NASA CONTRACTOR REPORT 166499

Basic JCL for the CRAY-1 Operating System (COS) With Emphasis on Making the Transition from CDC 7600/SCOPE

Greg Howe
David Saunders

(NASA-CR-166499) BASIC JCL FOR THE CRAY-1 OPERATING SYSTEM (COS) WITH EMPHASIS ON MAKING THE TRANSITION FROM CDC 7600/SCOPE (Informatics, Inc., Palo Alto, Calif.) 29 p HC A03/MF A01 CSCL 09B G3/61

N83-30053

Unclas 12833

CONTRACT NAS2-11555
July 1983

NASA

NASA CONTRACTOR REPORT 166499

Basic JCL for the CRAY-1 Operating System (COS) With Emphasis on Making the Transition from CDC 7600/SCOPE

Greg Howe
David Saunders
Informatics General Corporation
1121 San Antonio
Palo Alto, CA

Prepared for
Ames Research Center
Under Contract NAS2-11555

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

CONTENTS

1.0 INTRODUCTION

These pages are intended to help those already familiar with the CDC 7600 to make the switch to the CRAY-1. The more common COS (CRAY-1 Operating System) JCL statements are compared where possible with the equivalent SCOPE 2.1 JCL. Sample decks for the more common jobs are shown in parallel, with notes.

Transferring source codes via magnetic tape is touched on, but no attempt is made to illuminate other magnetic tape file handling.

Finally, the more obvious differences in UPDATE and FORTRAN to beware of when making the transition are summarized.

Consider this a handy reference only, for that trying period typical of transitions to new systems. Read the proper manuals and other literature eventually (sooner rather than later...).

Some useful miscellaneous information follows:

TELEPHONE NUMBERS:

ACF Consultants	ext 6515, Blg 233A, Rm 176
CRAY-1 Operator	ext 6412, Blg 233A, Rm 185

DOCUMENTATION:

All documentation for the Advanced Computational Facility can be obtained through the Computer Information Center, Building 233, Rooms 126-128, ext 6035, Mail Stop 233-13.

USER ID'S:

Call the ACF User Representative, ext 6515, for the appropriate form.

JOB ORDER NUMBERS:

Use Job Order Request form ARC 201 to apply for an ACF j.o. number.
Use Computer Service Request form ARC 159 to commit funds in advance.

2.0 LOCAL DATASETS AND PERMANENT DATASETS (FILES): SCOPE VS. COS

The local or logical file names (LFNs) of SCOPE 2.1 are termed (LOCAL) DATASET NAMES (DNs) under COS. Permanent file names are termed PERMANENT DATASET NAMES (PDNs).

The hierarchies for the structures of files or datasets on the two systems are as follows:

CDC 7600	CRAY-1
FILE	DATASET
PARTITION	FILE
SECTION	
RECORD	RECORD

The commonly defaulted COS local dataset names are shown below with their SCOPE 2.1 equivalents.

CDC 7600 LFNs	CRAY-1 DNs	Description
INPUT	$IN	(job input stream)
OUTPUT	$OUT	(printable output)
OLDPL	$PL	(Existing UPDATE library)
NEWPL	$NPL	(New UPDATE library)
SOURCE	$SR	(Source file from UPDATE - common decks not expanded)
COMPILE	$CPL	(Compile file from UPDATE - common decks expanded)
LGO	$BLD	(Binary load file, output from compiler)
TAPE7	FT07	(Associates dataset with FORTRAN logical unit number)
TAPE10	FT10	(Note use of exactly 4 characters, from FT00 through FT99)

(Points to note appear on the next page.)

Points to note concerning dataset names include the following:

* Local dataset names may be 1-7 alphanumeric characters, the first of which must be A-Z, $, %, or @ (i.e., non-numeric). Average users should refrain from using $ as the first character, however.

  WARNING: Sometimes local dataset names lead to file names on a PDP-11 or VAX. (The same is true of JOB names.) Trouble has been encountered with such names less than 5 characters, so 5-7 characters are always safe.

* Permanent dataset names (PDNs) may be 1-15 characters, all of which may be alphanumeric, even the first.

