1904S Emulator

Bare Metal Installation

Author: B W Spoor

Date: June 2017

Issue: 1 (Initial)

Contents

1. Introduction

This document is designed to assist you in the installation of a system running on the ICL 1904S emulator (or indeed a real 1904S or similar hardware) in a ‘bare metal’ situation, where you only have the hardware, blank disc cartridges and the various Executive and Library tapes available.

You will notice that we are using a date 28 years previous (currently 1989) as the year, picked to keep the day of the week to be in line with the current calendar. We are doing this to avoid any Y2K issues, also E6RM and EWG3, unmodified, will not accept a date in the current century.

The functions of both the Site Engineer and the Systems Programmer are covered, both of whom are needed at to complete this task. We will ignore the System Manager and assume that his requirements for the installation have been specified.

2. Executive

The Site Engineer is responsible for the installation and testing of the hardware which will be handed over to the customer after the successful completion of a set of acceptance tests.

Your (friendly) Site Engineer will be able to initialise (#XPJ1/#XPJ2) your initial system disc pack for you, ready for you to install any basic system software required. Once your initial system is up and running, you will be able to initialise any further disc packs required, install an operating system (if required) and any further system software and application software.

Another job for the Site Engineer is to install your Executive(s) onto the relevant discs, so that you can boot from disc; also initialise, where relevant, ICLKHARDFILE with the DCPs for the 2812 disc controllers. Neither of these jobs are regarded as 'user' functions.

2.1 Setup E6RM on EDS8

Using the method described, this will require the system to be booted 3 times, the first two from magnetic tape, the final time from the newly created system disc.

Load the E6RM tape on transport 0 of the first MT cluster (Operator's Unit 30), boot from SI Channel 26 and run with overlays in core. #XPJ1/#XPJ2/#XPJC and other utilities can now be loaded from the appropriate library tape and run using parameters on either cards, paper tape or from the console (#XPJ1/#XPJ2 only). Only a single disc cartridge is required for a basic E6RM system, additional ones can be initialised at this time for later use if desired. A typical set of card parameters for the run can be found in Appendix 2.

Once E6RM has loaded you will need to load a blank cartridge on the transport 0 of the first EDS8 cluster (Operator's Unit 40) and any other cartridges you wish to initialise now, as well as the General Purpose Library Tape. Leave the Executive tape loaded, it will be required again.

```
LIB. NO. / HOW MANY K? 001A/128K


L001A/EXEC 601 01A


EXEC   M/C  601 ISS 01A PE 000V GRT7/08 05/05/82
NAME 1904S EMULATOR INITIAL RELEASE (SYS601)
ANY MODIFICATIONS? N


E6RM MARK 1   VERSION 5   (GEN 54)


DATE PLEASE?   FRI05MAY89

TIME PLEASE?   2202

OVERLAY DEVICE NO.?CORE
```

```
USING DIRECT ACCESS DEVICES ?Y

USING DISK FIND ?N

 CLSD U30 EXEC PROGS  /0/1 :- *3600040 RET 365
 U40 NON UDAS
 CLSD U31 PROGRAM GPLT/0/1 :- *4650000 RET <658
  SD 40 *301000 203 80
 OK
22/03/00
  FI#XPJ1#GPLT
 OK
 0#XPJ1; DLTD:-LO #XPJ1 31 OK
 S#XPJ1
 0#XPJ1; HALT:-
  GO#XPJ121
 OK
 CLSD U31 PROGRAM GPLT/0/1 :- *4650000 RET <658
 0#XPJ1; CORE   6336
 0#XPJ1; U 9 AS  0
 0#XPJ1; U14 AS  0
22/06/00


NO RESERVE OVERLAY DEVICE


 0#XPJ1; U 9FREE
 0#XPJ1; HALT:-HH
 U40 SU *301000
  DE#XPJ1
 OK
 0#XPJ1; CLKD     4
 0#XPJ1; DLTD:-
  FI#XPJ2#GPLT
 OK
 0#XPJ2; DLTD:-LO #XPJ2 31 OK
 S#XPJ2
 0#XPJ2; HALT:-
  GO#XPJ221
 OK
 CLSD U31 PROGRAM GPLT/0/1 :- *4650000 RET <658
 0#XPJ2; CORE   7808
 0#XPJ2; U 9 AS  0
 0#XPJ2; U14 AS  0
 0#XPJ2; LOAD SU *301000
22/09/00
 U40 SU *301000
 U40 SU *301000
 0#XPJ2; U 9FREE
 0#XPJ2; HALT:-HH
  DE#XPJ2
 OK
 0#XPJ2; DLTD:-
  FI#XPJC#GPLT
 OK
 0#XPJC; DLTD:-LO #XPJC 31 OK
 P#XPJC
 0#XPJC; HALT:-
```

```
  GO#XPJC21
 OK
 CLSD U31 PROGRAM GPLT/0/1 :- *4650000 RET <658
 0#XPJC; U14 AS  0
 0#XPJC; U 9 AS  0
 0#XPJC; U 9FREE
 0#XPJC; U14FREE
 0#XPJC; HALT:-HH
22/10/00
  DE#XPJC
 OK
 0#XPJC; DLTD:-


NO RESERVE OVERLAY DEVICE


  FI#XPEU#GPLT
 OK
 0#XPEU; DLTD:-LO #XPEU 31 OK
 Q#XPEU
 0#XPEU; HALT:-
  GO#XPEU21
 OK
 CLSD U31 PROGRAM GPLT/0/1 :- *4650000 RET <658
 0#XPEU; U 9 AS  0
 0#XPEU; U14 AS  0
 0#XPEU; U31 AS 0 OM*00100
 0#XPEU; DISP:- RUN  OK
 0#XPEU; U14FREE
 0#XPEU; DLTD:- OK
  FI#XPEU#GPLT
 OK
 CLSD U31 PROGRAM GPLT/0/1 :- *4650000 RET <658
22/11/00
 0#XPEU; DLTD:-LO #XPEU 31 OK
 Q#XPEU
 0#XPEU; HALT:-
  GO#XPEU21
 OK
 CLSD U31 PROGRAM GPLT/0/1 :- *4650000 RET <658
 0#XPEU; U 9 AS  0
 0#XPEU; U14 AS  0
 0#XPEU; U31 AS 0 OM*00100
22/12/00
 0#XPEU; DISP:- RUN  OK
 0#XPEU; U14FREE
 0#XPEU; CLKD     1
 0#XPEU; DLTD:- OK
22/25/00
 CLSD U31 PROGRAM GPLT/0/1 :- *4650000 RET <658
22/26/00
```

This run has initialised the system cartridge, created the basic system files and installed basic utilities. The machine should be closed and re-booted from magnetic tape in order to transfer the E6RM Executive and overlays to the disc. Note that #ENGD will be loaded from the card reader.

```
LIB. NO. / HOW MANY K? 001A/128K


L001A/EXEC 601 01A


EXEC    M/C  601 ISS 01A PE 000V GRT7/08 05/05/82
 NAME 1904S EMULATOR INITIAL RELEASE (SYS601)
ANY MODIFICATIONS? N


E6RM MARK 1   VERSION 5    (GEN 54)


DATE PLEASE?   FRI05MAY89

TIME PLEASE?    2229

OVERLAY DEVICE NO.?40

TRANSFER OVERLAYS?Y

MAG. TAPE NO.?30

LIB. NO.?001A

 U40 SU *301000
22/30/02
CLSD U30 EXEC PROGS  /0/1 :- *3600040 RET 365
  LO#ENGD9
 OK
 S#ENGD0723705T
 0#ENGD; HALT:-
 0#ENGD; U9 FREE
  GO#ENGD21
 OK
 0#ENGD; DISP:-ENGD0723705T
 0#ENGD; U 9 AS  0
 0#ENGD; U 9FREE
 0#ENGD; DISP:-O0
 0#ENGD; U30 AS 0 OM*00200
 0#ENGD; DISP:-C0
 0#ENGD; DISP:-O1
 0#ENGD; DISP:-C1
 0#ENGD; U14 AS  0
 CLSD U30 EXEC PROGS  /0/1 :- *3600040 RET 365
 U40 SU *301000
22/32/00
 0#ENGD; DISP:-O0
 0#ENGD; DISP:-O1
 0#ENGD; DISP:-C1
 0#ENGD; DISP:-C0
 0#ENGD; U14FREE
 0#ENGD; HALT:-END OF RUN
22/33/00
  DE#ENGD
 OK
```

```
0#ENGD; DLTD:-
```

This second run has transferred the Executive fixed core and overlays to the new system cartridge and installed the necessary bootstrap block on the cartridge. The machine should be closed, tapes unloaded and re-booted from disc (SI Channel 24). This is now a working system.

```
LIB. NO. / HOW MANY K? 001A/128K

L001A/EXEC 601 01A

EXEC    M/C  601 ISS 01A PE 000V GRT7/08 05/05/82
 NAME 1904S EMULATOR INITIAL RELEASE (SYS601)
ANY MODIFICATIONS? N

E6RM MARK 1   VERSION 5    (GEN 54)

DATE PLEASE?   FRI05MAY89

TIME PLEASE?    2234

OVERLAY DEVICE NO.?40

TRANSFER OVERLAYS?N

 U40 SU *311000
22/35/02
```

In the future the system can be loaded from this system cartridge, the tape is no longer needed on a daily basis. The same is true of the library tape, the most common utilities are available in PROGRAM UTIL for faster, more convenient loading. Any other discs required, e.g. for operating systems, applications, compilers, etc., can now be set up.

2.2 Setup E6RM on EDS30/60

This is a very similar process to that described for EDS8, the differences being:-

1) The 2812 DCP needs to be loaded from paper tape as a first step
2) The parameters for #XPJ1/#XPJ2/#XPJC will be different (see Appendix 3)
3) On the second run #ENGF is used instead of #ENGD, but after loading and running #DCPD (FI#DCPD#DCPS from magnetic tape) to initialise ICLKHARDFILE and install the 2812 DCPs.

Once this process has been completed, the 2812 DCP can be loaded (must be) from disc before booting the Executive.

It is possible to set EDS30/60 cartridges up to dual boot E6RM and EWG3, a feature not available with EDS8s.

2.3 Setup EWG3 on EDS8 or EDS30/60

A working basic E6RM system is required (see 2.1 or 2.2 above) as EWG3/GEORGE 3 does not have the necessary facilities available.

On an EDS8 system, EWG3 cannot share a disc with E6RM; however for EDS30/60 it is possible for both Executives to be on one disc with a dual boot facility, or a dedicated disc as for EDS8.

Additional discs need to be initialised and files allocated using #XPJ1, #XPJ2 and #XPJC, example parameters for this can be found in Appendix 4, Appendix 5 or Appendix 6 as appropriate.

At the same time, the other discs needed for a GEORGE 3 system can be initialised (at this point can only be done under E6RM) and the files allocated (easier under E6RM whilst already running #XPJC). NB the GEORGE 3 loader/allocator is broken in 8.67.

2.3.1 EWG3 on EDS8

After setting up the disc, EWG3 needs to be loaded to disc from the Executive tape using #EXD1. Load the cartridge to be used on Operators Unit No. 40.

2.3.2 EWG3 on Dedicated EDS30/60

After setting up the disc, ICLKHARDFILE needs setting up with the 2812 DCPs and then EWG3 loaded to disc from the Executive tape using #EXD1. Load the cartridge to be used on Operators Unit No. 40.

2.3.3 EWG3 on Shared EDS30/60

After setting up the disc, E6RM needs to be re-installed with the dual boot capability and then EWG3 loaded to disc from the Executive tape using #EXD1. The cartridge will already be loaded on Operators Unit No. 30.

It should be noted that when setting up a dual boot E6RM/EWG3 system disc that the bootstrap used is for EWG3. To load E6RM, the filename question should be answered with ICLKE6RMBOOT, which will then initiate the e6RM bootstrap process.

