

Job Control Language (JCL)

Tanfolyami jegyzet

•©ICSS Kft 2012

Bevezetés

- Batch JOB-ok írása Job Control Language használatával történik
- A JOB lépésekből (STEP) áll
- Az egyes lépések programokat indítanak el
- A programok adatállományokkal dolgozhatnak
- Az egyes lépések végrehajtása egymás után történik

JCL utasítások

- COMMAND
- Comment
- CNTL
- DD
- Delimiter
- ENDCNTL
- EXEC
- IF/THEN/ELSE/ENDIF
- INCLUDE
- JOB
- Null
- OUTPUT
- PEND
- PROC
- SET
- XMIT

JCL utasítások szintakszisa

- Egy vagy több 80 byte-os rekordból áll
- 1-71. oszlopig írhatók az utasítások
- 72. oszlop folytatósort jelző mező
- Folytatósort 4-16. oszlopig kell elkezdeni
- 73.-80. oszlop sorszámot tartalmazhat

JCL utasítások szerkezete

- Azonosító mező (1-2. oszlop)
 - // : jelentése JCL utasítás kezdődik
 - /* : elválasztást jelöl (pl. instream adat vége) vagy JES2 kontrol utasítást
 - /** : megjegyzés
- Név mező (3. oszlopon kezdődő)
 - 1-8 karakter hosszú alfanumerikus vagy speciális (#,\$,@) karakter
 - Első karakter betű vagy speciális (#,\$,@) karakter
 - A név mezőt szóköz követi

JCL utasítások szerkezete

- Műveleti mező
 - Itt adjuk meg a JCL utasítást
 - Előtte és utána szóköz következik
- Paraméter mező
 - A műveleti mezőt követi
 - Vesszővel elválasztott paraméterek, amelyek az egyes JCL utasításoknál különböznek
 - Pozicionális (adott sorrendben megadható, vesszővel jelezzük a helyét, amennyiben nem adjuk meg)
 - Kulcsszavas (kulcsszó és = jel után megadható alparaméter-értékek melyek megint csak lehetnek kulcsszavasak és pozicionálisak)
- Megjegyzés mező (paraméter mező után szóközzel elválasztva)

Paraméterek szerkezete

- Példák pozicionális és kulcsszavas paraméterek megadására

- `//NEV UTASITAS P1,P2,P3,K1=A,K2=B`

- Pozicionális paraméter kihagyása

- `//NEV UTASITAS P1,,P3,K1=A,K2=B`

- Utolsó pozicionális paraméter elhagyása

- `//NEV UTASITAS P1,P2,K1=A,K2=B`

- Összes pozicionális paraméter elhagyása

- `//NEV UTASITAS K1=A,K2=B`

- Kulcsszavas paraméter elhagyása

- `//NEV UTASITAS P1,,P3,K2=B`

Folytatósor

- Példák folytatósor írására:

```
//DATA DD DSN=ROBI.DATA,DISP=SHR,VOL=SER=SYSRES,  
//      UNIT=3390
```

- | ← a 4-16. oszlopig folytatódjon

- A folytatósornak // karakterekkel kell kezdődnie
- Megjegyzés folytatása:

72. oszlopon X folytatókarakter szerepel → |

```
//DATA DD DSN=ROBI.DATA,DISP=SHR    EZ AZ INPUT ADAT  X  
//      AMIT A SZAMLAZO PROGRAM HASZNAL  
      | ← a 4-16. oszlopig folytatódjon
```

- A fenti példában az X egy tetszőleges non-space karakter

Paraméter folytatása

- Példa folytatósor írására, ha egy aposztrófok közé írt paramétert kell folytatni:

```
-----1-----2---      ....      --+-----7-----+  
//STEP1 EXEC          PGM=PROG425, PARM=(COUNT, '/usr/lpp/in  
//                   ternet/server_root/admin_bin ....  ibm/we  
//                   bexec/applets/IBM/IMWCAPPL')
```

| ← a 16. oszlopon kell folytatódjon

- Írjuk a paramétert a 71. oszlopig
- A folytatósornak // karakterekkel kell kezdődnie
- A paramétert a 16. oszlopban kell folytatni