  WARNING: If your PDN exceeds 15 characters in length on a SAVE statement (cf. CATALOG under SCOPE), the dataset (and possibly a lot of CPU time) is LOST - COS does not truncate the PDN or do anything else helpful...

* There is no practical limit on the number of EDitions (cf. CYcles, with limit of 5) of a given PDN on the system at the same time. (Actually, ED=1 through 4095 is valid.)

3.0 THE MORE COMMON JCL STATEMENTS: SCOPE VS. COS

The following shows the common CDC 7600/SCOPE job control statements and the corresponding CRAY-1/COS job control statements, if any.

CDC 7600	CRAY-1
"Job Card"	JOB...
ACCOUNT...	ACCOUNT...
MOUNT...	<None -- no private packs>
ATTACH...	ACCESS...
LIBRARY...	"LIB =" parameter on LDR statement.
FTN...	CFT...
LGO...	LDR...
EXIT.	EXIT.
EXIT,U.	<No equivalent, although most statements permit a No Abort parameter, NA.>
FILE.../REQUEST...	ASSIGN... <Not strictly equivalent.>
STAGE,PRE...	ACQUIRE,...
STAGE,POST...	DISPOSE,...
CATALOG...	SAVE...
PURGE...	DELETE...
RETURN...	RELEASE...
COPY...	COPYD...
COPYS.../COPYP...	COPYF...
COPYR...	COPYR...
COPYSP...	COPYS... <Echoing data requires shifted copies - see 5.9>
SKIPF.../SKIPB...	SKIPF...
REWIND...	REWIND...
AUDIT...	AUDIT...
COMMENT. <text>	* <text>

(Points to note appear on the next page.)

Points to note include the following:

* The syntax of COS is very like that of SCOPE 2.1. For instance, INITIAL SEPARATOR and STATEMENT TERMINATOR may be either COMMA/PERIOD or LEFT/RIGHT PARENTHESIS on both systems:

```
VERB,PARAMETER LIST.        and
VERB(PARAMETER LIST)
VERB.
```

  are all valid on both systems. (See also the syntax for COMMENTs below.)

* Both POSITIONAL and KEYWORD parameters will be seen, though KEYWORD parameters tend to be more widely used.

* Use of the COLON is new: it allows concatenation in association with keywords. For example: LDR,LIB=INSL:DISSPLA.

* BLANKs are ignored unless they are embedded in a LITERAL STRING, which is delimited by APOSTROPHEs: 'literal string'.

* Control statements may be CONTINUED beyond 80 characters by using the CARET character (shift-G) on the 029 card-punch, or CIRCUMFLEX (^) at a terminal. Exceptions are the JOB, ACCOUNT, DUMPJOB and EXIT statements. (Users of RJESND should use left square bracket, "["; card punchers should use the cent sign ("c" with a slash).)

* COMMENTs are indicated by the ASTERISK (*) as the first non-blank character in a statement (no terminating period needed).

* Characters after the statement terminator are ignored, and may also be used as COMMENTs. (They do appear in the logfile.)

* Characters after the statement continuation character are also ignored.

4.0 CRAY-1 JOB DECK STRUCTURE

The card decks or card image files for running batch jobs on the CRAY-1 are basically the same as for the CDC 7600. Two of the details depend on which front-end machine the job is being submitted to the CRAY through. These are the end-of-section and end-of-file markers, and the initial JOB cards:

CDC 7600			CRAY-1			Description
CARD DECK	C11 (VAX/PDP-11)	SUB (VAX)	CARD DECK	RJESND (PDP-11)	CSUB (VAX)	Mode of transmission
CYBER	PDP-11	VAX	CYBER	CYBER	VAX	Front-end machine
No	No	No	Yes	Yes	No	Two sets of job cards?
7/8/9	%%EOR	%%EOR	7/8/9	%%EOR	/EOF	End-of-section
6/7/8/9	%%EOF	%%EOF	6/7/8/9	%%EOF	/EOF	End-of-file

The additional job/user cards (first two in the deck) needed for the CYBER 170/720 front end to the CRAY-1 are as follows:

	DESCRIPTION
myjob,MICR.	CYBER JOB card. ("Machine Identifier CRay")
USER,myuserid,mypassword.	CYBER USER card.