2.3.4 EWG3 Bootstrap Installation (Common)

The bootstrap can only be written by EWG3, so it is necessary to bootstrap the system from magnetic tape. Load the EWG3 tape on transport 0 of the first MT cluster (Operator's Unit 30), boot from SI Channel 26.

```
TYPE IN PROGRAM NAME   EXEC 601 01A


 GEORGE 3 W.G. EXEC GEN 41.
MENDS  1 - 0788 INCLUDED


1904S EMULATOR INITIAL RELEASE (SYS601)
CHANGE CORE SIZE ?N


DATE PLEASE?     FRI05MAY89


TIME PLEASE?     2302


READY

DA ICLKGEXDUMP
NOT FOUND

DA ICLKNIBOOTEX
NOT FOUND
  BS *301010         (OR *301000,  *311000 AS APPROPRIATE)

BOOTSTRAP WRITTEN TO   301000
```

At this point EWG3 has been set up and can be rebooted from disc (SI Channel 24=Operators Unit No. 30, SI Channel 25=Operators Unit No. 40), or just continue to install the GEORGE 3 operating system.

```
  FI 21 G3MK8D
 OK

FOUND  ON  TSN  4672012

 LOADED


  TW
   OK
FS 3,6,G3FSRES3
   OK
FS 4,6,G3FSRES4
   OK
FS 5,6,G3FSRES5
   OK
FS 6,6,G3FSRES6
   OK
 FS 10,6,G3SWAPRES,,SWAP       (ON EDS 8)
   OK
 FS 10,6,G3SWAPRES10,,SWAP     (ON EDS 30/60)
   OK
 FS 11,6,G3SWAPRES11,,SWAP     (ON EDS 30/60)
   OK
DA GEORGE3C
   OK
LOADED
```

```
23.05.13
SCF SYSTEM NOT RUNNING
RESTORE?
  1

PLEASE ENGAGE INCREMENT 1
DUMP TAPE 15155 ENGAGED
23.06.00
JOURNAL DATA MAY HAVE BEEN LOST
GENERATION 1 OF SYSTEM JOURNAL STARTED
:SYSTEM.OUTPUT RECREATED-ALL LISTFILES LOST
ALL INSTALLATION PARAMETERS SET TO DEFAULT:PLEASE RUN IPEDIT AS REQUIRED
UNIT U29 HAS BECOME INOPERABLE



GEORGE MEAN TIME: 1005071
GEORGE 3 MK8.67 READY .
 ERROR  IN EMSJOBS : THIS IS NOT A COMMAND
ERROR IN PARAMETER 3  IN RJ : ENTRANT JUGGERNAUT DOES NOT EXIST
FINISHED :SYSTEM,BACKMAP : 1 LISTFILES
  FN NOW

23.06.46 SYSTEM CLOSING DOWN NOW
ALL USER JOBS NOW FINISHED
:DUMPER,A0 QUESTION: DO YOU REQUIRE TO DUMP?
  AN :DUMPER,A0,N

FINISHED :DUMPER,A0 : 1 LISTFILES
23.07.00    5MAY82
EVERYTHING HAS FINISHED

EXEC MESSAGE:- DELE
 OK
```

A basic GEORGE 3 system has now been loaded to disc, the machine can now be rebooted using the discs (SI Channel 24=Operators Unit No. 30, SI Channel 25=Operators Unit No. 40) after unloading any tapes used above and the other software need to create a ‘full’ working system can be installed.

```
B/S LOADED
EXEC FILE ?   G3EXEC


 GEORGE 3 W.G. EXEC GEN 41.
MENDS  1 - 0788 INCLUDED


1904S EMULATOR INITIAL RELEASE (SYS601)
CHANGE CORE SIZE ?N


DATE PLEASE?     FRI05MAY89
```

```
TIME PLEASE?     2308


READY

DA ICLKGEXDUMP
NOT FOUND

DA ICLKNIBOOTEX
NOT FOUND
  DA GEORGE3C
 OK
LOADED

23.08.08
SCF SYSTEM NOT RUNNING
RESTORE?
  N

JOURNAL DATA MAY HAVE BEEN LOST
GENERATION 2 OF SYSTEM JOURNAL STARTED
UNIT U29 HAS BECOME INOPERABLE



GEORGE MEAN TIME: 1005076
GEORGE 3 MK8.67 READY .
ERROR  IN EMSJOBS : THIS IS NOT A COMMAND
FINISHED :SYSTEM,BACKMAP : 1 LISTFILES
23.09.00    5MAY82
  FN NOW

23.09.06 SYSTEM CLOSING DOWN NOW
ALL USER JOBS NOW FINISHED
:DUMPER,A1 QUESTION: DO YOU REQUIRE TO DUMP?
  AN :DUMPER,A1,N

FINISHED :DUMPER,A1 : 1 LISTFILES
EVERYTHING HAS FINISHED

EXEC MESSAGE:- DELE
 OK
```

3. Installing Other Operating Systems

Once a bootable Executive disc has been created, various operating systems may be set up as required. Full details for these may be found in the relevant manuals, but basics are covered below.

3.1 GEORGE 3

The details given with the installing of EWG3 will give a very basic system, ready to setup and customise as required. Restoring from increment 2 will load a partly built system, while increment 3 contains a basic working system.

Full information on installing and running GEORGE 3 may be found in the following manuals:-

TP4345	Operating Systems GEORGE 3 and 4
TP4438	GEORGE 3 and 4 Operations Management
TP4441	GEORGE 3 and 4 Restore Time Macros

3.2 GEORGE 2

Appendix 7 contains details of a sample disc initialisation and parameters for setting up GEORGE 2 on an EDS 8. Full information on installing and running may be found in the following manual:-

TP4432	GEORGE 2 Disc Based Operating System

3.3. MAXIMOP

Appendix 8 contains details of a sample disc initialisation and parameters for setting up MAXIMOP on an EDS 8. Full information on installing and running may be found in the following manuals:-

TP4445	Using MAXIMOP
TP4448	MAXIMOP System

3.4 GEORGE 2+DOF/1T

Appendix 9 contains details of a sample disc setup for GEORGE 2+/DOF/1T on EDS 30/60. Full information on installing and running system this may be found in the following Dataskil manuals:-

GEORGE 2+ Software Issue System
GEORGE 2+ Operations Management Manual
GEORGE 2+ User Reference Manual
GEORGE 2+ Dynamic Output Files User Guide

Appendix 1 – System Files Required

E6RM/EDS8:-

```
ICLKEXECFILE – File from which E6RM is loaded
XPEBOOTSTRAP – File containing Search/Loader for programs
PROGRAM UTIL – Library containing general utilities (UTIL, DISC, FRED)
```

E6RM/EDS30/60:-

```
ICLKHARDFILE – File holding basic Executive bootstrap and 2812 DCPs
ICLKE6RMBOOT – File holding E6RM bootstrap (dual boot systems only)
ICLKEXECFILE – File from which E6RM is loaded
XPEBOOTSTRAP – File containing Search/Loader for programs
PROGRAM UTIL – Library containing general utilities (UTIL, DISC, FRED)
```

EWG3/EDS8:-

```
G3EXEC       - Executive File (example name)
GEORGE3A     - Chapter File 'A' System (example name)
GEORGE3B     - Chapter File 'B' System (example name)
GEORGE3C     - Chapter File 'C' System (example name)
```

EWG3/EDS30/60:-

```
ICLKHARDFILE – File holding basic Executive bootstrap and 2812 DCPs
G3EXEC       - Executive File (example name)
GEORGE3A     - Chapter File 'A' System (example name)
GEORGE3B     - Chapter File 'B' System (example name)
GEORGE3C     - Chapter File 'C' System (example name)
```

E6RM+EWG3/EDS30/60:-

```
ICLKHARDFILE – File holding basic Executive bootstrap and 2812 DCPs
ICLKE6RMBOOT – File holding E6RM bootstrap (dual boot systems only)
ICLKEXECFILE – File from which E6RM is loaded
XPEBOOTSTRAP – File containing Search/Loader for programs
PROGRAM UTIL – Library containing general utilities (UTIL, DISC, FRED)
G3EXEC       - Executive File (example name)
GEORGE3A     - Chapter File 'A' System (example name)
GEORGE3B     - Chapter File 'B' System (example name)
GEORGE3C     - Chapter File 'C' System (example name)
```

Appendix 2 – Example Parameters for E6RM on EDS 8

XPJ1 (system disc):-

```
XPJ11,ED,301000,203,*
****
```

XPJ2 (system disc):-

```
XPJ21,1,301000,S,*
XPJ21,3,301000,1,0,*
****
```

XPJC (system disc):-

```
XPJC1,1,301000,XPEBOOTSTRAP,0,0,3,1,
XPJC4,S(4,4,0,79)*
XPJC1,1,301000,ICLKEXECFILE,0,0,2,1,M,
XPJC4,S(5,70,0,79)*
XPJC1,1,301000,PROGRAM UTIL,0,0,3,1,
XPJC4,S(71,92,0,79)*
****
```

XPEU (XPEBOOTSTRAP):-

```
CDF       XPEBOOTSTRAP,0,0,1
IMT       PROGRAM GPLT,1
IP        XPMO,3,MT
IP        XPEO,6D,MT
IP        XPXX,4,MT
PRT
F
```

XPEU (PROGRAM UTIL):-

```
CDF       PROGRAM UTIL,0,0,1
IMT       PROGRAM GPLT,1
IP        XPMV,4B,MT
IP        XPMW,4B,MT
IP        XQMC,3,MT
IP        XQME,5A,MT
IP        XQMH,4A,MT
IP        XQMV,5A,MT
IP        XQMX,4A,MT
IP        XQMY,4A,MT
IP        XRMA,4A,MT
IP        XRMB,3,MT
IP        XRMD,3A,MT
IP        XRME,1A,MT
IP        XRMF,6A,MT
IP        XRMG,2A,MT
IP        XRMJ,4A,MT
IP        XRML,3B,MT
IP        XRMM,1A,MT
IP        XRMN,1,MT
IP        XRPM,5B,MT
```

```
IP       XKYA,10A,MT
IP       XQCC,4,MT
IP       XRCP,2A,MT
IP       XRCQ,2C,MT
IP       XRCT,2A,MT
IP       XRCV,3A,MT
IP       XRLP,6,MT
IP       XRST,1,MT
IP       XRTB,1F,MT
IP       XRTC,5,MT
IP       XRTV,2,MT
IP       XSDM,3B,MT
IP       XSME,3,MT
IP       XSMG,7A,MT
IP       XSML,2B,MT
IP       XFYZ,1C,MT
IP       XPEC,7B,MT
IP       XPEK,5C,MT
IP       XPES,5F,MT
IP       XPEU,7B,MT
IP       XMED,2E,MT
IP       XRAA,3,MT
IP       XREB,3B,MT
IP       XREJ,3C,MT
IP       XRPE,4B,MT
IP       XPJ1,7B,MT
IP       XPJ2,6C,MT
IP       XPJ7,4,MT
IP       XPJC,1H,MT
IP       XPJK,1A,MT
IP       XPJL,1D,MT
IP       XPJQ,1C,MT
IP       XPJR,1B,MT
IP       XPJW,2C,MT
IP       XPJX,1B,MT
IP       XPJZ,3A,MT
IP       XSDC,3B,MT
IP       XSDP,1F,MT
IP       XJGA,3,MT
IP       XJRT,12D,MT
PRT
F
```

ENGD:-

```
O/0/5/70^
B/6^
M/EXEC PROGS  ^
E^
```

Appendix 3 – Example Parameters for E6RM on EDS 30/60

XPJ1 (system disc):-

```
XPJ11,MD,31100,406,* (for EDS60, adjust for EDS30)
****
```

XPJ2 (system disc):-

```
XPJ21,1,311000,S,*
XPJ21,3,311000,1,0,*
****
```

XPJC (system files):-

```
XPJC1,1,311000,XPEBOOTSTRAP,0,0,3,1,
XPJC4,S(1,1,120,295)*
XPJC1,1,311000,ICLKHARDFILE,1,0,2,4,M,
XPJC4,S(2,3,0,295)*
XPJC1,1,311000,ICLKE6RMBOOT,0,0,2,4,M,
XPJC4,S(4,4,0,39)*
XPJC1,1,311000,ICLK7903SPEC,1,0,3,1,
XPJC4,S(4,4,40,295)*
XPJC1,1,311000,ICLKEXECFILE,60,0,2,1,M,
XPJC4,S(5,10,0,295)*
XPJC1,1,311000,PROGRAM UTIL,0,0,3,1,
XPJC4,S(11,16,0,295)*
****
```

XPEU (XPEBOOTSTRAP):-

See appendix 2 above

XPEU (PROGRAM UTIL):-

See appendix 2 above

DCPD:-

```
DCPD0,0,0,75,26,0,32,*
DCPD6,311000,*
DCPD5,311000,*
****
```

ENGF:-

```
O/60^
B/6^
M/EXEC PROGS  ^
E^
```

Appendix 4 – Example Parameters for EWG3 on EDS 8

XPJ1 (system discs):-

```
XPJ11,ED,301010,203,*
XPJ11,ED,301011,203,*
XPJ11,ED,301012,203,*
XPJ11,ED,301003,203,*
XPJ11,ED,301014,203,*
XPJ11,ED,301015,203,*
****
```

