

TOMT、TODAコマンドについて

赤司, 房子
九州大学情報処理教育センター

梅田, 政信
九州大学工学部情報工学科

藤村, 直美
九州大学情報処理教育センター

<https://doi.org/10.15017/1468095>

出版情報：九州大学大型計算機センター広報. 17 (2), pp.75-83, 1984-03-22. 九州大学大型計算機センター

バージョン：

権利関係：

TOMT, TODAコマンドについて

赤司 房子,* 梅田 政信,** 藤村 直美*

1. はじめに

大型計算機センターではMTをTSSから直接使用することができず、MTを使うためには必ずバッチジョブとして実行しなければならない。しかしながらバッチジョブを実行するためには、ジョブ制御文(JCL)などについての知識を必要とし、手間がかかる上に間違いやすく簡単にはいかないことが多い。このため、一時的に不必要なデータセットをMTに退避させておきたい場合や、全データセットをMTに吸い上げてバックアップをとっておきたい場合に、TSSコマンドでもっと簡単に処理できるような機能があると便利である。ここでは、順データセット(PS)か区分データセット(PO)をディスクとMT間で簡単にやりとりできるようにするためのコマンドTOMT, TODAを紹介する。これらのコマンドは九州大学情報処理教育センターで作成し公開しているものを、大型計算機センターへ移し換えたものである。

2. 使用例

(1) 課題番号F000Aに属する全部のデータセットをMTにバックアップ後、再び復元する例
次の①～③の順序で行う。

① 課題番号F000Aでセッションを開き、TOMTコマンドを入力する。

```
READY
IQMT
```

ジョブが起動されるので、MTを取りつける。MTは2度取りつける。1度めはMTの初期設定を行うため、2度めは全部のデータセットと、それらを後でディスクへ戻すためのジョブ制御文が入っている作業データセット@TODA.CNTLをMTに吸い上げるためである。(ディスク→MT)

② 課題番号F000Aでセッションを開き、TODAコマンドを入力する。

```
READY
TODA
```

ジョブの起動後、①のMTを取りつけると、MTに吸い上げたデータセットをディスクへ戻すための作業データセット'F000A.@TODA.CNTL'が作成される。(MT→ディスク)

③ 'F000A.@TODA.CNTL'をSUBMITする。

```
READY
SUBMIT @TODA.CNTL
```

ジョブの起動後、①のMTを取りつけると、MT中にバックアップした全部のデータセットを元通りに移すことができる。(MT→ディスク)

この時②の処理でやめずに③まで行わなければデータセットを移すことができないことに注意しなく

* 九州大学情報処理教育センター

** 九州大学工学部情報工学科

この論文は JEF による原稿を写真版にしたものである

てはならない。処理が正常終了すれば、作業データセット 'F000A.@TODA.CNTL' は必要ないので削除しておく。この処理はMTに吸い上げたデータセットと同じ名前のデータセットがディスクにあると異常終了するので、その場合は(2)の③のように復元されるデータセット名を変更するなどの処理が必要である。

(2) 課題番号F000Aの一部のデータセットを課題番号F000Bに移す例

次の①～③の順序で行う。

- ① 課題番号F000Aでセッションを開き、TOMTコマンドを入力する。

```
READY
TOMT
```

(1)の①と同じである。TOMTコマンドでは一部のデータセットだけでなく、全部のデータセットをMTに吸い上げる。(ディスク→MT)

- ② 課題番号F000Bでセッションを開き、TODAコマンドを入力する。

```
READY
TODA
```

ジョブの起動後、①のMTを取りつくと、MTに吸い上げたデータセットをディスクへ戻すための作業データセット 'F000B.@TODA.CNTL' が作成される。(MT→ディスク)

- ③ 'F000B.@TODA.CNTL' 中の元の課題番号名を新しい課題番号に変更し、移す必要のないデータセットのための制御文を削除後、SUBMITする。この時、データセット名は自由に変更してよいが SQ=n (MT中のデータセット保存順序) を変更してはならない。また既存のデータセット名と重複する場合は、新しいデータセット名に変更しなければならない。

```
READY
EDIT @TODA.CNTL
```

```
KEQ52338I DATA SET @TODA.CNTL NOT LINE NUMBERED, NONUM ASSUMED
```

```
EDIT
RENUM
```

```
KEQ52569I WARNING, RENUMBERING DATA SET MAY CAUSE LOSS OF DATA+
HIT CARRIAGE RETURN TO RENUMBER, OR ENTER A NEW SUBCOMMAND-
- ..... RETURNのみ入力する。
```

```
EDIT
LIST
```

```
00010 //F000A1 JOB ,CLASS=N
00020 /**
```