JOB utasítás

- Az első utasítás a JOB-ban
- Egy JOB-ban csak egy JOB utasítás lehet
- Szintakszis:

```
//JOBNEV JOB pozicionális_param,kulcsszavas_param
```

- Pozicionális paraméterek:
 - Számlázási információ
 - Programozó neve
- Kulcsszavas paraméterek:
 - CLASS, COND, MSGCLASS, MSGLEVEL, NOTIFY, TIME TYPRUN, USER, PASSWORD, REGION, RESTART, stb.

JOB utasítás kulcsszavas paramétere

- CLASS : a JOB melyik JES input osztályba kerüljön (A-Z, 0-9)
- MSGCLASS: a JOB log-jának JES output osztálya (A-Z, 0-9)
- MSGLEVEL: a JOB log-jába kerülő JCL és message üzenetek szintje
- NOTIFY: ki kapjon értesítést a JOB lefutásáról és a max. rc-ről
- TYPRUN: a JES hogyan dolgozza fel a JOB-ot (HOLD, SCAN)
- USER: kinek a nevében fusson a JOB
- PASSWORD: a USER paraméterben megadott felhasználó jelszava
- REGION: mekkora virtuális tárméretet kap a JOB
- RESTART: melyik lépés végrehajtásától induljon a JOB
- TIME: processzoridő használatának maximuma

JOB utasítás példák

- `//J1 JOB 1,ROBI,CLASS=F,MSGCLASS=A,MSGLEVEL=(1,1)`
- `//J2 JOB ,ROBI,TIME=(4,30),MSGCLASS=H,MSGLEVEL=(2,0)`
- `//J3 JOB ,ROBI,CLASS=3,MSGLEVEL=(1,1),COND=(8,LT)`
- `//J4 JOB 999,RESTART=STEP2,NOTIFY=ROBI`
- `//J5 JOB CLASS=A,MSGCLASS=R,NOTIFY=&SYSUID`
- `//J6 JOB ,ROBI,TYPRUN=SCAN,REGION=4M`
- `//J7 JOB ,ROBI,CLASS=A,USER=IBMUSER,PASSWORD=SECRET`
- `//J8 JOB '88/426-2012 Számlaszám.','Vári Róbert',
// LINES=(100,WARNING),PRTY=15,
// TIME=1440, ez a végtelen:)
// TYPRUN=HOLD,USER=ICROBI`

EXEC utasítás

- A JOB-lépés kezdetét jelzi
- Megadja az indítandó program vagy eljárás nevét
- Egy JOB-ban maximum 255 lépés lehet
- Szintakszis:

```
//STEPNEV EXEC pozicionális_param, kulcsszavas_param
```

- Pozicionális paraméterek:
 - PGM
 - PROC
- Kulcsszavas paraméterek:
 - COND, PARM, REGION, TIME, stb.

EXEC utasítás kulcsszavas paramétere

- COND : feltételes végrehajtás esetén használt paraméter
- PARM: a meghívott programnak átadott paraméterek
- REGION: mekkora virtuális tárméretet kap a lépésben meghívott program
- TIME: processzoridő használatának maximuma a lépésre vonatkozóan

EXEC utasítás példák

- `//STEP1 EXEC PGM=IEFBR14`
- `//STEP2 EXEC PGM=CALC,PARM='3,4'`
- `// EXEC PGM=IEBCOPY,REGION=2200K`
- `//PR1 EXEC ELJARAS,TIME=(2,30)`
- `//PROC2 EXEC PROC=ELJARAS,TIME=(2,30)`
- `//STEP3 EXEC PGM=WORKPGM,COND=(4,LT),`
`// PARM=(LIST,'LINECNT=121','Sáfrány u. 5.','Budapest')`

DD utasítás

- Input és output állományok leírására szolgál
- A programok számára szükséges állományokat az EXEC utasítás után, külön DD utasításokkal kell megadni
- Az EXEC után megadott DD utasítások sorrendje mindegy
- A DD nevére a név mező szabályai érvényesek
- A programban használt logikai file-neveknek meg kell feleljenek a DD nevek
- Szintakszis:
`//DDNÉV DD pozicionális_param, kulcsszavas_param`
- Pozicionális paraméterek:
 - *, DATA, DUMMY
- Kulcsszavas paraméterek:
 - DISP, DSNAMES, DBC, UNIT, VOLUME, SPACE, stb.