XPJ2 (system discs):-

```
XPJ21,1,301010,0,*
XPJ21,3,301010,1,1,*
XPJ21,1,301011,0,*
XPJ21,3,301011,1,1,*
XPJ21,1,301012,0,*
XPJ21,3,301012,1,1,*
XPJ21,1,301003,0,*
XPJ21,3,301003,1,1,*
XPJ21,1,301014,0,*
XPJ21,3,301014,1,1,*
XPJ21,1,301015,0,*
XPJ21,3,301015,1,1,*
****
```

XPJC (system files):-

```
XPJC1,1,301010,G3EXEC,0,0,1,1,
XPJC4,S(1,6,0,79)*
XPJC1,1,301010,GEORGE3A,0,0,1,1,
XPJC4,S(7,61,0,79)*
XPJC1,1,301010,GEORGE3B,0,0,1,1,
XPJC4,S(62,116,0,79)*
XPJC1,1,301010,GEORGE3C,0,0,1,1,
XPJC4,S(117,171,0,79)*
XPJC1,1,301011,G3SWAPRES,0,0,1,1,
XPJC4,S(1,202,0,79)*
XPJC1,1,301012,G3FSRES3,0,0,1,1,
XPJC4,S(1,202,0,79)*
XPJC1,1,301003,G3FSRES4,0,0,1,1,
XPJC4,S(1,202,0,79)*
XPJC1,1,301014,G3FSRES5,0,0,1,1,
XPJC4,S(1,202,0,79)*
XPJC1,1,301015,G3FSRES6,0,0,1,1,
XPJC4,S(1,202,0,79)*
****
```

EXD1:-

```
TAPENIEXECMC 601,*
FILEEXEC 601 01A,*
DISCG3EXEC,301010,*
****
```

Appendix 5 – Example Parameters for EWG3 on EDS 30/60

XPJ1 (system discs):-

```
XPJ11,MD,311000,406,*      (for EDS60, adjust for EDS30)
XPJ11,MD,311001,406,*      (for EDS60, adjust for EDS30)
XPJ11,MD,311002,406,*      (for EDS60, adjust for EDS30)
****
```

XPJ2 (system discs):-

```
XPJ21,1,311010,0,*
XPJ21,3,311010,1,0,*
XPJ21,1,311011,0,*
XPJ21,3,311011,1,0,*
XPJ21,1,311012,0,*
XPJ21,3,311012,1,0,*
****
```

XPJC (system files):-

```
XPJC1,1,311010,G3EXEC,0,0,1,1,
XPJC4,S(1,2,0,295)*
XPJC1,1,311010,GEORGE3A,0,0,1,1,
XPJC4,S(3,17,0,295)*
XPJC1,1,311010,GEORGE3B,0,0,1,1,
XPJC4,S(18,32,0,295*
XPJC1,1,311010,GEORGE3C,0,0,1,1,
XPJC4,S(33,47,0,295)*
XPJC1,1,311011,G3FSRES3,0,0,1,1,
XPJC4,S(1,192,0,295*
XPJC1,1,311011,G3SWAPRES10,0,0,1,1,
XPJC4,S(193,213,0,295*
XPJC1,1,311011,G3FSRES4,0,0,1,1,
XPJC4,S(214,405,0,295*
XPJC1,1,311012,G3FSRES5,0,0,1,1,
XPJC4,S(1,192,0,295*
XPJC1,1,311012,G3SWAPRES11,0,0,1,1,
XPJC4,S(193,213,0,295*
XPJC1,1,311012,G3FSRES6,0,0,1,1,
XPJC4,S(214,405,0,295*
****
```

EXD1:-

```
TAPENIEXECMC 601,*
FILEEXEC 601 01A,*
DISCG3EXEC,311010,*
```

Appendix 6 – Example Parameters for E6RM+EWG3 on EDS 30/60

XPJ1 (extra system discs):-

```
XPJ11,MD,311001,406,*      (for EDS60, adjust for EDS30)
XPJ11,MD,311002,406,*      (for EDS60, adjust for EDS30)
****
```

XPJ2 (extra system discs):-

```
XPJ21,1,311011,0,*
XPJ21,3,311011,1,0,*
XPJ21,1,311012,0,*
XPJ21,3,311012,1,0,*
****
```

XPJC (additional system files):-

```
XPJC1,1,301010,G3EXEC,0,0,1,1,
XPJC4,S(17,18,0,295)*
XPJC1,1,301010,GEORGE3A,0,0,1,1,
XPJC4,S(19,33,0,295)*
XPJC1,1,301010,GEORGE3B,0,0,1,1,
XPJC4,S(34,48,0,295*
XPJC1,1,301010,GEORGE3C,0,0,1,1,
XPJC4,S(49,63,0,295)*
XPJC1,1,311011,G3FSRES3,0,0,1,1,
XPJC4,S(1,192,0,295*
XPJC1,1,311011,G3SWAPRES10,0,0,1,1,
XPJC4,S(193,213,0,295*
XPJC1,1,311011,G3FSRES4,0,0,1,1,
XPJC4,S(214,405,0,295*
XPJC1,1,311012,G3FSRES5,0,0,1,1,
XPJC4,S(1,192,0,295*
XPJC1,1,311012,G3SWAPRES11,0,0,1,1,
XPJC4,S(193,213,0,295*
XPJC1,1,311012,G3FSRES6,0,0,1,1,
XPJC4,S(214,405,0,295*
****
```

ENGF:-

```
O/60^
B/6^
G
M/EXEC PROGS  ^
E^
```

EXD1:-

```
TAPENIEXECMC 601,*
FILEEXEC 601 01A,*
DISCG3EXEC,301010,*
```

Appendix 7 – Example Parameters for GEORGE 2 on EDS 8

XPJ1:-

```
XPJ11,ED,301002,203,*
****
```

XPJ2:-

```
XPJ21,1,301002,S,*
XPJ21,3,301002,1,0,*
****
```

XPJC:-

```
XPJC1,1,301002,XPEBOOTSTRAP,0,0,3,1,
XPJC4,S(4,4,0,79)*
XPJC1,1,301002,ICLKGEOGMCRO,0,0,2,1,
XPJC4,S(5,5,0,79)*
XPJC1,1,301002,ICLKCODEDJDS,0,0,1,1,
XPJC4,S(6,6,0,79)*
XPJC1,1,301002,ICLKCODEDJDS,1,0,1,1,
XPJC4,S(7,7,0,79)*
XPJC1,1,301002,ICLKCODEDJDS,2,0,1,1,
XPJC4,S(8,8,0,79)*
XPJC1,1,301002,ICLKCODEDJDS,3,0,1,1,
XPJC4,S(9,9,0,79)*
XPJC1,1,301002,ICLKXKEEFILE,0,0,2,4,
XPJC4,S(10,12,0,79)*
XPJC1,1,301002,ICLKXKYEFILE,0,0,2,1,
XPJC4,S(13,13,0,39)*
XPJC1,1,301002,ICLKXKZEFILE,0,0,2,1,
XPJC4,S(13,13,40,79)*
XPJC1,1,301002,PROGRAM GEOR,0,0,3,1,
XPJC4,S(14,23,0,79)*
XPJC1,1,301002,ICLKGEOINPUT,1,0,1,1,
XPJC4,S(24,26,0,79)*
XPJC1,1,301002,ICLKGEOINPUT,2,0,1,1,
XPJC4,S(27,29,0,79)*
XPJC1,1,301002,ICLKGEOINPUT,3,0,1,1,
XPJC4,S(30,32,0,79)*
XPJC1,1,301002,ICLKGEOINPUT,4,0,1,1,
XPJC4,S(33,35,0,79)*
XPJC1,1,301002,ICLKGEOINPUT,5,0,1,1,
XPJC4,S(36,38,0,79)*
XPJC1,1,301002,ICLKGEOINPUT,6,0,1,1,
XPJC4,S(39,41,0,79)*
XPJC1,1,301002,ICLKGEOUTPUT,1,0,1,1,
XPJC4,S(42,51,0,79)*
XPJC1,1,301002,ICLKGEOUTPUT,2,0,1,1,
XPJC4,S(52,61,0,79)*
XPJC1,1,301002,ICLKGEOUTPUT,3,0,1,1,
XPJC4,S(62,71,0,79)*
XPJC1,1,301002,ICLKGEOUTPUT,4,0,1,1,
XPJC4,S(72,81,0,79)*
XPJC1,1,301002,ICLKGEOUTPUT,5,0,1,1,
XPJC4,S(82,91,0,79)*
XPJC1,1,301002,ICLKGEOUTPUT,6,0,1,1,
XPJC4,S(92,101,0,79)*
```

```
XPJC1,1,301002,ICLKGEOUTPUT,7,0,1,1,
XPJC4,S(102,111,0,79)*
XPJC1,1,301002,ICLKGEOUTPUT,8,0,1,1,
XPJC4,S(112,121,0,79)*
XPJC1,1,301002,ICLKGEOUTPUT,9,0,1,1,
XPJC4,S(122,131,0,79)*
XPJC1,1,301002,ICLKGEOUTPUT,10,0,1,1,
XPJC4,S(132,141,0,79)*
XPJC1,1,301002,ICLKGEOUTPUT,11,0,1,1,
XPJC4,S(142,151,0,79)*
XPJC1,1,301002,ICLKGEOUTPUT,12,0,1,1,
XPJC4,S(152,161,0,79)*
XPJC1,1,301002,ICLK G2 LOG1,1,0,1,1,
XPJC4,S(162,163,0,79)*
XPJC1,1,301002,ICLK G2 LOG1,2,0,1,1,
XPJC4,S(164,165,0,79)*
XPJC1,1,301012,ICLK G2 LOG1,3,0,1,1,
XPJC4,S(166,167,0,79)*
****
```

XPEU:-

```
CDF       XPEBOOTSTRAP,0,0,2
IMT       PROGRAM GPLT,1
IP        XPMO,3,MT
IP        XPEO,6D,MT
IP        XPXX,4,MT
PRT
F
```

XKEE:-

```
XKEE I6
XKEE O12
XKEE B2/128
XKEE S*00340300
XKEE C301002
XKEE L4650000
XKEE #GEOA
DUMP CD
GEOA #GEOB
DUMP CD
GEOB #GEOC
DUMP CD
GEOC #GEOD
DUMP CD
****
```

XKYE:-

```
XKYE I6
XKYE B1/128
XKYE S*01000000
XKYE C301002
DUMP CD
****
```

XKZE:-

```
XKZE O12
XKZE B1/128
XKZE S*01000000
XKZE C301002
DUMP CD
****
```

XPEU:-

```
CDF       PROGRAM GEOR,0,0,2
IP        GEOA,9G,CD
CEG       GEOA,9G,3335
IP        GEOB,9G,CD
CEG       GEOB,9G,3335
IP        GEOC,9G,CD
CEG       GEOC,9G,3335
IP        GEOD,9G,CD
CEG       GEOD,9G,3335
IP        XKYE,9G,CD
CEG       XKYE,9G,1682
IP        XKZE,9G,CD
CEG       XKZE,9G,1702
IMT       PROGRAM GEOS,1
IP        XKWE,9G,MT
IP        XKXE,9G,MT
PRT
F
```