(中 略)

```
00310 /**
00320 //MTPD PROC SQ=,DSN=
00330 // EXEC PGM=JSECOPI
00340 //SYSPRINT DD SYSOUT=*
00350 //SYSIN DD DUMMY
00360 //SYSUT1 DD DISP=SHR,VOL=(,RETAIN,SER=999999),
00370 // LABEL=(&SQ,SL),DSN=DS&SQ,UNIT=OPNMT
00380 //SYSUT2 DD DISP=(,CATLG),UNIT=PUB,
```

```

00390 //          SPACE=(TRK,(20,20,10),RLSE),DSN=&DSN
00400 //SYSUT3   DD UNIT=WORK,SPACE=(TRK,(1,1))
00410 //SYSUT4   DD UNIT=WORK,SPACE=(TRK,(1,1))
00420 //          PEND
00430 //*
00440 //S2 EXEC  MTPS,SQ=2,DSN='FOO0A.CAL.DATA'
00450 //S3 EXEC  MTPS,SQ=3,DSN='FOO0A.CAL.FORT77'
00460 //S4 EXEC  MTPO,SQ=4,DSN='FOO0A.CMDLIB.CLIST'
00470 //S5 EXEC  MTPO,SQ=5,DSN='FOO0A.EX.PASCAL'
00480 //S6 EXEC  MTPO,SQ=6,DSN='FOO0A.EX.PLI'
00490 //S7 EXEC  MTPO,SQ=7,DSN='FOO0A.EXAMPLE.FORT77'
00500 //S8 EXEC  MTPS,SQ=8,DSN='FOO0A.FLIST.FILE'
00510 //S9 EXEC  MTPO,SQ=9,DSN='FOO0A.F13TSS.CLIST'
00520 //
KEQ52500I END OF DATA SET

```

```

EDIT
C 10 510 /FOO0A/FOO0B/ ALL ..... 元の課題番号名を新しい課題番号に変更する。

```

```

EDIT
DEL 450 ..... 移す必要のないデータセットのための制御文を削除する。

```

```

EDIT
DEL 500 510

```

```

EDIT
C 460 /CMDLIB/TSS/ ..... データセット名CMDLIB.CLISTが課題番号FOO0Bに
                               存在している時は変更しておく。

```

```

EDIT
LIST

```

```

00010 //FOO0B1 JOB ,CLASS=N
00020 //*

```

(中 略)

```

00420 //          PEND
00430 //*
00440 //S2 EXEC  MTPS,SQ=2,DSN='FOO0B.CAL.DATA'
00460 //S4 EXEC  MTPO,SQ=4,DSN='FOO0B.TSS.CLIST'
00470 //S5 EXEC  MTPO,SQ=5,DSN='FOO0B.EX.PASCAL'
00480 //S6 EXEC  MTPO,SQ=6,DSN='FOO0B.EX.PLI'
00490 //S7 EXEC  MTPO,SQ=7,DSN='FOO0B.EXAMPLE.FORT77'
00520 //
KEQ52500I END OF DATA SET

```

```

EDIT
SUBMIT

```

ジョブが起動されるので、①のMTを取りつけると、MT中の課題番号FOO0Aの中の指定した一部のデータセットが課題番号FOO0Bに移される。(MT→ディスク)

3. コマンド説明

3. 1 TOMTコマンド

(1) 機能

自分の課題番号に属しているデータセット (PSとPO) を全部MTに吸い上げる。その際、それらのデータセットを後でディスクへ戻すためのジョブ制御文が入っている作業データセット@TODA. CNTLを作成し、それも同時にMTの先頭に吸い上げる。

研究開発

(2) コマンド形式

コマンド名	オペランド
TOMT	[Keep]

