

InfoDesign GmbH

Customizing DB2 for Customers

Beschreibung zu Build/Transfer xmit-files

Verfahren zum
Übertragen von Dateien mit xmi-Verpackung

|

März 2011

Inhaltsverzeichnis

Inhaltsverzeichnis.....	2
Manuelles Vorgehen	3
Alternativ: Send EINES xmit files	4
Alternativ: Receive EINES xmit files.....	5

Manuelles Vorgehen

Zum Übertragen einer Anzahl von Dateien muß jede einzelne Datei mit xmit verpackt und per FTP oder IND\$FILE übertragen werden. Zum Einen ein mühsames Vorgehen, zum Anderen fehlerbehaftet, da files vergessen oder überschrieben werden können.

Beispiel mit Infomat Bibliotheken einer Version:

9 Bibliotheken zu InfoMAT V9.1 sollen ausgeliefert werden

Für alle aufgeführten Bibliotheken ist RECEIVE notwendig:

```
INFOFLSH.V9R1M0.ROLLOUT.CNTL  
INFOFLSH.V9R1M0.ROLLOUT.DBRM  
INFOFLSH.V9R1M0.ROLLOUT.LOAD  
INFOFLSH.V9R1M0.ROLLOUT.MLIB  
INFOFLSH.V9R1M0.ROLLOUT.PARM  
INFOFLSH.V9R1M0.ROLLOUT.PLIB  
INFOFLSH.V9R1M0.ROLLOUT.REXX  
INFOFLSH.V9R1M0.ROLLOUT.SLIB  
INFOFLSH.V9R1M0.ROLLOUT.SQL
```

Nun gilt es **x-mal** FTP oder IND\$FILE durchzuführen (je nach 3270-Emulation einzeln oder als Stapel).

Diese x PS-Dateien werden verschickt und nun das Ganze umgekehrt: **x-mal** FTP (oder IND\$FILE) und RECEIVE, wofür es wahrscheinlich noch keinen Automatismus gibt, bis jetzt.

Alternativ: Send EINES xmit files

Initial muß manuell ein Job erstellt werden:

```
///. . . JOB ,CLASS=A
/// SET JCLLIB=Bibliothek mit einer funktionierenden IKJ-Prozedur      A
/// SET IDREXX=Bibliothek mit der REXX #IDXMTPO                        B
/// SET IDRXISPF=Bibliothek mit einer REX2ISPF Prozedur                C
///PLIB      JCLLIB ORDER=&JCLLIB
///XMITPO     EXEC IKJPROC2                                           aus A
/// PARM='REX2ISPF #IDXMTPO MLIB REXX SLIB'                          Parameter | *
///SYSEXEC    DD DISP=SHR,DSN=&IDREXX
///           DD DISP=SHR,DSN=&IDRXISPF
///SYSPRINT   DD SYSOUT=*
///REXSYSIN   DD *
DSN2XMIT     = INFOFLSH.V9R1M0.ROLLOUT      -- WHAT TO TRANSMIT
IDCLIST      = SYS1.IDRZ.CLIST              -- REXX(#IDRCVPO) RESIDES IN
```

Man kann jede vorhandene IKJ- und REX2ISPF-Prozedur benutzen. Befindet sich die #IDXMTPO in der gleichen Bibliothek wie die REX2ISPF, muss SYSEXEC auch nicht verkettet werden.

Als Parameter benötigt die Prozedur eine Liste von Low-Level-Qualifier, durch blank getrennt, oder einen `\*` (Stern), wenn alle Dateien von &dsn2xmit.* übertragen werden sollen. Die Liste der Parameter kann beliebig lang sein, auch mehrzeilig. Dann sind Fortsetzungszeichen zu benutzen.

Damit werden alle Dateien mit INFOFLSH.V9R1M0.* (* bedeutet hier nur ein Qualifier) ermittelt. Transmittiert werden aber zunächst nur die Dateien, deren **LLQ** in der PARM-Liste aufgeführt sind (1).