Appendix 8 – Example Parameters MAXIMOP on EDS 8

XPJ1:-

```
XPJ11,ED,301003,203,*
****
```

XPJ2:-

```
XPJ21,1,301003,S,*
XPJ21,3,301003,1,0,*
****
```

XPJC:-

```
XPJC1,1,301003,XPEBOOTSTRAP,0,0,3,1,
XPJC4,S(4,4,0,79)*
XPJC1,1,301003,JOB NUMBERS,0,0,1,1,
XPJC4,S(5,5,0,79)*
XPJC1,1,301003,MAXIMOP,0,0,1,1,
XPJC4,S(6,10,0,79)*
XPJC1,1,301003,MAXIMOP SWAP,0,0,1,1,
XPJC4,S(11,50,0,79)*
XPJC1,1,301003,MAXIMOP WORK,0,0,1,1,
XPJC4,S(51,57,0,79)*
XPJC1,1,301003,PROGRAM MAXI,0,0,3,1,
XPJC4,S(58,93,0,79)*
XPJC1,1,301003,PROGRAM MAXU,0,0,3,1,
XPJC4,S(94,101,0,79)*
XPJC1,1,301003,MAXIMOP SOFA,1,0,1,1,
XPJC4,S(102,104,0,79)*
XPJC1,1,301003,MAXIMOP SOFA,2,0,1,1,
XPJC4,S(105,107,0,79)*
XPJC1,1,301003,MAXIMOP SOFA,3,0,1,1,
XPJC4,S(108,110,0,79)*
XPJC1,1,301003,MAXIMOP SOFA,4,0,1,1,
XPJC4,S(111,113,0,79)*
XPJC1,1,301003,MAXIMOP SOFA,5,0,1,1,
XPJC4,S(114,116,0,79)*
XPJC1,1,301003,MAXIMOP SOFA,6,0,1,1,
XPJC4,S(117,119,0,79)*
XPJC1,1,301003,MAXIMOP SOFD,0,0,1,1,
XPJC4,S(120,122,0,79)*
XPJC1,1,301003,UMBRELLAFILE,0,0,1,1,*
XPJC1,1,301003,UMBRELLACOPY,0,0,1,1,*
XPJC1,1,301003,MACROS,0,0,2,1,
XPJC4,S(123,123,0,79)*
XPJC1,1,301003,LIBRARY,0,0,2,1,
XPJC4,S(124,126,0,79)*
XPJC1,1,301003,MAXIHELPTEXT,0,0,2,1,
XPJC4,S(127,129,0,79)*
XPJC1,1,301003,COMMON 1,0,0,2,1,
XPJC4,S(130,130,0,79)*
XPJC1,1,301003,COMMON 2,0,0,2,1,
XPJC4,S(131,131,0,79)*
XPJC1,1,301003,COMMON 3,0,0,2,1,
XPJC4,S(132,132,0,79)*
XPJC1,1,301003,COMMON 4,0,0,2,1,
XPJC4,S(133,133,0,79)*
```

```
XPJC1,1,301003,COMMON 5,0,0,2,1,
XPJC4,S(134,134,0,79)*
XPJC1,1,301003,COMMON 6,0,0,2,1,
XPJC4,S(135,135,0,79)*
XPJC1,1,301003,COMMON 7,0,0,2,1,
XPJC4,S(136,136,0,79)*
XPJC1,1,301003,COMMON 8,0,0,2,1,
XPJC4,S(137,137,0,79)*
XPJC1,1,301003,COMMON 9,0,0,2,1,
XPJC4,S(138,138,0,79)*
XPJC1,1,301003,BRIAN-11,0,0,2,1,
XPJC4,S(139,141,0,79)*
XPJC1,1,301003,BILLG-21,0,0,2,1,
XPJC4,S(142,144,0,79)*
XPJC1,1,301003,PROGRAM CMGR,0,0,3,1,
XPJC4,S(145,154,0,79)*
****
```

XPEU:-

```
CDF       XPEBOOTSTRAP,0,0,3
IMT       PROGRAM GPLT,1
IP        XPMO,3,MT
IP        XPEO,6D,MT
IP        XPXX,4,MT
PRT
F
```

QMEF:-

```
QMEF,301003
123,144,0,79
****
```

QMEF:-

```
QMEF,301003,UMBRELLACOPY
123,144,0,79
****
```

QMFS:-

```
QMFS
COMMON 1,0000
COMMON 2,0000
COMMON 3,0000
COMMON 4,0000
COMMON 5,0000
COMMON 6,0000
COMMON 7,0000
COMMON 8,0000
COMMON 9,0000
BRIAN-11,0000
BILLG-21,0000
****
```

QLJU:-

```
*SETUP
B******* 0 (0000) EGI
*FINISH
```

QMMA:-

```
CART !
CART 301003
CART 301002
CART 301011
CART 301010
FORB !
FORB PROGRAM MAXU
FORB PROGRAM GEOR
FORB ICLKHARDFILE
READ !
READ PROGRAM UTIL
READ PROGRAM COMP
READ SUBGROUPS-RS
READ SUBGROUPS-CP
READ SUBGROUPSRA4
READ SUBGROUPSRF4
READ XPEBOOTSTRAP
CORE 24000
MAXC 22000
SAVE
GO
```

Appendix 9 – Example Parameters GEORGE 2+DOF/1T on EDS30/60

XPJ1:-

```
XPJ11,ED,311003,203,*      (for EDS60, adjust for EDS30)
****
```

XPJ2:-

```
XPJ21,1,301003,S,*
XPJ21,3,301003,1,0,*
****
```

XPJC:-

```
XPJC1,1,311000,XPEBOOTSTRAP,0,0,3,1,
XPJC4,S(1,1,120,295)*
XPJC1,1,311003,PROGRAM GE2P,0,0,3,1,
XPJC4,S(2,6,0,295)*
XPJC1,1,311003,DIMEDSCWOVER,0,0,1,1,
XPJC4,S(7,7,0,295)*
XPJC1,1,311003,DIMEDKWHOVER,0,0,2,1,
XPJC4,S(8,8,0,295)*
XPJC1,1,311003,DIMEDHKWOVER,0,0,2,1,
XPJC4,S(9,9,0,295)*
XPJC1,1,311003,DIMEDHLWOVER,0,0,2,1,
XPJC4,S(10,10,0,295)*
XPJC1,1,311003,DIMEDKOWOVER,0,0,2,1,
XPJC4,S(11,11,0,295)*
XPJC1,1,311003,DIMEDKLWOVER,0,0,2,1,
XPJC4,S(12,12,0,295)*
XPJC1,1,311003,ICLKGEOGWMCQ,0,0,1,1,
XPJC4,S(13,15,0,295)*
XPJC1,1,311003,ICLKCODEDJDS,0,0,1,1,
XPJC4,S(16,16,0,295)*
XPJC1,1,311003,ICLKCODEDJDS,1,0,1,1,
XPJC4,S(17,17,0,295)*
XPJC1,1,311003,ICLKCODEDJDS,2,0,1,1,
XPJC4,S(18,18,0,295)*
XPJC1,1,311003,ICLKCODEDJDS,3,0,1,1,
XPJC4,S(19,19,0,295)*
XPJC1,1,311003,ICLKCODEDJDS,4,0,1,1,
XPJC4,S(20,20,0,295)*
XPJC1,1,311003,ICLKCODEDJDS,5,0,1,1,
XPJC4,S(21,21,0,295)*
XPJC1,1,311003,ICLKCODEDJDS,6,0,1,1,
XPJC4,S(22,22,0,295)*
XPJC1,1,311003,ICLKCODEDJDS,7,0,1,1,
XPJC4,S(23,23,0,295)*
XPJC1,1,311003,DIMEGEOGMCRO,0,0,3,1,
XPJC4,S(24,24,0,295)*
XPJC1,1,311003,DIMEUSEREMPT,0,0,3,1,
XPJC4,S(25,25,0,295)*
XPJC1,1,311003,DIME DA TRAP,0,0,3,1,
XPJC4,S(26,26,0,295)*
XPJC1,1,311003,DIME MT TRAP,0,0,3,1,
XPJC4,S(27,27,0,295)*
```

```
XPJC1,1,311003,ICLK G2 LOG1,0,0,3,1,
XPJC4,S(28,28,0,295)*
XPJC1,1,311003,ICLK G2 LOG1,1,0,3,1,
XPJC4,S(29,29,0,295)*
XPJC1,1,311003,ICLK G2 LOG1,2,0,3,1,
XPJC4,S(30,30,0,295)*
XPJC1,1,311003,ICLK G2 MSGE,0,0,3,1,
XPJC4,S(31,31,0,295)*
XPJC1,1,311003,ICLK G2 MSGE,1,0,3,1,
XPJC4,S(32,32,0,295)*
XPJC1,1,311003,ICLK G2 MSGE,2,0,3,1,
XPJC4,S(33,33,0,295)*
XPJC1,1,311003,ICLKMONJRNL1,0,0,3,1,
XPJC4,S(34,34,0,295)*
XPJC1,1,311003,ICLKMONJRNL1,1,0,3,1,
XPJC4,S(35,35,0,295)*
XPJC1,1,311003,ICLKMONJRNL1,2,0,3,1,
XPJC4,S(36,36,0,295)*
XPJC1,1,311003,DSCW PM DUMP,0,0,1,1,
XPJC4,S(37,37,0,295)*
XPJC1,1,311003,ICLKLASERSET,0,0,1,8,
XPJC4,S(38,38,0,295)*
****
```

XPEU:-

```
CDF       XPEBOOTSTRAP,0,0,63
IMT       PROGRAM GPLT,1
IP        XPMO,3,MT
IP        XPEO,6D,MT
IP        XPXX,4,MT
PRT
F
```

Appendix 10 – Utility Program Specifications

#DCPD	Write DCPs and OLTs to ICLKHARDFILE
#ENGD	Construct an Engineers' Library on EDS 8
#ENGF	Construct an Engineers' Library on EDS 30/60
#EXD1	Copy EWG3 to disc

A10.1 #DCPD – Write DCPs and OLTs to ICLKHARDFILE

A10.1.1 TITLE

To create/update/copy ICLKHARDFILE on EDS30/60/200. Write DCPs and OLTs to the library file from magnetic tape and write those programs back to magnetic from EDS. This program is written for the requirements of West Gorton, rather than Stevenage, systems. (Details from observation, may be incomplete/incorrect.)

A10.1.2 HARDWARE REQUIREMENTS

5888 words of core store
1 paper tape reader and/or 1 card reader
1 or more EDS transports (EDS30, EDS60 or EDS200)
1 or more MT transports

A10.1.3 USE OF PERIPHERALS

TR0 and/or CR0	Parameters	The readers are allocated and released as required
DA0 (optional)	Program file from which insertions are to be made	This is opened when the parameter calling for it is obeyed and remains open until another file is required or the end of the run
DA1	Program file to be updated or created	The ICLKHARDFILE is opened at the start of the run and released at the end
DA2 (optional)	The system file of the disc on which the level 1 bootstrap is written (sector 0)	This is opened when the L1 bootstrap is to be written and closed shortly after
MT0	Program tape from which the programs are to be loaded	The tape PROGRAM DCPS is opened when the parameter calling for it is obeyed and remains open until the end of the run
MT1 (optional)	Program tape to which the programs are to be copied	The tape PROGRAM DCPS is opened when the parameter calling for it is obeyed and remains open until the end of the run

#DCPD

A10.1.4 DESCRIPTION

#DCPD is a free-standing re-enterable program which uses a permanent file with the reserved name of ICLKHARDFILE. This file is used to contain executive code (Stevenage only) programs (Executives, Test Programs, etc.), DCPs, OLTs and Microcode, which can be rapidly loaded from the disc. Bootstraps for DCPs can also be written by this program.

#DCPD is controlled by means of parameters input from paper tape, cards or the console typewriter (by means of the 'MS' facility; switches are used to supplement the parameters in defining the action to be taken. DCPs, OLTs and bootstraps can be input to the library file from magnetic tape or another ICLKHARDFILE.

A10.1.5 METHOD EMPLOYED

The parameters are composed of a sequence of directives, with optional switch settings, to copy programs to ICLKHARDFILE from magnetic tape, optionally creating or updating the file.

The first level DCP bootstrap is written to sector 0 (absolute) and its second level to blocks 0-5 of the ICLKHARDFILE, while a single level Executive bootstrap is written to sectors 6-9 of the ICLKHARDFILE by #ENGF. Either #DCPD or #SS33 **must** be run to initialise ICLKHARDFILE before running #ENGF.

A new disc is set up in the standard way using #XPJ1, #XPJ2 and optionally #XPJC. If the file is be created manually it must have 4 blocks per bucket, an integrity code of 2 and own monitoring of flaws. For EDS60/200, the ICLKHARDFILE must be retained on cylinder numbers not greater than 255. For all EDS types the file must be a single area.

A10.1.6 PROGRAM SWITCHES

Switch	0	ON	–	Append instead of overwrite (set when requested)
	2	ON	–	No error when file the extends beyond cyl. 255 (DCPD7)
	3	ON	–	Only copy ED30 DCPs and OLTs from MT
	4	ON	–	Only copy ED60 DCPs and OLTs from MT
	5	ON	–	Unknown, used with sw.6
	6	ON	–	Only copy ED200 DCPs and OLTs from MT
	7	ON	–	Unknown
	8	ON	–	Verbose mode
	10	ON	–	Put non-2812 programs into the file
	23	ON	–	Unknown, internal use?