The job name, user-id, and password, MUST MATCH those on the next (3rd) card (see below). The job name may be 1-7 alphanumeric characters. The password must be AT LEAST 4 characters. (What is the upper limit? At least 6.)

WARNING: Use at least 5 characters for the job name unless you are working with card decks.

The basic CRAY job deck setup is as follows (preceded by the above two cards unless the VAX front-end is used):

	DESCRIPTION
JOB,JN=myjob,T=20,US=myuserid,PW=mypassword. CS=10,NA=myname,X=myext ACCOUNT,AC=myjonumber.	CS,NA,X comments are for I/O handling CRAY ACCOUNT card
<Rest of CRAY JCL>	UPDATE, CFT, etc.
7/8/9	Appropriate end-of-section marker
<Source code>	FORTRAN, or inputs to UPDATE, say
7/8/9	Another end-of-section marker
<Data>	If any
6/7/8/9	End-of-file marker

5.0 EXAMPLES

5.1 Simple Compile And Execute

CDC 7600

```
myjob,T5.                          myname, X myext
ACCOUNT,myuserid,myjonumber.
FTN,R=3,OPT=2.
LGO.
7/8/9
<FORTRAN>
7/8/9
<data>
6/7/8/9
```

CRAY-1

```
myjob,MICR.                              (Unless VAX front-end)
USER,myuserid,mypassword.                ( "     "     "    " )
JOB,JN=myjob,T=5,US=myuserid,PW=mypassword.  CS=10,NA=myname,X=myext
ACCOUNT,AC=myjonumber.
CFT,ON=ADIX.
LDR,MAP=PART,SET=INDEF.
7/8/9
<FORTRAN>
7/8/9
<data>
6/7/8/9
```

NOTES:

* On the CFT card, the ON=ADIX parameter indicates overriding of four defaults as recommended from experience. The default is to CONTINUE after compiling, even if there were errors. ON=A causes an ABORT if there were compile errors. The X gives the cross reference listing that is recommended. The I adds compiler-generated statement labels to the symbol table printed with the cross references, and these can assist locating execution errors using the Program Counter. The D adds listing of the DO-loop table to the cross-reference listing, handy for the same reason.

* On the LDR card, the default loader map option is OFF. MAP=ON or MAP=FULL give a bulky map. MAP=PART (or just MAP), is the nearest to the default for the 7600.

* On the LDR card, SET=INDEF is recommended during code development. (The default for presetting memory is NOT to preset it to anything at load-and-go time. SET=ZERO is NOT recommended, as it tends to obscure programming errors.)

5.2 Saving Data On Disk

(Cards marked with an asterisk (*) in the following examples are the subject of the example; lower case is used to indicate an application-specific string.)

CDC 7600

```
myjob,T5.                          myname, X myext
ACCOUNT,myuserid,myjonumber.
REQUEST,tape2,SN=system,*PF.
FTN,R=3,OPT=2.
LGO.
CATALOG,tape2,testdata,ID=myuserid.
7/8/9
```

CRAY-1

```
myjob,MICR.                        (Unless VAX front-end)
USER,myuserid,mypassword.          ( "     "     "    " )
JOB,JN=myjob,T=5,US=myuserid,PW=mypassword.  CS=10,NA=myname,X=myext
ACCOUNT,AC=myjonumber.
* ASSIGN,DN=results,A=FT02.
CFT,ON=ADIX.
LDR,MAP=PART,SET=INDEF.
* SAVE,DN=results,PDN=testdata,ID=myuserid.
7/8/9
```

NOTES:

* On the ASSIGN card, use of A=FT02 ("alias=...") is preferable to ASSIGN,DN=FT02 followed by SAVE,DN=FT02,...
* On the SAVE card, for public storage monitoring purposes, the Advanced Computational Facility is requesting that IDs associated with PDNs match valid user-ids (USs).
* There is no practical limit on the number of editions of given PDN (SCOPE limit is 5), but remember: permanent disk storage is at a premium on the CRAY.