Instream adat

- A JOB-on belül megadható a feldolgozandó adat

- Szintakszis:

```
//INPUT DD *
```

```
    instream adat
```

```
/*
```

vagy

```
//INPUT DD DATA,DLM=ZZ
```

```
    instream adat
```

```
ZZ
```

Delimiter utasítás

- Az instream adat végét jelentő utasítás: /*
- A rendszer felismer más delimiterert is, ha a DLM paramétert kódoltuk a DD * vagy DD DATA utasításon.
- Példa beágyazott delimiter utasításokra:

```
//JOB      JOB
//STEPA    EXEC PGM=
//DD1      DD DATA,DLM=AA
//JOBBA    JOB
//STEP1    EXEC PGM=
//DD       DD *
  ADATOK
/*

AA
```

DD utasítás kulcsszavas paramétere

- ACCODE
- AMP
- AVGREC
- BLKSIZE
- BLKSIZLIM
- BURST
- CCSID
- CHARS
- CHKPT
- CNTL
- COPIES
- DATA
- DATACLAS
- DCB
- DDNAME
- DEST
- DISP
- DLM
- DSID
- DSNAME
- DSNTYPE
- DUMMY
- DYNAM
- EATTR
- EXPDT
- FCB
- FILEDATA
- FLASH
- FREE
- FREEVOL
- HOLD
- KEYLABL1
- KEYLABL2
- KEYENCD1
- KEYENCD2
- KEYLEN
- KEYOFF
- LABEL
- LGSTREAM
- LIKE
- LRECL
- MGMTCLAS
- MODIFY
- OUTLIM
- PATH
- PATHDISP
- PATHMODE
- PATHOPTS
- PROTECT
- RECFM
- REORG
- REFDD
- RETPD
- RLS
- SECMODEL
- SEGMENT
- SPACE
- SPIN
- STORCLAS
- SUBSYS
- SYSOUT
- TERM
- UCS
- UNIT
- VOLUME

DD utasítás kulcsszavas paramétere

- DSNNAME: megadja az állomány nevét. DSN-re rövidíthető
- UNIT: milyen típusú készülékre történjen az allokálás
- VOLUME: melyik lemezen van az állomány vagy hová allokálódjon
- SPACE: új állomány méretét adja meg
- DCB: Data Control Block az állomány jellemzőit adja meg, mint például logikai rekordhossz, blokkméret, rekordformátum
- LRECL: az állomány logikai rekordhosszát adja meg
- RECFM: az állomány rekordformátumát adja meg
- BLKSIZE: az állomány blokkméretét adja meg

DD utasítás kulcsszavas paramétere

- DISP: megadja az állomány diszpozícióját
- Szintakszis:
 - DISP=(initdisp,normaldisp,abnormaldisp)
- Initdisp: kezdetben az állományt hogyan alokálja a rendszer
 - NEW, OLD, SHR, MOD
- Normaldisp: a program normál lefutása esetén mi történjen az állománnyal
 - DELETE, KEEP, PASS, CATLG, UNCATLG
- Abnormaldisp: program abend esetén mi történjen az állománnyal
 - DELETE, KEEP, CATLG, UNCATLG

DD utasítás kulcsszavas paramétere

- SPACE: megadja az állomány méretét
- Használata:
 - `SPACE=({mertekegység,}(primary[,secondary][,directory][,RLSE])`
- Mertekegység: TRK, CYL, blokkhossz
- Primary: az állomány elsődleges mérete (a primary extent mérete, a létrehozáskor allokálható méret), az első paraméterként megadott mértékegységben
- Secondary-qty: amennyiben betelik az állomány, ekkora méretben extendálódjon (a secondary extent-ek mérete)
- Partícionált állomány (PDS) tartalomjegyzékének mérete (256 bytes directory blokkokban megadva)