オペランドの説明

Keep : データセットをMTに吸い上げるための作業データセット@TOMT. CNTLを保存する場合に指定する。省略時は保存しない。

(3) 使用上の注意

① データセットは21文字以内の名前でなければならない。

例) F7001. EXAMPLE1. FORT77...21文字なのでよい

F7001. EXAMPLE1. LISP. TEXT...24文字なので処理できない

② 1つの課題番号に属するPSとPOのデータセットをMTに吸い上げる。その他の編成のデータセットは吸い上げない。

③ ジョブが起動されると、MTの取りつけ要求がサブコンソールに表示されるので、指示された機番にMTを取りつける。

④ 吸い上げたあとのMTの先頭には、実際のデータセットのほかに、ディスクに吸い上げたデータセットを戻すための作業データセット@TODA. CNTL (TODAコマンドで使用する)が入っている。

(4) 備考

使用コマンド : DSLISTコマンド

一時作業ファイル : @TOMT. CNTL, @WORK

保存作業ファイル : @TODA. CNTL

3. 2 TODAコマンド

(1) 機能

TOMTコマンドでMTに吸い上げているデータセット (PSかPOに限る) をディスクへ戻すためのデータセット'課題番号. @TODA. CNTL' (ジョブ制御文からなる) をMTからロードする。

(2) コマンド形式

コマンド	オペランド
TODA	なし

(3) 使用上の注意

- ① MTはTOMTコマンドで吸い上げたものに限る。
- ② このコマンドはMTの先頭の作業データセットを取り出して、' 課題番号. @TODA. CNTL' を作成するところまで実行する。実際にMTからデータセットをディスクへ戻すには、自分で' 課題番号. @TODA. CNTL' の中の課題番号を書き替えたり、データセット名が重複する場合はデータセット名を書き替えるなどの任意の編集を行った後、SUBMITしなければならない。

(4) 備考

保存作業ファイル : @TODA. CNTL

一時作業ファイル : @WORK

4. 他センターへデータセットを移す場合

TODAコマンドで作成したジョブ制御文(@TODA. CNTL)中の課題番号、パスワード、UNIT名、JOBクラスなどの変更が必要である。(2. (2)参照)

例えば情報処理教育センターにおいては、以下の変更が必要となる。

大型計算機センター		情報処理教育センター
① 課題番号	⇒	課題番号
② パスワードなし	⇒	パスワードの指定が必要
③ CLASS=N	⇒	CLASS=B
④ UNIT=OPNMT	⇒	UNIT=TAPE