A10.1.7 OPERATING INSTRUCTIONS

1.		Load the required EDS cartridge.	
2.		Load/Find #DCPD	HALT:- LD
3.		Either:-	
		a) For parameters on PT, input:	GO #DCPD 20
	or	b) For parameters on Cards, input:	GO #DCPD 21
	or	c) For parameters from the console:	GO #DCPD 22
	or	d) ???:	GO #DCPD 23
	or	e) Automatic DCPD6, x1=csn and GO25:	GO #DCPD 24
	or	f) Automatic DCPD5, x1=csn:	GO #DCPD 25
	or	g) For next directive (after malfunction):	GO #DCPD 28
	or	h) To abandon the run:	GO #DCPD 29
4.		If parameters from console	
		Under E6RM use	MS#DCPD=parameter
		Under G3, program will HALT:- QQ	RM ,PARAM(parameter)
5.		At the end of run	RUN COMPLETE

A10.1.8 EXCEPTION CONDITIONS

Typewriter Messages

Message	**Reason**	**Action**
HALT - 'ON 0' TO APPEND		GO to continue
	Set switch if initialise not required	
AL X1 TO CSN		GO to continue
	Alter required csn into x1	

DISP - CSN nnnnnn
EDS xx LEV1 B/S VERSION n WRITTEN
EDS xx LEV2 B/S VERSION n WRITTEN
ppppppppppp WRITTEN AS x
EZ5A/nnnnn WRITTEN AS x

HALT - INPUT FILE NOT ON CYL BOUNDARY
HNON-STANDARD O/P FILE
DCP'S OK BUT #SS26/RDCP/GEO3 NO GO
xxxx INDEX TOO SMALL FOR PROG

#DCPD

FILE TOO SMALL FOR SPECIFIED AREAS
NO OLTS IN A SHORT FILE
DUMP AREA TOO LARGE FOR EXEC AREA
NON-UDAS EXEC
HPRG xxxxxxxxxxxx NOT FOUND
CHECKSUM ERROR ON MT
BLOCK COUNT ERROR ON MT
CREATE FAIL
UNABLE TO EXTEND FILE
UNABLE TO DELETE FILE USE XPJC
FILE DELETED
FILE NOT LARGE ENOUGH
FAIL TYPE x OPENING FILE
READ FAIL
TRANSFER CROSSES CYL BOUNDS-CORRUPT
LBN OUT OF RANGE-PROG OR FILE CORRUPT
BOOTSTRAP WRITE FAIL
NEED CORE
RUN ABANDONED
NO AREAS SPECIFIED
NO B/S EMBEDDED IN PROG
USE GO 22
ICLKHARDFILE NOT ON DISC
INTEGRITY CODE FAILURE
SCA NOT AVAILABLE FOR READING
SCA NOT AVAILABLE FOR WRITING
SCA FULL
ICLKHARDFILE ALREADY ON DISC
NOT ENOUGH EXEC CORE
DISC NOT ONLINE
NO SPACE BELOW CYL 255
CYL 2(+) IN USE
PRG x ALREADY IN xxxx INDEX
FILE CREATED CYL xxx-xxx
PROG x IN xxxx INDEX IS NONSTANDARD
PROG. IGNORED
HARDFILE IS NOT 4 BLOCK BUCKETS
32 DISCS USED - NO MORE ACCEPTED
xxxxK FOR EXEC DUMPS
INDEX BUCKETS
PRG X - BUCKETS IN XXXX INDEX
THIS DISC IS G3 FILESTORE DISC
VERSION OF LEV? B/S
DISC SIZES DIFFER
FILE GOES BEYOND CYL 255
INPUT FILE NOT INITIALIZED!!

#DCPD

A10.1.9 PARAMETERS

Input parameters start as the first character of a line or card and take the form:-

```
DCPDn,parameters,*
```

The parameter format must be rigorously adhered.

Specify Program Data Area Sizes

This parameter allows you to specify how much space, if any, to allow for each type of program to be held in a newly initialised ICLKHARDFILE.

```
DCPD0,cd,b1,b2,b3,b4,b5,*
```

cd	is the size of the executive postmortem dump area in KWORDS (1 KWORD = 2).
b1-5	are the numbers of buckets assigned to Stevenage Executives, 2812, 7903, 7901 DCPs and OLTs, respectively.

Copy Specified Program from another ICLKHARDFILE

This parameter allows you to copy specified program from another file into the current ICLKHARDFILE being created/updated.

```
DCPD1,id1,icsn,id2,ocsn,*
```

id1	is the program identifier to be copied.
icsn	is csn of the ICLKHARDFILE to copy from.
id2	is the program identifier to be given to the new program.
ocsn	is csn of the ICLKHARDFILE where the program is to be inserted.

Insert Specified Program from a Library Tape

This parameter allows you to copy specified program from the library tape PROGRAM DCPS into ICLKHARDFILE.

```
DCPD2,name,id,csn,*
```

name	is the name of the program (12 characters) on the tape.
id	is the program identifier to be given to the program.
csn	is csn of the ICLKHARDFILE where the program is to be inserted.

#DCPD

Copy Specified Program to a Library Tape

This parameter allows you to copy specified program from ICLKHARDFILE to a library tape PROGRAM DCPS.

```
DCPD3,id,csn,name,*
```

id is the program identifier to be copied.
csn is csn of the ICLKHARDFILE from which the program is to be copied.
name is the name to be given to the program (12 characters) on the tape.

Copy ICLKHARDFILE from one cartridge to another

This parameter allows you to copy a complete ICLKLHARDFILE to another cartridge. Both files need to have the same specification (indexes, size, etc.)

```
DCPD4,icsn,ocsn,*
```

icsn is csn of the ICLKHARDFILE to copy from.
ocsn is csn of the where the ICLKHARDFILE is to be copied to.

Write L2 Bootstrap and Copy All Programs from a Library Tape

This parameter causes a level 2 bootstrap and all programs on the Library Tape (PROGRAM DCPS) to be written to the specified ICLKHARDFILE .

```
DCPD5,csn,*
```

csn is the csn of the ICLKHARDFILE to be written to.

Write L1 Bootstrap and Optionally Initialise Indexes

This parameter causes a level 1 bootstrap to be written to block 0 of the specified cartridge and optionally the indexes of the ICLKHARDFILE to be initialised. Whether or not the indexes are initialised depends on the setting of Switch 0 when asked (see 5.1.8).

```
DCPD6,csn,*
```

csn is the csn of the ICLKHARDFILE to be written to.

Create a new ICLKHARDFILE

This parameter causes a new ICLKHARDFILE of random fgn to be created on the specified cartridge. The size of the file created depends on the DCPD0 parameters.

```
DCPD7,csn,*
```

csn is the csn where the ICLKHARDFILE is to be created.

Delete Specified ICLKHARDFILE

This parameter allows you to delete an existing ICLKHARDFILE.

```
DCPD8,csn,*
```

csn is the csn of the ICLKHARDFILE to be deleted.

.

Display Program Data Area Size Table

This parameter allows you to see what sizes are currently set in #DCPD.

```
DCPD?,*
```

5.1.9.11 Parameter Terminator

This sequence of parameters is terminated by record containing 4 stars

```
****
```

A10.1.10 EXAMPLE SET OF DIRECTIVES

The following example will create, initialise and insert bootstraps and programs into an ICLKHARDFILE.

```
DCPD0,0,0,53,26,0,32,*
DCPD7,444200,*
DCPD6,444200,*
DCPD5,444200,*
****
```

This run will:-

- Set program data areas up for 2812 DCPs (53 buckets), 7903 DCPs (26 buckets) and 2812 OLTs (32 buckets)
- Create a new file on cartridge 444200
- Write a L1 bootstrap to Block 0 of the cartridge
- Initialise all of the indexes
- Write a L2 bootstrap to the file
- Insert all program from a library tape to the file

A10.2 #ENGD – Construct an Engineers' Library on EDS 8

A10.2.1 TITLE

To install Operators' Executives and Executive Mode Programs into an Engineers' Library (ICLKEXECFILE) on EDS 8, with a Bootstrap for loading them, in Block 0 of the same cartridge; also print a list of the programs in ICLKEXECFILE.

A10.2.2. HARDWARE REQUIREMENTS

5120 words of core store
1 paper tape reader and/or 1 card reader
1 line printer
1-3 EDS 8 cartridges
0-2 MT transports

A10.2.3. USE OF PERIPHERALS

TR0 and/or CR0	Parameters	The readers are allocated and released as required
LP0	Report	The printer is allocated and released as required
DA0	Engineers' library to be updated or created	This ICLKEXECFILE is opened and closed as required
DA1 (optional)	Engineers' library to be updated or created to access the program list	The ICLKEXECFILE is opened when required to read the 'program list' and then closed
DA1 (optional)	The system file of the disc on which the level 1 bootstrap is written (sector 0)	This is opened when the L1 bootstrap is to be written and closed shortly after
DA2 (optional)	Engineers' library to be copied from	The ICLKEXECFILE is opened when required to read a program
DA3 (optional)	The system file of the disc on which the level 1 bootstrap is written (sector 0)	This is opened when the when the system file is to be read and closed shortly after
MT0 & MT1 (optional)	Program tape from which the executive mode programs are to be loaded	The specified tape is opened when the parameter calling for it is obeyed and remains open until the end of the run

#ENGD

A10.2.4 DESCRIPTION

#ENGD first reads a Steering Line stating whether the output file is empty or not, what kind of Bootstrap is wanted, and what sources are to be input.

If opening an empty output file, it writes an Extension Bootstrap to the first 2 buckets, and clears the Program List; if amending the output file, it reads the existing Program list and prepares to add to it.

It then transfers the sources to EDS8, in the order given in the Steering Line.

It writes the Program List, and writes the Bootstrap (containing pointers to ICLKEXECFILE) to Bucket 1 of the SYFLE/SYFIL of the cartridge, and also to the last bucket of ICLKEXECFILE.

It finally checks the output file, printing a list of its contents. An entry point (22) is provided to do this step only.

A10.2.5 STEERING LINES

This list of directives is read at the start from paper tape or cards, one line or card per directive. Each directive consists of a letter and parameter(s) separated by solidi (/) and terminating in Newline (tape) or ^ (cards).

Directives are as follows:-

O or A or R (Mandatory)

The first directive defines what action is to be taken on opening the output file, and gives its generation number (g) and, if needed, the location and size of its single File Area via its first and last cylinders (a) and (b) as set previously in the parameters for the File Allocator (#XPJC or #XJEC). One of the 3 following directives must head the steering line:-

Open a new ICLKEXECFILE : format:-

O/g/a/b

Program Action

Open ICLKEXECFILE, generation g
Set starting address and size from a and b
Write Extension Bootstrap to 1st 2 flaw-free buckets
Clear Program List
Prepare to write to 3rd flaw-free bucket

Example : O/19/23/94

#ENGD

Amend an existing ICLKEXECFILE : format:-

A/g

Program Action

Open ICLKEXECFILE, generation g, get csn from reply
Open SYFLE or SYFIL, read bootstrap and close
Read Program List from position defined in bootstrap
Prepare to write over old Program List

Example : A/1234

Restore bootstrap lost by other software, and Amend:-

R/g/a/b

Where a and b are the new first and last cylinders (b-a) must be the same as the original.

Program Action

Open ICLKEXECFILE, generation g
Read last bucket, containing copy of original bootstrap
Check file size in bootstrap against b-a. Set new value of a.
Read Program List and continue as in 4.1.2

Example : R/19/33/104

B (Mandatory)

This directive defines the Bootstrap type required. There are 4 forms:-

B/4 : 1904/5/9 processors
B/6 : 1906/7, 1904E/F, 1905E/F, 1904A, 1904S, 1903T single processors
B/6A : 1906A, 6S processors
B/6D : Dual processors

C – Copy EDS

Format:-

C/g

Up to 2 such directives may be included in one run.

#ENGD

Program Action

Copy from ICLKEXECFILE, generation g, all programs except those already present in the output file, and any due to be added from punched media and any listed for deletion

Example : C/913

M –Magnetic Tape

Format:-

M/Name of Tape (12 characters)

Up to 2 such directives may be included in one run.

Program Action

Copy from the Magnetic Tape named, all Executive Mode programs including Main Executives, except except those already present in the output file

Example : M/PROGRAM ELIB

T, X : Programs from paper tape or cards

These directives supply a Library Number and description to be attached to an Executive Mode program read from paper tape or cards. The initial letter (a) of the directive distinguishes between the expected input formats, and an asterisk (*) should follow it if the program is punched for the 2102 (reduced facility, decoder E) card reader.