SPACE paraméter példák

- SPACE=(TRK,(2,1))
- SPACE=(CYL,10)
- SPACE=(CYL,(10,,10),RLSE)
- SPACE=(4096,(50,20))
- SPACE=(1024,150)
- SPACE=(CYL,(4369,4369),RLSE)

DD utasítás kulcsszavas paramétere

- DCB: Data Control Block információ megadása
- Használata:
 - DCB=(alparaméter[,alparaméter]...)
- A főbb alparaméterek lehetnek:
 - RECFM
 - LRECL
 - BLKSIZE
 - BUFNO
 - KEYLEN
 - DSORG
- A DCB alparaméterek megadhatók önálló DD paraméterként is

DCB paraméter példák

- DCB=(RECFM=FB,LRECL=240,BLKSIZE=96M)
- DCB=(ICROBI.DATASET,KEYLEN=8)
- DCB=(*.STEP1.DD8,BUFNO=6)
- DCB=(DSORG=PO,LRECL=80,RECFM=F)

DD utasítás kulcsszavas paramétere

- Unit: a lemez vagy kazetta-egység típusát vagy címét adja meg
- Használata:
 - UNIT=([ddd][,unit-count][,DEFER])
 - UNIT=([device-type])
 - UNIT=AFF=ddname
- Példák:
 - Specifikus: UNIT=/1800 UNIT=1800
 - Generikus: UNIT=3390 UNIT=3490
 - Ezoteric: UNIT=VIO UNIT=SYSDA
 - Mount elhalasztása: UNIT=(3490,,DEFER)
 - Unit affinity: UNIT=AFF=DD1

DD utasítás kulcsszavas paramétere

- VOLUME: a volume információ megadására
- Használata:
 - VOLUME=(`[PRIVATE][,RETAIN][,vol-seq-num][,vol-count][,][SER=serial]`)
 - VOLUME=(`[PRIVATE][,RETAIN][,vol-seq-num][,vol-count][,][REF=*.ddname]`)
- Példák:
 - VOL=SER=MVSRES
 - VOL=SER=(ZP19X1,ZP19X2)
 - VOL=(,RETAIN,SER=123456)
 - VOL=(,,3,,SER=(111111,111112,111113,111114))

DD utasítás kulcsszavas paramétere

- SYSOUT: a DD-t spool output dataset-nek definiálja és hozzárendeli az állományt egy output osztályhoz
- Használata:
 - SYSOUT=osztály vagy SYSOUT=*
- Az osztály egy karakteres A-Z vagy 0-9, a „*” azt jelenti, hogy a JOB utasításnál megadott MSGCLASS paraméter osztályával egyezzen meg az osztály
- Az osztályhoz tartozó output diszpozíció a JES2-ben van definiálva
- Az output diszpozíció explicit megadása az OUTPUT utasítás OUTDISP paraméterében lehetséges

Output diszpozíció

- JES2 output osztályhoz tartozó paraméter vagy az OUTPUT utasítás OUTDISP paramétere által megadott érték
- Az output diszpozíció értékei lehetnek:
 - KEEP: a rendszer nyomtassa ki az output-ot, hagyja meg a spool területen az output-ot és a diszpozíciót változtassa LEAVE-vé
 - LEAVE: felengedés (release) után a diszpozíció változzon KEEP-pé
 - WRITE: a rendszer nyomtassa ki az output-ot és törölje a spool-ból
 - HOLD: a rendszer a spool-ban hagyja az output-ot és nem nyomtatja ki
 - PURGE: törli a spool-ból az output-ot