5. おわりに

MTの利用目的としてデータセットのバックアップだけでなく、大型計算機センターのようにデータセットの保存に課金される所では、長期に使用しないデータセットをMTに退避させておくことは使用料金の節約にもなる。しかしながらMTを使用する作業は、手間がかかるためつい億劫になりがちである。ここで紹介したTOMT, TODAコマンドを使用すれば、TSSコマンドで比較的簡単にMTとディスク間でデータセットのやりとりを行うことができる。このコマンドで作成したMTはIBM互換のある計算機ではいずれも容易に読み書きが可能である。このコマンドはコマンドプロシジャを用いて作成しており、移植性も考慮している。従って他センターにおいてもこれらのコマンドを個人的に所有しておけば、複数のセンター間でデータセットのやりとりを簡単に行うこともできる。プログラムのリストを図1, 2, 3に示す。

```

00010 PROC 0 UID() KEEP #DEBUG
00020 /*****
00030 /* PROGRAM TOMT */
00040 /*          VO1/LO1  CREATED BY F.AKASHI ON 83/09/20 */
00050 /*          UPDATED BY M.UMEDA ON 83/11/01 */
00060 /*****
00070 SET &NILL =
00080 CONTROL NOFLUSH NOLIST NOMSG
00090 IF &#DEBUG = &NILL THEN CONTROL LIST MSG CONLIST NOSYMLIST
00100
00110 /* ENVIRONMENT DEPENDENT PART */
00120 SET &DSLST = &STR(CALLM #DSLST)
00130 SET &CATAPRO = &STR('F2018.SOURCE.CNTL(KERNEL)')
00140 SET &JOBCLASS = &STR(N) /* MT JOB CLASS */
00150 SET &PSW = &STR(UNNECESSARY) /* PASSWORD */
00160 SET &ID = &STR(&SYSUID)
00170
00180 /* INITIALIZE */
00190 SET &TOMT = &STR(@TOMT.CNTL) /* WORK DATASET FOR TOMT */
00200 SET &TODA = &STR(@TODA.CNTL) /* WORK DATASET FOR TODA */
00210
00220 /* READ PASSWORD FROM TERMINAL IF NECESSARY */
00230 IF &STR(&PSW) = &STR(NECESSARY) THEN +
00240 DO
00250 WRITENR ENTER YOUR PASSWORD =>
00260 READ &PSW
00270 END
00280 ELSE SET &PSW = &NILL
00290
00300 /* GET DSLIST */
00310 ATTN GOTO ENDDO
00320
00330 FREE AT(#DSLST #FTO6)
00340 ATTR #DSLST BL(137)
00350 ATTR #FTO6 BL(141) LR(137) REC(V B A)
00360 ALLOC F(FT90F001) NEW US(#DSLST) REU
00370 LISTC O(FT90F001) L('&ID')
00380
00390 ALLOC F(FTO6F001) NEW REU US(#FTO6)
00400 &DSLST
00410
00420 /* CREATING WORK DATASETS */
00430 ATTN DO /* DISABLE ATTENTION */
00440 SET &CMD = &NILL
00450 &CMD
00460 END
00470 FREE AT(#CNTL)
00480 ATTR #CNTL BL(3120) LR(80) REC(F B)
00490
00500 ERROR DO
00510 ALLOC F(OUTDD1) DA(&TOMT) SHR REU US(#CNTL)
00520 RETURN
00530 END
00540 ALLOC F(OUTDD1) DA(&TOMT) NEW REU SP(2 2) T US(#CNTL)
00550
00560 ERROR DO
00570 ALLOC F(OUTDD2) DA(&TODA) SHR REU US(#CNTL)
00580 RETURN
00590 END
00600 ALLOC F(OUTDD2) DA(&TODA) NEW REU SP(2 2) T US(#CNTL)
00610
00620 ERROR GOTO IOER1
00630 OPENFILE OUTDD1 OUTPUT
00640 OPENFILE OUTDD2 OUTPUT
00650 SET &OUTDD1 = &STR(//&SYSUID.A JOB &PSW,CLASS=&JOBCLASS)
00660 SET &OUTDD2 = &STR(//&SYSUID.A JOB &PSW,CLASS=&JOBCLASS)
00670 PUTFILE OUTDD1
00680 PUTFILE OUTDD2
00690 IOER1: ERROR OFF
00700 CLOSFIL OUTDD1
00710 CLOSFIL OUTDD2
00720 FREE F(OUTDD1 OUTDD2)
00730
00740 /* APPENDING KERNEL JCL TO &TOMT AND &TODA */
00750 MERGE &STR(&CATAPRO) &TOMT
00760 MERGE &STR(&CATAPRO) &TODA
00770
00780 /* PICK UP DSNAME & ORG(PSIPO) FROM OUTPUT OF DSLIST */