Im letzten Schritt werden die transmittenden Dateien zu einer Datei zusammen gefasst (2). Der Dateiname bildet sich aus der Angabe in DSN2XMIT ergänzt um den Qualifier XMIdmm.

```
CPYRX2IS V9R1M000 24 Mar 2011 10:39:46
Bei Parameter *:
  Argument: * .
  LLQ found: MLIB .
  LLQ found: REXX .
  LLQ found: SLIB .
Bei Parameterliste:
  Argument: MLIB REXX SLIB .
***** transmitting INFOFLSH.V9R1M0.ROLLOUT.MLIB.XMI2403 .
INMX000I 0 message and 18 data records sent as 335 records to U.U
INMX001I Transmission occurred on 03/24/2011 at 10:39:46.
***** transmitting INFOFLSH.V9R1M0.ROLLOUT.REXX.XMI2403 .
INMX000I 0 message and 103 data records sent as 15282 records to U.U
INMX001I Transmission occurred on 03/24/2011 at 10:39:47.
***** transmitting INFOFLSH.V9R1M0.ROLLOUT.SLIB.XMI2403 .
INMX000I 0 message and 53 data records sent as 1267 records to U.U
INMX001I Transmission occurred on 03/24/2011 at 10:39:47.
***** copying INFOFLSH.V9R1M0.ROLLOUT.MLIB.XMI2403 .
***** copying INFOFLSH.V9R1M0.ROLLOUT.REXX.XMI2403 .
***** copying INFOFLSH.V9R1M0.ROLLOUT.SLIB.XMI2403 .
***** transmitting INFOFLSH.V9R1M0.ROLLOUT.XMI2403 .
INMX000I 0 message and 57 data records sent as 17147 records to U.U
INMX001I Transmission occurred on 03/24/2011 at 10:39:49.
***** deleting INFOFLSH.V9R1M0.ROLLOUT.MLIB.XMI2403 .
***** deleting INFOFLSH.V9R1M0.ROLLOUT.REXX.XMI2403 .
***** deleting INFOFLSH.V9R1M0.ROLLOUT.REXX.XMI2403 .
***** deleting INFOFLSH.V9R1M0.ROLLOUT.SLIB.XMI2403 .
*****
* INFOFLSH.V9R1M0.ROLLOUT.XMI2403 ready for filetransfer binary *
```

Diese Datei kann nun versendet werden.

Alternativ: Receive EINES xmit files

Der Empfänger muss die Datei wie gewohnt mit BINARY FB 80 übertragen.

Zum ersten Entpacken benötigt man einen (manuell zu erstellenden) Job:

```
//. . . JOB ,CLASS=A                                A
//RECEIVE EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
  RECEIVE INDSN('INFOFLSH.V9R1M0.ROLLOUT.XMI2403')    B
  RESTORE DSN('IDBUNK1.V9R1M0.ROLLOUT.PO2403')        C
```

Anzupassen ist hier A: die Jobkarte
 B: der Name der BINARY hochgeladenen Datei
 C: ein beliebiger Dateiname für die Ausgabe

Ergebnis des Submit ist eine PO-Datei mit dem Namen aus C.

```
RECEIVE INDSN('INFOFLSH.V9R1M0.ROLLOUT.XMI2403')
INMR901I Dataset INFOFLSH.V9R1M0.ROLLOUT.PO2403 from IDBUNK1 on N1
INMR906A Enter restore parameters or 'DELETE' or 'END' +
INMR908A The input file attributes are: DSORG=PARTITIONED, RECFM=FB, BLKSIZE=2792
INMR909A You may enter DSNAME, SPACE, UNIT, VOL, OLD/NEW, or RESTORE/COPY/DELETE/
INMR001I Restore successful to dataset 'IDBUNK1.V9R1M0.ROLLOUT.PO2403'
```

In dieser Datei sind vier Member enthalten, unsere drei Dateien laut der Parameterliste und eine REXX.

```
IEB167I FOLLOWING MEMBER(S) LOADED FROM INPUT DATA SET REFERENCED BY SYS00004
IEB154I #IDRCVPO HAS BEEN SUCCESSFULLY LOADED
IEB154I MLIB HAS BEEN SUCCESSFULLY LOADED
IEB154I REXX HAS BEEN SUCCESSFULLY LOADED
IEB154I SLIB HAS BEEN SUCCESSFULLY LOADED
IEB1098I 4 OF 4 MEMBERS LOADED FROM INPUT DATA SET REFERENCED BY SYS00004
```

Nun noch mit P.3.4 in die Memberliste gehen und EX(ecute) als Zeilenkommando in der Zeile der REXX eingeben und ENTER.

```
ID12      BROWSE      INFOFLSH.V9R1M0.ROLLOUT.PO2403
Command ==>
      Name      Prompt      Size      Created
EX_____ #IDRCVPO      101    2011/03/01
_____ MLIB
_____ REXX
_____ SLIB
```

Damit erhält man nun die gewünschten Dateien mit der Namensregel:

Der ursprünglichen Datei-HLQ (hier IDBUNK1) wird durch die UserID dessen ersetzt, der das EX ausführt.

Die aktuellen Sourcen (#IDXMTPO und #IDRCVPO) befinden sich im zipfile zu xmit-sndrcv. Siehe InfoDesign Portal für registrierte Benutzer: www.idconsulting.de