5.3 Using Existing Disk Datasets

CDC 7600

```
myjob,T5.                                    myname, X myext
ACCOUNT,myuserid,myjonumber.
ATTACH,tape2,testdata,SN=system,ID=myuserid,CY=1.
FTN,R=3,OPT=2.
LGO.
7/8/9
```

CRAY-1

```
myjob,MICR.                                  (Unless VAX front-end)
USER,myuserid,mypassword.                    ( "     "     "    " )
JOB,JN=myjob,T=5,US=myuserid,PW=mypassword.  CS=10,FA=myname,X=myext
ACCOUNT,AC=myjonumber.
* ACCESS,DN=FT02,PDN=testdata,ID=myuserid,ED=1.
CFT,ON=ADIX.
LDR,MAP=PART,SET=INDEF.
7/8/9
```

NOTES:

* On the ACCESS card, if the edition number is omitted, the default is highest edition.
* DN=FT02 associates the existing file with FORTRAN logical unit number 2.

5.4 Deleting And Auditing Disk Datasets

CDC 7600

```
myjob,T5.                          myname, X myext
ACCOUNT,myuserid,myjonumber.
ATTACH,kill,testdata,SN=system,ID=myuserid,CY=1,MR=0.
PURGE,kill.
EXIT,U.
AUDIT,SN=system,ID=myuserid.
6/7/8/9
```

CRAY-1

```
myjob,MICR.                                (Unless VAX front-end)
USER,myuserid,mypassword.                  ( "    "    "    " )
JOB,JN=myjob,T=5,US=myuserid,PW=mypassword.  CS=...,NA=...,X=..
ACCOUNT,AC=myjonumber.
* ACCESS,DN=kill,PDN=testdata,ID=myuserid,ED=1,UQ.
* DELETE,DN=kill,NA.

* AUDIT,PDN=-,ID=myuserid.
  6/7/8/9
```

NOTES:

* For ACCESS, the UQ parameter specifies unique access (same as MR=0). (If LE (for Lowest Edition) is used in place of ED=..., the equivalent of SCOPE's LC=1 feature is obtained, for avoiding precise specification of cycle number. However, LE has yet to be fully implemented, apparently.)

* For DELETE, the NA parameter will prevent the job from aborting because of error(s) (effectively the same as EXIT,U). NA is valid for many of the JCL statements, but it should be used judiciously - frequently you don't WANT to proceed if something was wrong.

* PDN=- on the AUDIT card will list all the permanent datasets with ID=... (as specified). The PDNs can be partially specified in a variety of ways, such as PDN=ABC- for all permanent dataset names beginning with ABC, and PDN=-A**- for those containing the letter A followed by at least two other characters. AUDIT(ID=myuserid) is also valid.

* Ames's SOFTLIB, accessible over DECnet, contains reference to a utility for generating the bulk of the JCL needed to delete permanent datasets on the CRAY (or the CDC 7600). KILLJCL runs on a VAX. It can save a lot of tedious work.

5.5 Creating A New UPDATE Source Program Library And Binary Dataset

CDC 7600

```
myjob,T5.                              myname. X myext
ACCOUNT.myuserid.myjonumber.
REQUEST,NEWPL,SN=system,*PF.
*REQUEST,LGO,*PF.
UPDATE,F,N.
FTN,R=3,OPT=2,I=COMPILE.
CATALOG,LGO,myprogb,ID=myuserid.
CATALOG,NEWPL,myprogs,ID=myuserid.
RETURN,NEWPL.
RETURN,COMPILE.
PURGE,olds,myprogs,ID=myuserid,LC=1,HR=0.


PURGE,oldb,myprogb,ID=myuserid,LC=1,HR=0.


LGO.
7/8/9
*COMDECK ABCDEF
      COMMON / ABCDEF / ABC(100), ...
*DECK myprog
      ...
7/8/9
      (data)
6/7/8/9
```