Format:-

a/1234/DESCRIPTION OF NOT MORE THAN 55 CHARACTERS

where 1234 represents the library number, and a is as follows:-

T : Test program dumped by #NSBL or #LBRY
X : EXecutive

Up to 16 such directives may be included in one run.

Program Action

Copy the program from paper tape or cards into the output file. If the library number is already in the Program List, overwrite the old list entry, otherwise add it on the end. If copying EDS or Magnetic Tape, omit any programs with this Library Number when doing so.

Examples:-

T*/9876/FLIT6257804T FOR OUR PROCESSOR
X/0056/E6RM FOR COMPUTERS ANONYMOUS

D –Delete

Format:-

D/1234

Up to 128 D+X+T directives may be given.

Program Action

When copying from EDS or Magnetic Tape, omit the program with Library Number 1234.

N.B. This does not delete a program present in the output file.

E –End

Format:-

E

A10.2.6 EXPECTED FORMATS OF INPUT DATA

Paper Tape or Cards

These consist of:-

(i) Any amount of bootstrap
(ii) N data blocks (binary type 0) in either ascending or descending order
(iii) Entry block (Binary type 2-4) preferably containing a block count

The bootstraps are recognised (and ignored) by not consisting of standard binary blocks, except for the 1904 paper tape Executives, whose bootstrap has no Newlines and is assumed to end at the first runout.

The block count in the entry block is N+2 for all T-programs and 1904 Executives, and N+1 for other Executives.

Magnetic Tapes

Programs to be copied from magnetic have the following form:-

(i) A 5 word qualifier block following a file mark:-

Word 0 = -1
Word 1 = -20
Word 2 = Library number in character form
Word 3 = Spare
Word 4 = Checksum of words 0-3 (negative)

(ii) Any number of pairs of pre-data (binary type 5) and data blocks with a maximum length of 512 words whose destinations must all be multiples of 512 unless less than 512.

(iii) An Entry Block (binary type 2-4)

(iv) A 512-word Information Block whose first record is a description in the form:-

Word 0	:	No. of words in record
Words 1 onwards	:	DESCRIPTION OF NOT MORE THAN 55 CHARACTERS

N.B. The old format, with no Entry Block, and where the last Data Block had destination 0 and contained the description (words 0-15), word count (17), block count (18) and entry jump (22) is still permitted.

The Library Tape may contain other types of data (for example, Normal Mode Programs) interspersed with programs in the above format, but these will be ignored.

#ENGD

ICLKEXECFILEs on EDS8

The expected format is described below in section A10.2.7.

A10.2.7 FORMAT OF COMPLETED ICLKEXECFILE

The bucket length is one block (128 words). The ICLKEXECFILE occupies a continuous area of one cartridge and is not split over several areas.

Block 0 of the cartridge contains a bootstrap, plus some file parameters.

The first two buckets of the file contain an extension bootstrap and this is followed by Executive Mode programs whose format is described below. The last program is followed by a 4-bucket Program List.

The last bucket of the file contains a copy of the bootstrap in Block 0, for use by the "R/g/a/b" facility.

Bootstrap Block (Bucket 1 of SYFLE csn/SYFIL csn)

This contains a bootstrap program, a "Position and Read" subroutine, the starting cylinder number of ICLKEXECFILE, and the bucket numbers of the Extension Bootstrap and Program list.

Programs

Data is written in blocks whose destinations are multiples of 128, omitting all-zero blocks. The description is placed in words 0-15 of the program.

Each data block has its destination, bucket number and checksum listed in a "Post-Data Block", which may contain up to 41 such entries. When a Post-Data Block is full, it is written to the next bucket, and a fresh one started containing a pointer to the first, forming a chain.

The last (usually the only) Post-Data Block serves also as an Entry Block. Word 0 contains a flag, word 126 contains the Entry Jump, and its bucket number is entered in the Program List together with the Library Number.

When a program is replaced (via X or T) in an "Amend" run, the new version is appended to the programs already present, and the new position entered in the Program List. This leaves a dead area in the file, occupied by the old version, which is no longer accessible.

Post-Data Block Format

Word	Contents	Description
* 0	*73000X06	Binary block type
1	1234	Library Number in characters
2	N/O	Number of data entries, as a counter/modifier
3	M	Bucket number of previous post-data block. Zero if none.
4	Destination	3 word entry for data.
5	Bucket number	
6	Checksum (-ve)	
7	Destination	3 word entry for data.
8	Bucket number	
9	Checksum (-ve)	
Repeating 3 word entries		
124	Destination	3 word entry for data.
125	Bucket number	
* 126	Checksum (-ve)	
127	Checksum (-ve)	Negative checksum of words 0-126

* For an “Entry Block”, X=1 and word 126 contains “074 0 /Entry Point”. Otherwise X=0.

Program List

This occupies the first 4 buckets of ICLKEXECFILE. The format is:-

Word	Contents
0	2N = count of N programs: maximum N=255
1	Negative checksum of whole list
2	Library Number of first program
3	Bucket Number of “Entry” Post-Data Block
--	----------
--	----------
2N	Library Number of last program
2N+1	Bucket Number of “Entry” Post-Data Block

A10.2.8 ERROR ACTION

#ENGD acts on errors in one of 3 ways; depending on the gravity, it

DISPLAYS a message and continues (see 7.1)
HALTS if the operator can take some remedial action (7.2)
ABANDONS the run after DISPLAYing the reason (7.3)

DISPLAY message

When copying from magnetic tape (via M), if a program's format is not as in 5.2, #ENGD will leave it out and display:-

1234 OMITTED: MT FORMAT

Giving the library number

HALTED messages

These are used where it s possible to correct the error and continue, or to try repeating some failed transfer. In general there are 3 courses open to the operator:-

(a) correct the fault and repeat
(b) omit the offending item and continue (if applicable)
(c) abandon the run, by "GO #ENGD 29"

The messages are:-

NEEDS TR or CR or LP

#ENGD needs a peripheral. Make one available and "GO #ENGD"

?? Steering Line Record

The last steering line directive read has the wrong format.

Remedy

(a) Correct and reload the rejected line/card, and "GO #ENGD"
(b) "GO #ENGD", and that directive will be omitted
(c) Abandon by "GO #ENGD 29"

For correct formats, see:-

Directive	Directive	Section
A	M	4.4
B	O	4.1.1
C	R	4.1.3
D	T	4.5
E	X	4.5

And remember that the first directive must be A, O, or R.

++ Steering Line Record

There are already enough steering line directives of this type.

Remedy

As above.

UNOPENED X: Filename and generation

X is the reply from Executive on failing to open the file named; it has the following values:-

X	Reason
1	File not in system
2	Failure on Integrity Code check
3	Insufficient store to hold further file
6	Another file open when trying to open System File for writing
7	System File is open for writing

Remedy

(a) Check that the file is on-line, or that other programs are not using EDS, as appropriate, and "GO #ENGD."

The "open" PERI will be repeated.

(c) Abandon by "GO ENGD 29"

LIST SUM FAIL

This could occur at the start of an Amend run, after reading the existing Program List.

Remedy

(a) "GO #ENGD", the Program List will be re-read

(c) Abandon by "GO #ENGD 29"

1234 EDS FAULT: SW?

This indicates some checksum or format error in program 1234 being copied from EDS (via C).

Remedy

(a) To repeat reading program 1234, "ON #ENGD 0" and "GO #ENGD"

(b) To omit the program and continue the run, "GO #ENGD"

(c) Abandon by "GO #ENGD 29"

(RELOAD BLOCK/CARD): SW?

Occurs during input of a punched source, if an individual block fails to checksum.

Remedy

(a) To repeat: reload the last paper tape block or card, "ON #ENGD 0" and "GO #ENGD"

(b) To omit: If the fault persists or if he block is genuinely damaged, "GO #ENGD"

(c) Abandon: If there is no point in continuing without the item, "GO #ENGD 29"

(RELOAD PROG): SW?

Occurs on reading the Entry Block of a punched source if the block count therein is not the number of blocks read.

Remedy

As above, reloading the tape/pack from the start if using option (a).

(SORT PROG AND RELOAD) : SW?

This indicates that the punched data block just read should have been packed into an EDS block already written. (More likely with cards than paper tape).

Remedy

As above, putting the offending card in the right place (the last word on each card is a binary sequence number) and reloading the whole pack if repeating.

EDS WRITE FAIL

#ENGD uses "own monitoring" on writing EDS (so that the bootstrap can find things without going through the System Control Area). If 10 attempts fail to write a block, it increases the Bucket number by 1 and tries again. If 9 consecutive Buckets fail, it halts with the above message.

Remedy

(a) "GO #ENGD", to continue on the next Bucket.

(c) Abandon by "GO #ENGD 29". If this occurs after abandoning the run, "DE #ENGD".

BLOCK 0 WRITE FAIL

This is similar to EDS Write Fail' above, when writing the Bootstrap block, but without increasing the Bucket number. At this point, the ICLKEXECFILE is in a completed state, with a copy of the bootstrap in its last block. It is useless for loading or copying without the bootstrap in Block 0, but it would be possible, after maintenance, to do a "Restore" run on it later.

Remedy

(a) "GO #ENGD", to repeat writing another 10 times.

(c) Abandon by "DE #ENGD".

DISPLAY messages before "RUN ABANDONED"

B MISSING

Occurs immediately after reading the steering line, if no B-directive was present.

Remedy

Correct the steering line and restart at 20 or 21.

FILE WRONG SIZE

Occurs in a "Restore" run (see 4.1.3) on reading the "spare bootstrap" from the "last" bucket of the output file. It is likely to be a steering line error in the R directive.

Remedy

Check the true location of the output ICLKEXECFILE, correct the steering line, and re-start at 20 or 21.

O/P FILE OBSOLETE FORMAT: NEEDS COPYING

Occurs in an "Amend" or "Restore" run if the output file was produced by #ENGD versions 1-4, which corrupted words 17-22 of every program. Version 5 introduces a change of format and a changed Extension Bootstrap to cope with it Since the Extension Bootstrap is written only by the "Open" run, the old files cannot be Amended.

Remedy

Create a new ICLKEXECFILE, and do an "Open" run on it, which includes Copying the old one.

LIST FULL

Occurs if the number of programs in the file passes 255 (see 6.4). In this case, #ENGD completes the file and goes on to read, check and print it.

Remedy

See what programs, if any, can be weeded out, create another ICLKEXECFILE, and use some D-directives in the steering line (see 4.6).

#ENGD

FILE FULL

Occurs if the file gets full. The run is completed as in 7.3.4, omitting the incomplete program.

Remedy

Create a new larger ICLKEXECFILE and copy the old one.

A10.2.9 FORMAT OF LINE PRINTER LISTING

Heading

The heading gives the file generation number, cartridge serial number, date of printing and the number of blocks still free in the form:-

ICLKEXECFILE 987 ON CARTRIDGE 007654 ON 25/12/74 465 BLOCKS LEFT

Contents

There is one line per program, containing:-

Line number
Library Number/Description
Lowest Bucket number
Number of buckets occupied

E.G.:-

99	0056/E6RM FOR ICL ISSUE 7	260	253
100	9604/TEST PROGRAM FOR TR	413	17

#ENGD

A10.2.10 OPERATING INSTRUCTIONS

File Creation

ICLKEXECFILE must occupy a single File Area with 1 block buckets, Integrity Code 2, Own Monitoring of flaws. Only one ICLKEXECFILE should be present on any one EDS cartridge. The generation number should be unique and non-zero.

Entry Points

20: Read steering line and any punched data from paper tape
21: Read steering line and any punched data from cards
22: Check file and print listing only. See 9.4
29: Abandon Run. See 9.5

Produce a Library in ICLKEXECFILE

1. Load #ENGD
2. Load the steering line on TR or CR and GO#ENGD 20 or 21.
3. #ENGD displays the 12-character name from its own Request Block.
4. If punched data is to be input (via X or T), watch for the console message

   #ENGD; DISPLAY:- GET 1234

Load the appropriate data in TR or CR and allocate; each such message is output in steering line order as soon as the reader is free of the previous data.

