROM JUPJMLI
JUP0058I END READING MEMBER LIST FROM JUPJMLI: MEMBERS 1, EXCLUDE 0
MSG.NO.  ERROR MESSAGE                                           JOBNAM
-----
MSG.NO.  ERROR MESSAGE                                           LIB  M
-----

NUMBER OF JOBS READ FROM SEQUENTIAL INPUT JCKJCLI:           0
NUMBER OF MEMBERS READ FROM JUPJI001 TO JUPJI999 :           17

MA a                                                                04/021
```

Ohne Fehler!

17 Member verarbeitet.

Ergebnis DD-JUPJCLO

```
Sitzung A - [24 x 80]
File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT          DATA.JCL.DEXIA.WORK1          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** Top of Data *****
000001 ./ ADD NAME=AC1ECD2C
000002 //A21ECD2C JOB ('C0000  ','1  '),
000003 //          GKB-CCB-YYDDD,REGION=0002M,
000004 //          MSGLEVEL=(1,1),MSGCLASS=6,CLASS=A,USER=ACEC
000005 //*
000006 //JOB LIB DD DISP=SHR,
000007 //          DSN=GTU.CM.LOADLIB
000008 //          DD DISP=SHR,
000009 //          DSN=UAT.AC.LOADLIB
000010 //          DD DISP=SHR,
000011 //          DSN=UAT.CM.LOADLIB
000012 //          DD DISP=SHR,
000013 //          DSN=PRD.AC.LOADLIB
000014 //          DD DISP=SHR,
000015 //          DSN=PRD.CM.LOADLIB
000016 //          DD DISP=SHR,
000017 //          DSN=EX.USERLIB
000018 //*****
000019 //* CHAINE D'ARCHIVAGE EN IAFC (DISQUE OPTIQUE) DE LA LISTE *
```

Ausgabe veränderte JCL. Format für IEBUPDTE

Ausgabe im PO-Format ist geplant.

Vielen Dank für Ihre Aufmerksamkeit! Haben sie Fragen, Anregungen oder Wünsche?



HORIZONT

**Besuchen Sie uns in München oder senden Sie uns bitte eine E-Mail
an info@horizont-it.com**