CRAY-1

```
myjob.MICR.                            (Unless VAX front-end)
USER,myuserid,mypassword.              ( "    "    "    " )
JOB,JN=myjob,T=5,US=myuserid,PW=mypassword.  CS=10,NA=myname,X=myext
ACCOUNT,AC=myjonumber.


* UPDATE,P=0,N=SNPL.
* CFT,ON=ADIX,I=SCPL.
* SAVE,DN=SELD,PDN=myprogb,ID=myuserid.
* SAVE,DN=SNPL,PDN=myprogs,ID=myuserid.
* RELEASE,DN=SNPL.
* RELEASE,DN=SCPL.
* ACCESS,DN=olds,PDN=myprogs,ID=myuserid,UQ,NA,ED=3.
* DELETE,DN=olds.
* RELEASE,DN=olds.
* ACCESS,DN=oldb,PDN=myprogb,ID=myuserid,UQ,NA,ED=3.
* DELETE,DN=oldb.
* RELEASE,DN=oldb.
  LDR,MAP=PART,SET=INDEF.
  7/8/9
  *COMDECK ABCDEF
        COMMON / ABCDEF / ABC(100), ...
  *DECK myprog
        ...
  7/8/9
        (data)
  6/7/8/9
```

(Points to note appear on the next page.)

NOTES:

* For UPDATE, P=0 indicates a creation run.

* For CFT, default is I=SIN, so I=SCPL is essential here.

* Don't forget the DN= when using RELEASE. (Another alternative: RELEASE,DN=SNPL:SCPL.)

* Note the suggestion not to SAVE the source and binary datasets until both UPDATE and CFT have executed properly. Unless the source code is huge, this is recommended for such reasons as keeping edition numbers on source and binary the same, and reducing problems with which editions to delete in the later portion of the JCL.

* Note the suggestion to delete previous editions of the source and binary datasets. Arranging for this to be two editions earlier than the datasets just saved is a recommended practice, with some commenting-out of these lines on early creation runs until you're up to at least ED=3...

5.6 Using An Existing UPDATE Source Program Library (Running With Updates)

CDC 7600

```
myjob,T5.                          myname, X myext
ACCOUNT,myuserid,myjonumber.
ATTACH,OLDPL,myprogs,ID=myuserid.
UPDATE.
RETURN,OLDPL.
FTN,R=3,OPT=2,I=COMPILE.
RETURN,COMPILE.
ATTACH,myprog,myprogb,ID=myuserid.
LOAD,LGO,myprog.
EXECUTE.
7/8/9
*IDENT test

<UPDATE directives>

7/8/9
```

CRAY-1

```
myjob,MICR.                        (Unless VAX front-end)
USER,myuserid,mypassword.          (  "     "    "   "  )
JOB,JN=myjob,T=5,US=myuserid,PW=mypassword.  CS=10,NA=myname,X=myext
ACCOUNT,AC=myjonumber.
* ACCESS,DN=SPL,PDN=myprogs,ID=myuserid.
* UPDATE,P=SPL,IN.
* RELEASE,DN=SPL.
* CFT,ON=ADIX,I=SCPL.
* RELEASE,DN=SCPL.
* ACCESS,DN=myprog,PDN=myprogb,ID=myuserid.
* LDR,DN=SBLD:myprog,MAP=PART,SET=INDEF.

7/8/9
*IDENT test

<UPDATE directives>

7/8/9
```

NOTES:

* UPDATE does not echo your changes by default. Hence the IN parameter.
* Note the use of the colon (:) to load the new and old binary datasets in the right order.

5.7 Retrieving An UPDATE Source Dataset From The CRAY-1

CDC 7600

```
myjob,T5.                                  myname, X myext
ACCOUNT,myuserid,myjonumber.
ATTACH,OLDPL,myprogs,ID=myuserid.
UPDATE(F,S,C=0).
DISPOSE,SOURCE,....
7/8/9
*ID test
<possible update directives>
6/7/8/9
```

CRAY-1

```
myjob,MICR.                                (Unless VAX front-end)
USER,myuserid,mypassword.                  (  "    "    "    "  )
JOB,JN=myjob,T=5,US=myuserid,PW=mypassword.   CS=10,NA=myname,X=myext
ACCOUNT,AC=myjonumber.
* ACCESS,DN=SPL,PDN=myprogs,ID=myuserid.
* UPDATE,P=SPL,IN,F,S=source,C=0.
* DISPOSE,DN=source.
7/8/9
*ID test
<possible update directives>
6/7/8/9
```


NOTES:

* Use of the default, SSR, for the source file local dataset name can cause trouble - hence use of S=source.