Normal Output Messages

As a guide to which files are currently in use, #ENGD displays 2 characters whenever it opens or closes a file, giving the unit number preceded by"O" (open) or "C" (closed). The unit numbers are:-

0	Output ICLKEXECFILE
1	Output SYFLE csn/SYFIL csn
2	Input ICLKEXECFILE
3	Input SYFLE csn/SYFIL csn

The normal sequence is:-

O0	Open output file
(O1, C1)	"Amend" only. To locate Program List
(O2, O3,	Input file if Copying

C3, C2)
C0 Close output file
O1, C1 Write SYFLE/SYFIL
O0, O1,
C1, C0 read and check (Sequence for GO22)

The program ends with:-

#ENGD; HALTED:- END OF RUN

If a second listing is required, “GO #ENGD 22” and see 9.4

To List an ICLKEXECFILE

1. Load #ENGD if necessary
2. GO #ENGD 22
3. #ENGD displays its name as above, then types

#ENGD; HALTED:- ALTER 0 TO GEN

At this point, if #ENGD is still in core after completing a run, X0 contains the Generation Number of the file last printed; if just loaded X0 contains the sign bit only, signifying “latest generation”.

If required, type e.g.

AL #ENGD 0 93 (for generation 93)

then

GO #ENGD

Normal Output Messages

#ENGD gives a sequence of 2 displays

O0, O1, C1, C0

Followed by

#ENGD; HALTED:- END OF RUN

#ENGD

To Abandon the run

Unless one of the WRITE FAIL messages has been typed (see 7.2.10-11)

GO #ENGD 29

To terminate the run.

After typing

#ENGD; DISPLAY:- RUN ABANDONED

Action depends on what stage has been reached so far. If it is in the writing stage (see 3.3), it completes a usable file be entering stage 3.4 and continuing. If it has not yet written to the file or is already in the printing stage, it releases all peripherals and ends

#ENGD; HALTED:- END OF RUN

After a persistent WRITE FAIL message, “GO 29” is ineffective since it will only try to write again, so the only course left is “DE #ENGD”.

Summary of HALTED error messages

NEEDS TR/CR/ LP)

?? Steering line record

++ Steering line record

UNOPENED X: Filename

EDS WRITE FAIL

BLOCK 0 WRITE FAIL

LIST SUM FAIL

Numb EDS FAULT:SW?

(RELOAD BLOCK/CARD):SW?

(SORT PROG & RELOAD):SW?

(RELOAD PROG):SW?

#ENGD

A10.2.11 BOOTSTRAPPING FROM ICLKEXECFILE

Operating Instructions

Load the cartridge on Unit 0 if possible, and use the normal bootstrap procedure for the channel, or if unit 0 is unusable, or if no hardware exists for bootstrapping the EDS8, load the cartridge on any unit, load the appropriate BTD (4, 6, D or A) via some other medium, and reply to

GIVE CHANNEL/UNIT

By typing, e.g.

29/2

BTDx then reads Block 0 and enters it.

Block 0 reads the Extension Bootstrap from ICLKEXECFILE, locating it via its own pointers and enters it.

The Extension Bootstrap types:-

LIB.NO/HOW MANY K?

Reply by typing, e.g.

0056/128

giving the Library Number (4 digits) of the required program, and store size in decimal.

It locates and loads the program, types out its description, winds it down to the proper store position and enters it.

Error Action

Transfers are repeated up to 10 times if they fail, and all checksums are checked. Space limits Block 0 to indicate errors by loop stops; the Extension Bootstrap types a message and returns to step 10.1.3 above.

The octal addresses of the loop stops in Block 0 are:-

Bootstrap Type			
4	6, 6D	6A	Meaning
112	41	43	Error status on reading Block 0
116	61	57	Error status on reading Extension
123	65	64	Checksum fail, Extension

The messages typed by the Extension are, e.g.:-

FAIL 169 2 6

Transfer failed 10 times on cylinder 169, Head 2, Sector 6.

NOT FOUND 1234

Library Number 1234 is not in the Program List.

DATA FAIL 1234 or DATA FAIL LIST

Checksum or format error in Program 1234 or in the Program List.

A10.3 #ENGF – Construct an Engineers' Library on EDS 30/60

A10.3.1 TITLE

To install Operators' Executives and Executive Mode Programs into an Engineers' Library (ICLKEXECFILE) on EDS 30/60, with a Bootstrap for loading them, on the same cartridge, normally into ICLKHARDFILE, but if this already holds the George Executive Bootstrap, #ENGF puts it into ICLKE6RMBOOT.

A10.3.2. HARDWARE REQUIREMENTS

5440 words of core store
1 paper tape reader and/or 1 card reader
1 line printer
1 EDS 30/60 cartridge
0-2 EDS 8 or 30/60 cartridges
0-2 MT transports

A10.3.3. USE OF PERIPHERALS

TR0 and/or CR0	Parameters	The readers are allocated and released as required
LP0	Report	The printer is allocated and released as required
DA0	Engineers' library to be updated or created	This ICLKEXECFILE is opened and closed as required
DA1 (optional)	Engineers' library to be updated or created to access the program list	The ICLKEXECFILE is opened when required to read the 'program list' and then closed
DA2 (optional)	Engineers' library to be copied from	The ICLKEXECFILE is opened when required to read a program
DA3 (optional)	The system file of the disc on which the level 1 bootstrap is written (sector 0)	This is opened when the when the system file is to be read and closed shortly after
MT0 & MT1 (optional)	Program tape from which the executive mode programs are to be loaded	The specified tape is opened when the parameter calling for it is obeyed and remains open until the end of the run

#ENGF

A10.3.4 DESCRIPTION

#ENGF constructs an ICLKEXECFILE from one or more of the following sources:-

(a) A previous ICLKEXECFILE on EDS30/60 or EDS8
(b) Binary punched paper tape or cards (but not both during one run)
(c) Magnetic tapes containing Executive Mode programs in the format described in Section 5.2

#ENGF first reads a Steering Line stating a) whether opening or amending the library, b) on which processor type it is to work, c) where to put the Bootstrap, and d) a list of sources, if any.

#ENGF next opens the output ICLKEXECFILE and finds its hardware addresses which it plants in the required Bootstrap before writing it to ICLKHARDFILE or ICLKE6RMBOOT.

Then, either starting from scratch or continuing where it left off last time, it works through the sources listed, reading and checking programs and writing them into ICLKEXECFILE.

When finished, it writes a Program List to the first 4 blocks.

Lastly, it reads and checks the entire file, listing the contents on a line printer. An entry point is provided to do this step only.

A10.3.5 STEERING LINES

If the steering line is punched on paper tape it must have one line per required item and if on cards each item must be punched on a new card and terminated by the character ^ (rows 11, 7, 8) to represent the *76 of newline on paper tape.

The types of items of which the steering file consists are as follows:-

The first item must be either A or O, and must appear only once in the steering file.

O/g Open a new ICLKEXECFILE with generation number g where g can have up to 4 decimal digits and must be different from the generation numbers of any EDS files used as sources.

A/g Amend an existing ICLKEXECFILE with generation number g.

In operation, "open" clears the program List and starts writing at Bucket 5, while "Amend" reads the existing List and starts writing where it left off previously.

B – Bootstrap type required on the new file. This item must appear once and only once, and has four permitted forms:-

B/4 (for 1904, 1905 and 1909 bootstraps)
B/6 (for 1906, 1907 and 1904E etc. bootstraps)
B/6A (for 1906A bootstrap)
B/6D (for Dual processor bootstrap)

G – George. The presence of G cause the Bootstrap to be put into ICLKE6RMBOOT since the cartridge is to be shared with George Executives. If G is absent, the Bootstrap is put in ICLKHARDFILE. (The combination "G" and "B/4" is forbidden.)

C/g – Copy all "unlisted" programs from an ICLKEXECFILE generation number g (where g is a number of up to 4 decimal digits).

"Unlisted" programs are those which are not already present in the EDS file under construction, not due to be added from punched media, nor listed for deletion.

M/name of tape – copy all "unlisted" Executive Mode programs from the named magnetic tape (12 character name). Normal Mode Programs, Overlays and Executive Listings are ignored.

X, X*, T, T* – Add Executive Mode punched program. Since these are punched in several formats this must be defined by the initial letter (a) of the steering line items, followed by the required library number and description.

e.g. a/1234/DESCRIPTION OF NOT MORE THAN 55 CHARACTERS (MIN 12)

where a is as follows:-

X Executive on paper tape or ordinary cards.
X* Executive on cards for 2102 (reduced facility) card reader
T Test program dumped by #NSBL on paper tape or ordinary cards.
T* Test program dumped by #NSBL for 2102 card reader

Any previous version of the program is cancelled, and it is omitted when copying an EDS or Magnetic Tape.

D/1234 – Omit the program whose library number is 1234 when copying an EDS or a Magnetic Tape. (Note: this does not cancel an existing program)

E – End of steering lines.

Limitations on the Number of Items in the Steering Line

These have been determined by considerations of storage space for the various list formed, and by practical experience of the number of items normally used. The following tables show the maximum number of items allowed:-

Item Type	Relationship Defining Limit
B	B = 1
M	M =< 2
C	C =< 2
A, O	A + O = 1
X, X*, T, T*	X + X* + T + T* < 16
D	D + X + X* + T + T* < 128

A10.3.6 EXPECTED FORMATS OF INPUT DATA

Punched Media

Executive Mode programs and Executives (input via steering line items T, T* and X, X* respectively) will be expected to begin with a bootstrap.

Paper tape : Any amount of bootstrap, ending in a newline or, for X, 1904, in runout.
Data blocks (binary type 0) with checksums.
Entry block (type 2, 3, or 4), the third word of which is the block count.
Cards : Any number of bootstrap cards.
Data and Entry blocks as for paper tape

Magnetic Tapes

Programs to be copied from magnetic will be expected to have the following form:-

(i) A 5 word qualifier block following a file mark:-

Word 0 = -1
Word 1 = -20
Word 2 = Library number in character form
Word 3 = Spare
Word 4 = Checksum of words 0-3 (negative)

(ii) Any number of pairs of pre-data (binary type 5) and data blocks with a maximum length of 512 words whose destinations must all be multiples of 512 unless less than 512.

(iii) An Entry Block (binary type 2-4)

(iv) A 512-word Information Block whose first record is a description in the form:-

Word 0	:	No. of words in record
Words 1 onwards	:	DESCRIPTION OF NOT MORE THAN 55 CHARACTERS

N.B. The old format, with no Entry Block, and where the last Data Block had destination 0 and contained the description (words 0-15), word count (17), block count (18) and entry jump (22) is still permitted.

The Library Tape may contain other types of data (for example, Normal Mode Programs) interspersed with programs in the above format, but these will be ignored.

#ENGF

ICLKEXECFILEs on EDS 30/60

The expected format is described below in section A10.3.7.

ICLKEXECFILEs on EDS 8

The expected format is the same as EDS 30/60 except that the Program List follows the last program, and its bucket numbers are contained in words 121-124 of block 0 of the cartridge (Bucket 1 of SYFLE).

A10.3.7 FORMAT OF COMPLETED ICLKEXECFILE

The bucket length is one block (128 words). The ICLKEXECFILE must be contained in one File Area only.

Programs

Data is written in blocks whose destinations are multiples of 128, omitting all-zero blocks. The description is placed in words 0-15 of the program.

Each data block has its destination, bucket number and checksum listed in a "Post-Data Block", which may contain up to 41 such entries. When a Post-Data Block is full, it is written to the next bucket, and a fresh one started containing a pointer to the first, forming a chain.

The last (usually the only) Post-Data Block serves also as an Entry Block. Word 0 contains a flag, word 126 contains the Entry Jump, and its bucket number is entered in the Program List together with the Library Number.

When a program is replaced (via X or T) in an "Amend" run, the new version is appended to the programs already present, and the new position entered in the Program List. This leaves a dead area in the file, occupied by the old version, which is no longer accessible.

Post-Data Block Format

Word	Contents	Description
* 0	*73000X06	Binary block type
1	1234	Library Number in characters
2	N/O	Number of data entries, as a counter/modifier
3	M	Bucket number of previous post-data block. Zero if none.
4	Destination	3 word entry for data.
5	Bucket number	
6	Checksum (-ve)	
7	Destination	3 word entry for data.
8	Bucket number	
9	Checksum (-ve)	
Repeating 3 word entries		
124	Destination	3 word entry for data.
125	Bucket number	
* 126	Checksum (-ve)	
127	Checksum (-ve)	Negative checksum of words 0-126

* For an “Entry Block”, X=1 and word 126 contains “074 0 /Entry Point”. Otherwise X=0.