```

```

00790 ERROR GOTO IOER2
00800
00810 ALLOC F(OUTDD1) DA(&TOMT) MOD REU
00820 ALLOC F(OUTDD2) DA(&TODA) MOD REU
00830 OPENFILE FT06F001 INPUT /* OUTPUT OF DSLIST */
00840 OPENFILE OUTDD1 OUTPUT
00850 OPENFILE OUTDD2 OUTPUT
00860
00870 SET &OUTDD1 = &STR(//INIT EXEC MTINIT,VOL=999999,+
00880 LABEL=SL,DEN=4)
00890 PUTFILE OUTDD1
00900 SET &OUTDD1=&STR(//LOADER EXEC PSMT,SQ=1,DSN='&SYSUID..&TODA')
00910 PUTFILE OUTDD1
00920
00930 SET &I = 2
00940 SET &IDLEN = &LENGTH(&ID)
00950 DO WHILE (1 = 1)
00960 GETFILE FT06F001
00970 SET &SRC = &STR(&FT06F001)
00980 IF &LENGTH(&STR(&SRC)) >= 65 THEN +
00990 IF &STR(&SUBSTR(2:&IDLEN+1,&STR(&SRC))) = &ID THEN +
01000 DO
01010 SET &DRG = &SUBSTR(61:62,&STR(&SRC))
01020 SET &DSN = &SUBSTR(&IDLEN+3:&IDLEN+18,&STR(&SRC))
01030
01040 /* TRIMMING DATASET'S NAME */
01050 SET &LEN = &LENGTH(&STR(&DSN))
01060 SET &CHR = &SUBSTR(&LEN:&LEN,&STR(&DSN))
01070 DO WHILE (&LEN > 1) && (&STR(&CHR) = &STR( ))
01080 SET &DSN = &SUBSTR(1:&LEN-1,&STR(&DSN))
01090 SET &LEN = &LEN - 1
01100 SET &CHR = &SUBSTR(&LEN:&LEN,&STR(&DSN))
01110 END
01120
01130 IF ( &STR(&ORG) = &STR(PS) |
01140 &STR(&ORG) = &STR(PO) ) && +
01150 ( &STR(&DSN) ^= &STR(&TOMT) ) && +
01160 ( &STR(&DSN) ^= &STR(&TODA) ) THEN +
01170 DO
01180 SET &OUTDD1=&STR(//S&I EXEC &ORG.MT,SQ=&I,+
01190 DSN='&ID..&STR(&DSN)')
01200 SET &OUTDD2=&STR(//S&I EXEC MT&ORG,SQ=&I,+
01210 DSN='&ID..&STR(&DSN)')
01220 PUTFILE OUTDD1
01230 PUTFILE OUTDD2
01240 SET &I = &I + 1
01250 END
01260 END
01270 END
01280
01290 IOER2: ERROR OFF
01300 SET &OUTDD1 = &STR(//)
01310 SET &OUTDD2 = &STR(//)
01320 PUTFILE OUTDD1
01330 PUTFILE OUTDD2
01340 CLOSFILE FT06F001
01350 CLOSFILE OUTDD1
01360 CLOSFILE OUTDD2
01370 FREE F(FT06F001 OUTDD1 OUTDD2) DA(&TODA)
01380
01390 /* SUBMIT TOMT JOB */
01400 SUBMIT &TOMT
01410
01420 ENDDO: ERROR OFF
01430 ATTN OFF
01440 FREE F(FT90F001 FT06F001) AT(#DSLIST #FT06 #CNTL)
01450 FREE DA(&TODA)
01460 ALLOC F(FT06F001) DA(*) REU
01470
01480 IF &KEEP = &NILL THEN DELETE &TOMT
01490 EXIT

```

☒ 1. TOMT コマンドリスト

```

00010 PROC 0 #DEBUG
00020 /*****
00030 /* PROGRAM TODA */
00040 /*          VO1/LO1  CREATED BY F.AKASHI ON 83/09/20 */
00050 /*          UPDATED BY M.UMEDA  ON 83/11/01 */
00060 /*****
00070 SET &NILL =
00080 CONTROL NOFLUSH NOLIST NOMSG
00090 IF &#DEBUG ^= &NILL THEN CONTROL LIST MSG CONLIST NDSYMLIST
00100
00110 /* ENVIRONMENT DEPENDENT PART */
00120 SET &CATAPRO = &STR('F2018.SOURCE.CNTL(KERNEL)')
00130 SET &JOBCLASS = &STR(N) /* MT JOB CLASS */
00140 SET &PSW = &STR(UNNECESSARY) /* PASSWORD */
00150
00160 /* INITIALIZE */
00170 SET &TODA = &STR(@TODA.CNTL) /* OLD DATASET IN MT */
00180 SET &WORK1 = &STR(@WORK1) /* WORK DATASET FOR IPL */
00190
00200 /* READ PASSWORD FROM TERMINAL IF NECESSARY */
00210 IF &STR(&PSW) = &STR(NECESSARY) THEN +
00220 DO
00230 WRITENR ENTER YOUR PASSWORD =>
00240 READ &PSW
00250 END
00260 ELSE SET &PSW = &NILL
00270
00280 /* CREATING WORK DATASET FOR INITIAL PROGRAM LOADER */
00290 ATTN GOTO ENDDO
00300
00310 DELETE &TODA
00320 FREE AT(#CNTL)
00330 ATTR #CNTL BL(3120) LR(80) REC(F B)
00340
00350 ERROR DO
00360 ALLOC F(OUTDD) DA(&WORK1) SHR REU US(#CNTL)
00370 RETURN
00380 END
00390 ALLOC F(OUTDD) DA(&WORK1) NEW REU SP(2 2) T US(#CNTL)
00400
00410 ERROR GOTO IOER1
00420 OPENFILE OUTDD OUTPUT
00430 SET &OUTDD = &STR(//&SYSUID.A JOB &PSW,CLASS=&JOBCLASS)
00440 PUTFILE OUTDD
00450 IOER1: ERROR OFF
00460 CLOSFIL OUTDD
00470 FREE F(OUTDD)
00480
00490 /* APPENDING KERNEL JCL TO &WORK1 */
00500 MERGE &STR(&CATAPRO) &WORK1
00510
00520 /* MAKING EXEC STATEMENT */
00530 ALLOC F(OUTDD) DA(&WORK1) MOD REU
00540
00550 ERROR GOTO IOER2
00560 OPENFILE OUTDD OUTPUT
00570 SET &OUTDD= &STR(//IPL EXEC MTPS,SQ=1,DSN='&SYSUID..&TODA')
00580 PUTFILE OUTDD
00590 SET &OUTDD= &STR(//)
00600 PUTFILE OUTDD
00610 IOER2: ERROR OFF
00620 CLOSFIL OUTDD
00630 FREE F(OUTDD)
00640
00650 /* SUBMIT INITIAL PROGRAM LOADER */
00660 SUBMIT &WORK1
00670
00680 ENDDO: ERROR OFF
00690 ATTN OFF
00700 FREE AT(#CNTL)
00710 DELETE &WORK1
00720 EXIT