  WARNING: Don't use less than 5 characters for this local dataset name, because of potential problems with the name of the file needed on the target machine.

* The DISPOSE sends, by default, the indicated DN to the place that originated the job, with some convention for the name of the resulting file on the originating PDP-11 or VAX. See also next example. Banner pages may or may not have to be edited out.

5.8 Retrieving A Plottable Dataset Stored On The CRAY-1

CDC 7600

```
myjob,T5.                          myname, X myext
ACCOUNT,myuserid,myjonumber.
ATTACH(myplot,myplotfile,ID=myuserid)
COPY(myplot,OUTPUT)

6/7/8/9
```

CRAY-1

```
myjob,MICR.                              (Unless VAX front-end)
USER,myuserid,mypassword.                (  "     "    "   "  )
JOB,JN=myjob,T=5,US=myuserid,PW=mypassword.  CS=10,MA=myname,X=myext
ACCOUNT,AC=myjonumber.
ACCESS(DN=myplot,PDN=myplotfile,ID=myuserid)
* COPYD(I=myplot,O=mycopy)
* DISPOSE(DN=mycopy,SDN=copyplt,DEFER)
  6/7/8/9
```

NOTES:

* Three local dataset names are involved here, including one that defines the file name on the target machine.

  WARNING: Don't use less than 5 characters for the SDN=copyplt name.

5.9 Echoing Input Data

CDC 7600

```
myjob,T5.                          myname, X myext
ACCOUNT,myuserid,myjonumber.
FTN,R=3,OPT=2.
COPYSP,INPUT,OUTPUT.
REWIND,INPUT.
SKIPF,INPUT,1.
LGO.
7/8/9
<FORTRAN source code>
7/8/9
<Data to be echoed>
6/7/8/9
```

CRAY-1

```
myjob,MICR.                                (Unless VAX front-end)
USER,myuserid,mypassword.                  ( "     "     "    " )
JOB,JN=myjob,T=5,US=myuserid,PW=mypassword.  CS=10,NA=myname,X=myext
ACCOUNT,AC=myjonumber.
CFT,ON=ADIX.
* COPYF,I=SIN,O=SOUT,NF=1.                 (or  just COPYF.)

* SKIPF,DN=SIN,NF=-1.                      (or  just SKIPF,NF=-1.)
LDR,MAP=PART,SET=INDEF.
7/8/9
<FORTRAN source code>
7/8/9
<Data to be echoed>
6/7/8/9
```

NOTES:

* All parameters shown for the COPYF may be defaulted for the example given. (I.e., NF=1 is the default number of files copied, etc.) (A file here refers to part of a dataset -- equivalent to a partition of a file in SCOPE terminology.)

* Note that SKIPF (Skip File) can be used to skip backwards safely, which is not always true under SCOPE - hence the REWIND there rather than use of SKIPB.

* Post Script: Loss of column 1 has been observed as a problem. The following JCL solves the problem in typical cases:

```
CFT,...
COPYF,I=SIN,O=data.
REWIND,DN=data.
COPYS,I=data,O=SOUT.
REWIND,DN=data.
ASSIGN,DN=data,A=FT05.
LDR,...
```

where A=FT05 indicates the "alias" necessary for the copy to be treated as the normal input stream...

5.10 Accessing DISSPLA And IMSL Libraries

A job using both of these popular libraries would contain something equivalent to the following:

```
ASSIGN ( DN=myplot, A=FTnn )
ACCESS ( DN=DISSPLA, PDN=TDISSPLA, ID=AMESLIB )
ACCESS ( DN=IMSL, PDN=IMSL, ID=AMESLIB )
LDR ( DN=..., LIB=IMSL:DISSPLA, SET=... )
DISPOSE ( DN=myplot, SDN=plotnam, DEFER )
```

where it is assumed that the DISSPLA plot file is to be post-processed on a PDP-11 or VAX with a Versatec printer/plotter. Note that PDN=TDISSPLA, not DISSPLA. This also requires

```
CALL VERSA ( nn )
```

in the FORTRAN program for the logical unit nn chosen to identify the plot file.