DD utasítás példák

- `//DD1 DD DSN=ROBI.DATASET1,DISP=OLD`
- `//PARMLIB DD DSN=SYS1.PARMLIB,DISP=SHR`
- `//INPUT DD DSN=SYS1.PARMLIB,DISP=SHR`
`// DD DSN=CPAC.PARMLIB,DISP=SHR`
- `//IN DD DSN=ROBI.NEW,VOL=SER=SYSRES,SPACE=(TRK,(2,1)),`
`// DISP=(,PASS),RECFM=FB,LRECL=200,BLKSIZE=8000,`
`// UNIT=SYSDA`
- `//DD2 DD DSN=*.STEPA.DD1,DISP=(OLD,PASS,DELETE)`
- `//DD3 DD DSN=&&TEMP,UNIT=3390,SPACE=(CYL,10)`

Speciális DD utasítások

- JOBLIB DD:
 - Egy darab lehet a JOB-ban, a JOB utasítás után, az első EXEC előtt és az összes STEP-re vonatkozik
 - Privát könyvtár megadására, ahol az EXEC utasításokban meghívott programokat keresi a rendszer. A privát könyvtár PDS vagy PDSE lehet
- STEPLIB DD:
 - Az egyes STEP-ekre vonatkozóan megadható privát könyvtár. A rendszer az adott lépésben meghívott programot először itt keresi. STEPLIB és JOBLIB DD együttes megadása esetén a STEPLIB-et veszi figyelembe annál a lépésnél amire a STEPLIB vonatkozik, a többi lépésnél pedig a JOBLIB-et

Speciális DD utasítások

- A rendszer készíthet dump-ot, ha egy lépés abnormális véget ér vagy ha abnormális véget érne, de a rendszer recovery eljárásai miatt normál terminálása lesz
- A készülő dump-ok fajtáját és jellemzőit a SYSABEND, SYSMDUMP és SYSUDUMP DD utasításokkal adhatjuk meg
- SYSUDUMP DD: felhasználói tárterületekről készülő dump, nyomtatható formájú
- SYSABEND DD: a felhasználói és rendszer tárterületek dumpja (LSQA és IOS kontrol blokkok), nyomtatható formában
- SYSMDUMP DD: a rendszer tárterületekről és a program address space-éről készülő dump, nem formatált. Az IPCS által feldolgozható

Megjegyzés utasítás

- `//*megjegyzés`
- A 4-80. oszlopig tetszőleges megjegyzést írhatunk
- A JOB-ban bárhol használható
- Megjegyzést írhatunk az egyes JCL utasítások mögött is a paramétermező után

Null utasítás

- A „//” utasítás jelzi a JOB végét
- Az első két karakter / jel, a 3-72. oszlopnak üresnek kell lennie
- Egy memberben több JOB is lehet; a null utasítás és a következő JOB utasítás közötti sorokat nem veszi figyelembe a rendszer
- A null utasítás használata nem kötelező

OUTPUT utasítás

- A SYSOUT állományokra vonatkozó feldolgozási opciókat adhatjuk meg az OUTPUT utasítás segítségével
- A megadott feldolgozási opciók akkor érvényesek, ha arra a SYSOUT DD utasításnál hivatkozunk
- A JES a SYSOUT DD utasításon megadott opciókat kombinálja a hivatkozott OUTPUT JCL utasításon megadott opciókkal
- Példa:

```
//OUT1   OUTPUT  DEST=ALAGSORI.PRINTER  
//OUT2   OUTPUT  CONTROL=DOUBLE  
//DD1    DD      SYSOUT=Z,OUTPUT=(*.OUT1,*.OUT2)
```