```

図 2. TODA コマンドリスト

```

/**
//PSMT   PROC   SQ=,DSN=
//        EXEC   PGM=JSDGENER
//SYSPRINT DD  SYSOUT=*
//SYSIN   DD   DUMMY
//SYSUT1  DD   DISP=SHR,DSN=&DSN
//SYSUT2  DD   DISP=(,KEEP),VOL=(,RETAIN,SER=999999),
//        LABEL=(&SQ,SL),DSN=DS&SQ,UNIT=OPNMT
//        PEND
/**
//POMT   PROC   SQ=,DSN=
//        EXEC   PGM=JSECOPY
//SYSPRINT DD  SYSOUT=*
//SYSIN   DD   DUMMY
//SYSUT1  DD   DISP=SHR,DSN=&DSN
//SYSUT2  DD   DISP=(,KEEP),VOL=(,RETAIN,SER=999999),
//        LABEL=(&SQ,SL),DSN=DS&SQ,UNIT=OPNMT
//SYSUT3  DD   UNIT=WORK,SPACE=(TRK,(1,1))
//        PEND
/**
//MTPS   PROC   SQ=,DSN=
//        EXEC   PGM=JSDGENER
//SYSPRINT DD  SYSOUT=*
//SYSIN   DD   DUMMY
//SYSUT1  DD   DISP=SHR,VOL=(,RETAIN,SER=999999),
//        LABEL=(&SQ,SL),DSN=DS&SQ,UNIT=OPNMT
//SYSUT2  DD   DISP=(,CATLG),UNIT=PUB,
//        SPACE=(TRK,(20,20),RLSE),DSN=&DSN
//        PEND
/**
//MTPO   PROC   SQ=,DSN=
//        EXEC   PGM=JSECOPY
//SYSPRINT DD  SYSOUT=*
//SYSIN   DD   DUMMY
//SYSUT1  DD   DISP=SHR,VOL=(,RETAIN,SER=999999),
//        LABEL=(&SQ,SL),DSN=DS&SQ,UNIT=OPNMT
//SYSUT2  DD   DISP=(,CATLG),UNIT=PUB,
//        SPACE=(TRK,(20,20,10),RLSE),DSN=&DSN
//SYSUT3  DD   UNIT=WORK,SPACE=(TRK,(1,1))
//SYSUT4  DD   UNIT=WORK,SPACE=(TRK,(1,1))
//        PEND
/**

```

図3. KERNELジョブ制御文リスト

参考文献

1. 赤司房子, 梅田政信, 藤村直美 TOMT, TODAコマンドについて, 情報処理教育センター広報, 6, 2, 1984。
2. 計算機マニュアル FACOM OSIV ジョブ制御言語文法書 (64SP-1062-3), 富士通㈱。
3. 計算機マニュアル FACOM OSIV データセットユーティリティ使用手引書 (64SP-3172-2), 富士通㈱。