(See the ACF User's Guide for using DISSPLA in connection with the COM (microfilm) facility.)

See the ACF User's Guide also for information on other libraries such as NAG, and interesting packages such as the BLAS routines, LINPACK, and EISPACK, all contained in SSCILIB (searched automatically).

5.11 Changing CRAY Job Passwords

The ACF requests that passwords be changed from time to time for security reasons. To change your password, include the following JCL in a CRAY job, using at least 4 characters in your new password:

```
AMESLIB.
PASSWOR,PW=oldpassword,NPW=newpassword.
```


5.12 Debug Dumps

Tracing fatal execution errors can be assisted by rerunning your job with the Z option turned on for CFT (best to recompile whole program rather than just some routines), and invoking the DUMPJOB and DEBUG utilities:

```
CFT,ON=ADIXZ.
LDR,...
EXIT.
DUMPJOB.
DEBUG.          or, say,    DEBUG,SYMS=A:B:C:D,MAXDIM=100.
...
```

where the example from the ACF User's Guide indicates displaying of arrays A, B, C, and D up to a maximum first dimension of 100. (The default appears to be 20, for all local variables/arrays, but COMMON variables are not displayed.)

6.0 PROGRAM TRANSFER VIA MAGNETIC TAPE

The procedures in the next two sections were valid at the original time of writing. There are at least two other alternatives for transferring files. Firstly, those with access to both machines via a PDP-11 or VAX should retrieve their programs from the CDC 7600 on to the PDP-11 or VAX, edit the files suitably, then submit them in jobs to the CRAY-1. Secondly, transfer of files by three new magnetic tape utilities running on stations using CDC's NOS operating system (supported by both CDC and CRAY) is described in the memo of July 1, 1983 to CDC 7600 users, with further information as of July 20.

6.1 CDC 7600 Procedures And JCL

Step 1 - From the CDC 7600 tape library request a 9-track 1600 bpi unlabeled transmittal magnetic tape.

Step 2 - Execute the program that follows.

Step 3 - The procedure recommended at the time of writing is to carry the tape yourself from the CCF I/O room to the ACF I/O room.

```
myjob,T10.                                    myname, X myext
ACCOUNT,myuserid,T....Y.
COMMENT.                          JCL to copy an UPDATE Source file to
COMMENT.                          Magnetic Tape for Transfer to the CRAY-1:
MOUNT,SN=mydisk,VSN=.....
ATTACH,OLDPL,myprogs,SN=mydisk,ID=myuserid.
UPDATE,F,S,C=0.
FILE,mytape,RT=Z,BT=C,FL=80.
STAGE,mytape,POST,PE,NT,VSN=.....
COPY,SOURCE,mytape.
REWIND,SOURCE.
COPYSP,SOURCE,OUTPUT.
7/8/9
*ID xyz

<UPDATE directives, if any>

6/7/8/9
```

6.2 CRAY-1 Procedures And JCL

Upon checking in the 9-track 1600 bpi unlabeled tape from the CDC 7600 containing the UPDATE source program, execute the following on the CRAY-1:

```
myjob,MICR.                          (Unless VAX front-end)
USER,myuserid,mypassword.            ( "     "     "    " )
JOB,JN=myjob,T=10,US=myuserid,PW=mypassword.   CS=10,NA=myname.A=myext
ACCOUNT,AC=myjonumber.
*                                    JCL to read source code from tape written on CDC 7600
ACQUIRE,DN=mytape,TEXT=_^
   'USER,myuserid,mypassword.'_^
   'CHARGE,myjonumber'_^                                    <returns to front end
   'LABEL,mytape,PO=R,D=1600,F=SI,LB=KU,VSN=.....'_^            to mount tape>
   'DAYFILE.CTASK.'.
UPDATE,P=0,N=SNPL,I=mytape.
<SAVE,DN=SNPL,PDN=...,ID=...>
<AUDIT,...>
6/7/8/9
```


NOTES:

* The indented section represents NOS JCL for a job to run on the CYBER front-end. Each line should end with a continuation character. The TEXT= line should also have this continuation character.