OUTPUT utasítás paramétere

- ADDRESS
- AFPPARMS
- AFPSTATS
- BUILDING
- BURST
- CHARS
- CKPTLINE
- CKPTPAGE
- CKPTSEC
- CLASS
- COLORMAP
- COMPACT
- COMSETUP
- CONTROL
- COPIES
- COPYCNT
- DATAACK
- DEFAULT
- DEPT
- DEST
- DPAGELBL
- DUPLEX
- FCB
- FLASH
- FORMDEF
- FORMLEN
- FORMS
- FSSDATA
- GROUPID
- INDEX
- INTRAY
- JESDS
- LINDEX
- LINECT
- MAILBCC
- MAILCC
- MAILFILE
- MAILTO
- MODIFY
- OFFSETXB
- OFFSETXF
- OFFSETYB
- OFFSETYF
- OUTBIN
- OUTDISP
- OVERLAYB
- OVERLAYF
- OVFL
- PAGEDEF
- PIMSG
- PORTNO
- PRMODE
- PRTATTRS
- PRTERORR
- PRTOPTNS
- PRTQUEUE
- PRTY
- REPLYTO
- RESFMT
- RETAINS
- RETRYL
- ROOM
- SYSAREA
- THRESHLD
- TITLE
- TRC
- UCS
- USERDATA
- USERLIB
- USERPATH
- WRITER

Hivatkozás OUTPUT utasításra

- JOB szintű OUTPUT utasítás a JOB utasítás után és az első EXEC utasítás előtt
- STEP szintű OUTPUT utasítás az EXEC utasítás után
- Az OUTPUT utasításra kétféle módon történhet hivatkozás:
 - Explicit mód: a SYSOUT DD utasítás tartalmazza az OUTPUT paramétert
 - Implicit mód: a JOB-ra vagy a lépésre vonatkozó OUTPUT utasítás érvényben van az OUTPUT utasításon megadott DEFAULT=YES paraméter miatt

Feltételes végrehajtás

- A job-ban az egyes lépések végrehajtása függhet feltételek vizsgálatától
- A lépések végrehajtása vagy kihagyása alapulhat előzőleg végrehajtott lépések kiértékelésétől, ekkor az IF/THEN/ELSE/ENDIF utasításokat használjuk
- A lépések végrehajtása vagy kihagyása vezérelhető return code-ok alapján a COND paraméter használatával is

Az IF/THEN/ELSE/ENDIF utasítás

- Szintakszis:

```
//név          IF (relációs kifejezés)   THEN
//STEPTRUE    EXEC
//név          ELSE
//STEPFALS     EXEC
//             ENDIF
```

- A relációs kifejezésben szerepelhet: & (AND), | (OR), ^ (NOT), =, >, <
- A lépések visszatérési értékére a lépésnév.rc-vel hivatkozhatunk
- Példa:

```
//TEST        IF (STEP1.RC<8)   THEN
//STEP2        EXEC PGM=JORC
//             ELSE
//STEP3        EXEC PGM=ROSSZRC
//             ENDIF
```

A COND paraméter

- Megadható a JOB és az EXEC utasításon is
- Használata:
 - COND=(code,operator)
 - COND=(code,operator[,stepname][.procstepname])
 - COND=EVEN: ha egy előző lépés abnormálisan terminált még akkor is hajtódjon végre a lépés
 - COND=ONLY: ha egy előző lépés abnormálisan terminált csak akkor hajtódjon végre a lépés
- Operátor lehet: GT, GE, EQ, LT, LE, NE
- Amennyiben a feltétel igaz a lépés NEM hajtódik végre

A COND paraméter

- Amennyiben a JOB és az EXEC utasításon is megadtunk COND paramétert, akkor ha a JOB utasításon megadott feltétel igaz a JOB terminál. Ha a JOB utasításon megadott feltétel nem teljesül, akkor a rendszer ellenőrzi az EXEC utasításon megadott feltételt.
- Az első lépés COND paraméterét mindig hamisnak értékeli ki a rendszer
- Példa:

```
//STEP2 EXEC PGM=IEBCOPY,COND=(4,LT,STEP1)
```

JES2 Kontrol utasítások

- A JES2 kontrol utasítások a JOB input és output feldolgozására vonatkoznak
- Formájuk:
 - /*utasítás paraméterek
- Főbb JES2 utasítások:
 - /*JOBPARM
 - /*MESSAGE
 - /*NOTIFY
 - /*OUTPUT
 - /*PRIORITY
 - /*ROUTE
 - /*XEQ
 - /*XMIT

JCL Könyvek

- z/OS V1R13.0 MVS JCL User's Guide
- z/OS V1R13.0 MVS JCL Reference