7.0 UPDATE DIFFERENCES TO BEWARE OF

Differences between the two UPDATE implementations are well covered in the User's Guide for the ACF, so the following are included just for completeness. Basically, there are only minor differences, such as *PURGEDK instead of *PURDECK, *MOVEDK instead of *MOVE. *SEQUENCE is not available (nearest is *EDIT), nor is *ADDFILE (*INSERT instead?). (*SEQ and *NOSEQ turn on and off the sequence numbers written to SCPL.) On the UPDATE statement itself, the different use of L= is most notable. (L=ldn is now used to indicate the listing dataset name; other parameters have been introduced to control what goes to this dataset, such as IN for the equivalent of L=4.)

Post Script:

Adding and replacing decks has caused people problems on the CRAY. The following might help:

Adding:

```
*ID xyz
*DECK newdk
*I precedingdk.999    <last line of preceding deck>
.............         <text of new deck>
```

Replacing:

```
*ID xyz
*PURGEDK dkname
*DECK dkname
.............    <text of revised deck>
```

3.0 FORTRAN CHANGES THAT MAY BE NECESSARY

Even with the advent of FORTRAN 77 on CDC, CRAY, and DEC machines, certain machine dependencies are inevitable. The following covers some of the more likely points.

* The PROGRAM statement necessary on the CDC 7600 may be either left intact, shortened by removal of the (INPUT,...) portion, or omitted altogether (though use of certain compiler options do depend on its presence).

* Any LEVEL n, ... statements should be removed.

* There are now 8 characters per 64-bit word, rather than 10 per 60-bit word. This means switching from A10 to A8 format, and adjusting code dealing with alphanumeric strings accordingly. (Use of CHARACTER variables and just A format is urged.)

* ENCODE and DECODE statements are otherwise unaffected. (Again, use of CHARACTER variables and internal READs and WRITEs is recommended in place of ENCODE/DECODE.)

* Character strings should be delimited by APOSTROPHES under FORTRAN 77, though CFT supports QUOTATION marks and ASTERISKS too (not recommended).

* Calls to the DATE and JDATE system routines will need attention.

* Other system routines such as TIME have different names, but the one for cpu seconds used so far (SECOND) is unchanged.

* NAMELISTs should not be troublesome. Both $ and & are valid delimiters.

* ENTRY statements in SUBROUTINEs may be left unchanged, although they are now permitted to have argument lists differing from that of the relevant SUBROUTINE. However, any ENTRYs in FUNCTION subprograms MUST have an argument list, even if it is null. E.g.: ENTRY IBUG () but not ENTRY IBUG.

* BUFFER IN and BUFFER OUT are supported as CFT extensions. So is the UNIT function for checking the status of these operations, though possible returned values are -2., -1., 0., 1., and 2 rather than -1., 0., and 1.

* READMS/WRITMS (in fact any direct-access I/O) will not be available until the release of CFT Version 1.11, expected with the release of COS 1.12.

* The EOF and IOCHEC functions need to be replaced by the END=... and ERR=... constructs.

<continued>

* Certain bounds have been raised, permitting (if not necessitating) FORTRAN changes as follows:

  > The maximum number of subscripts is no longer 3 - it is now 7;

  > The maximum array or COMMON block length is no longer 131K words (2**17) - it is theoretically 2**22 now (4194K);

  > The maximum integer magnitude is increased from 2**46 - 1 to 2**63 (i.e., 2.8 x 10**14 to 10**19, approximately);

  > The max/min real magnitudes are expanded from 10**-293 --> 10**+322 to 10**(+/-2466) = 2**(+/-8191) approximately.

* Users can expect their programs to run at least 2.5 times more quickly on the CRAY-1 than on the CDC 7600. This is the scalar rate of improvement. Some programs will see considerably greater factors without change (since CFT automatically vectorizes eligible inner DO loops). But vectorization is another story, not covered here, except to suggest the use of the FLODUMP utility for identifying which routines would make most sense to spend time on for improving vectorization:

```
...

CFT,ON=F.
LDR.
EXIT.
DUMPJOB.
FLODUMP.
...
```

where the main program MUST have a PROGRAM statement. See the ACF User's Guide, and the FORTRAN and COS Reference Manuals for much more.