Program List

This occupies the first 4 buckets of ICLKEXECFILE. The format is:-

Word	Contents
0	2N = count of N programs: maximum N=255
1	Negative checksum of whole list
2	Library Number of first program
3	Bucket Number of “Entry” Post-Data Block
--	----------
--	----------
2N	Library Number of last program
2N+1	Bucket Number of “Entry” Post-Data Block

A10.3.8 ERROR ACTION

#ENGF acts on errors in one of 3 ways; depending on the gravity, it

DISPLAYS a message and continues
HALTS if the operator can take some remedial action
ABANDONS the run after DISPLAYing the reason

DISPLAY message

When copying from magnetic tape (via M), if a program's format is not as in A10.3.6, #ENGF will leave it out and display:-

1234 OMITTED: MT FORMAT

Giving the library number

HALTED messages

These are used where it s possible to correct the error and continue, or to try repeating some failed transfer. In general there are 3 courses open to the operator:-

(a) correct the fault and repeat
(b) omit the offending item and continue (if applicable)
(c) abandon the run, by "GO #ENGF 29"

The messages are:-

NEEDS TR or CR or LP

#ENGF needs a peripheral. Make one available and "GO #ENGF"

?? Steering Line Record

The last steering line directive read has the wrong format.

Remedy

(a) Correct and reload the rejected line/card, and "GO #ENGF"
(b) "GO #ENGF", and that directive will be omitted
(c) Abandon by "GO #ENGF 29"

For correct formats, see:-

Directive	Section	Directive	Section
A	A10.3.5	G	A10.3.5
B	A10.3.5	M	A10.3.5
C	A10.3.5	O	A10.3.5
D	A10.3.5	T	A10.3.5
E	A10.3.5	X	A10.3.5

And remember that the first directive must be A or O.

++ Steering Line Record

There are already enough steering line directives of this type. Remedy as above.

UNOPENED X: Filename and generation

X is the reply from Executive on failing to open the file named; it has the following values:-

1 File not in system
2 Failure on Integrity Code check
3 Insufficient store to hold further file
6 Another file open when trying to open System File for writing
7 System File is open for writing

Remedy

(a) Check that the file is on-line, or that other programs are not using EDS, as appropriate, and "GO #ENGF."
The "open" PERI will be repeated.
(c) Abandon by "GO ENGF 29"

LIST SUM FAIL

This could occur at the start of an Amend run, after reading the existing Program List.

Remedy

(a) "GO #ENGF", the Program List will be re-read
(c) Abandon by "GO #ENGF 29"

#ENGF

1234 EDS FAULT: SW?

This indicates some checksum or format error in program 1234 being copied from EDS (via C).

Remedy

(a) To repeat reading program 1234, “ON #ENGF 0” and “GO #ENGF”
(b) To omit the program and continue the run, “GO #ENGF”
(c) Abandon by “GO #ENGF 29”

(RELOAD BLOCK/CARD): SW?

Occurs during input of a punched source, if an individual block fails to checksum.

Remedy

(a) To repeat: reload the last paper tape block or card, “ON #ENGF 0” and “GO #ENGF”
(b) To omit: If the fault persists or if he block is genuinely damaged, “GO #ENGF”
(c) Abandon: If there is no point in continuing without the item, “GO #ENGF 29”

(RELOAD PROG): SW?

Occurs on reading the Entry Block of a punched source if the block count therein is not the number of blocks read.

Remedy

As above, reloading the tape/pack from the start if using option (a).

(SORT PROG AND RELOAD) : SW?

This indicates that the punched data block just read should have been packed into an EDS block already written. (More likely with cards than paper tape).

Remedy

As above, putting the offending card in the right place (the last word on each card is a binary sequence number) and reloading the whole pack if repeating.

#ENGF

EDS WRITE FAIL

#ENGF uses "own monitoring" on writing EDS (so that the bootstrap can find things without going through the System Control Area). If 10 attempts fail to write a block, it increases the Bucket number by 1 and tries again. If 9 consecutive Buckets fail, it halts with the above message.

Remedy

(a) "GO #ENGF", to continue on the next Bucket.
(c) Abandon by "GO #ENGF 29". If this occurs after abandoning the run, "DE #ENGF".

B/S WRITE FAIL

This is similar to EDS Write Fail' above, when writing the Bootstrap to ICLKHARDFILE or ICLKE6RMBOOT, without increasing the Bucket number.

Remedy

(a) "GO #ENGF", to repeat writing another 10 times.
(c) Abandon by "DE #ENGF".

B/S WRITE FAIL

This is similar to above, but without increasing the bucket number, when writing the Program List to its fixed position in buckets 1-4.

Remedy

(a) "GO #ENGF", to repeat the transfer.
(c) Abandon by "DE #ENGF".

DISPLAY messages before "RUN ABANDONED"

B MISSING

Occurs immediately after reading the steering line, if no B-directive was present.

Remedy

Correct the steering line and restart at 20 or 21.

G & B/4

Occurs if "G" was present on a steering line together with "B/4".

Remedy

As above.

O/P FILE OBSOLETE FORMAT: NEEDS COPYING

Occurs in an "Amend" run if the output file was produced by #ENGF versions 1-2, which corrupted words 17-22 of every program. Version 3 introduces a change of format and a changed bootstrap to cope with it

Remedy

Create a new ICLKEXECFILE, and do an "Open" run on it, which includes Copying the old one.

LIST FULL

Occurs if the number of programs in the file passes 255. In this case, #ENGF completes the file and goes on to read, check and print it.

Remedy

See what programs, if any, can be weeded out, create another ICLKEXECFILE, and use some D-directives in the steering line.

FILE FULL

Occurs if the file gets full. The run is completed, but omitting the incomplete program.

Remedy

Create a new larger ICLKEXECFILE and copy the old one.

#ENGF

A10.3.9 LISTING OF NEW ICLKEXECFILE

Heading

The heading gives the file generation number, cartridge serial number, date of printing and the number of blocks still free in the form:-

ICLKEXECFILE 987 ON CARTRIDGE 007654 ON 25/12/74 465 BLOCKS LEFT

Contents

There is one line per program, containing:-

Line number
Library Number/Description
Lowest Bucket number
Number of buckets occupied

E.G.:-

99	0056/E6RM FOR ICL ISSUE 7	260	253
100	9604/TEST PROGRAM FOR TR	413	17

#ENGF

A10.3.10 OPERATING INSTRUCTIONS

File Creation

All files written to be #ENGF must be created on the same cartridge, have integrity code 2, Own Monitoring of flaws, and occupy only one File Area each.

ICLKEXECFILE 1-block buckets; ICLKHARDFILE and ICLKE6RMBOOT have 4-block buckets. File sizes are:-

ICLKE6RMBOOT: 8 blocks=2 buckets, starting on a cylinder boundary. (If required)

ICLKEXECFILE: Depends on requirements. As a guide each Executive takes about 150 blocks and room should be left for later additions since the file cannot be extended.

ICLKHARDFILE: See the instructions for the program (#SS33 at the date of writing) which puts the PF56 bootstraps on the disc and the DCP into ICLKHARDFILE, and which must be run before the first run of #ENGF.

Entry Points

20 Normal entry, parameters on paper tape
21 Normal entry, parameters on cards
22 Listing only of required generation, see 9.4
29 Abandon the run, see 9.5

Produce a Library in ICLKEXECFILE

(a) Load #ENGF
(b) Load the steering line on TR or CR and GO#ENGF 20 or 21.
(c) #ENGF displays the 12-character name from its own Request Block.
(d) If punched data is to be input (via X or T), watch for the console message

```
#ENGF; DISPLAY:- GET 1234
```

Load the appropriate data in TR or CR and allocate; each such message is output in steering line order as soon as the reader is free of the previous data.

#ENGF

Normal Output Messages

As a guide to which files are currently in use, #ENGF displays 2 characters whenever it opens or closes a file, giving the unit number preceded by ”O” (open) or “C” (closed). The unit numbers are:-

0 Output ICLKEXECFILE
1 Output ICLKHARDFILE (if “G” absent)
2 Input ICLKEXECFILE
3 Input SYLE (if 2 is on EDS 8)
4 Output ICLKE6RMBOOT (if “G” present)

The normal sequence is either

O0, O1, C1, (others if copying), C0 or
O0, O4, C4, (others if copying), C0

The sequence for GO 22 is merely

O0, C0

The program ends with:-

```
#ENGF; HALTED:- END OF RUN
```

If a second listing is required, GO#ENGF22 and se below

To List an ICLKEXECFILE

(a) Load #ENGF if necessary
(b) GO #ENGF 22
(c) #ENGF displays its name as above, then types

```
#ENGF; HALTED:- ALTER 0 TO GEN
```

At this point, if #ENGF is still in core after completing a run, X0 contains the Generation Number of the file last printed; if just loaded X0 contains the sign bit only, signifying “latest generation”.

If required, type e.g.

AL #ENGF 0 93 (for generation 93)
then

GO #ENGF

#ENGF

Normal Output Messages

#ENGF gives a sequence of 2 displays

O0, C0

Followed by

```
#ENGF; HALTED:- END OF RUN
```

To Abandon the run

Unless one of the WRITE FAIL messages has been typed (see above)

```
GO #ENGF 29
```

To terminate the run.

After typing

```
#ENGF; DISPLAY:- RUN ABANDONED
```

Action depends on what stage has been reached so far. If it is in the writing stage, it completes a usable file efore continuing. If it has not yet written to the file or is already in the printing stage, it releases all peripherals and ends

```
#ENGF; HALTED:- END OF RUN
```

After a persistent WRITE FAIL message, “GO 29” is ineffective since it will only try to write again, so the only course left is “DE #ENGF”.

Summary of HALTED error messages

MESSAGE: #ENGF; HALTED:-
NEEDS XX (XX=TR, CR or LP)
?? Steering item
++ Steering item
UNOPENED X: Name of File, Gen
EDS WRITE FAIL
B/S WRITE FAIL
LIST WRITE FAIL
LIST SUM FAIL
1234 EDS FAULT:SW?
(RELOAD BLOCK/CARD):SW?
(SORT PROG & RELOAD):SW?
(RELOAD PROG):SW?

A10.3.11 BOOTSTRAPPING FROM ICLKEXECFILE

Operating Instructions

This is a brief summary of operating instructions. For full details see the appropriate technical manuals.

(a) Ensure the DCP is loaded in the PF56 and that the PF56 hand keys are set to the right mechanism number (zero if the cartridge is shared with George Executives).
(b) Use normal "Freeze and Crest" procedure for the PF56 channel.
(c) If asked

```
EXEC FILE?
```

reply

```
ICLKE6RMBOOT
```

(d) When asked

```
LIB.NO./HOW MANY K?
```

reply e.g.

```
0056/128
```

(i.e. Library Number, solidus, store size)

(e) The bootstrap will then load the required program, type its description, wind the program down and enter it.

Error Action

This consists of typing a message, and returning to step (d) above.

NOT FOUND

The CSN has not been found, after trying all mechanisms 0-8.

CSUM LIST

The Program List checksum failed.

NOT FOUND:1234

The Library Number 1234 is not present in the List.

CSUM:1234

A data or post-data block failed to checksum.

FAILED

Either a transfer has failed or the DCP is not working properly.

INOP

The mechanism has gone inoperable.

Post Mortem

In cases of extreme difficulty, it may be useful to take a Post Mortem off the store. An entry point has been provided for moving the lower store to the appropriate area, as follows:-

Bootstrap Type	Manual Jump To	Loop Stop in
1904	*170	*175
1906, 4E(single)	*4004	*4012
1906A	*4006	*4014
Dual	*4004	*4012

The PM can then be taken in the usual way, and may be terminated when the store address reaches 5 octal digits.

A10.4 #EXD1 – Copy EWG3 to Disc

A10.4.1 TITLE

To copy Executive EWG3 to disc.

A10.4.2 HARDWARE REQUIREMENTS

5120 words of core store
1 paper tape reader and/or 1 card reader
1 line printer
1 EDS tranasport
1 MT transports

A10.4.3 USE OF PERIPHERALS

TR0 and/or CR0	Parameters	The readers are allocated and released as required
LP0	Report	The printer is allocated and released as required
DA0	File to be updated	This file is opened and closed as required
MT0	Program tape from which the executive is to be loaded	The specified tape is opened and closed as required

A10.4.4 DESCRIPTION

#EXD1 is a free-standing re-enterable program which will copy EWG3 from tape to disc, it does not install a bootstrap.

A10.4.5 METHOD EMPLOYED

The tape, executive subfile and disc file names are specified by parameters, the files are opened and the specified executive copied

A10.4.6 PROGRAM SWITCHES

None used.

A10.4.7 OPERATING INSTRUCTIONS

1. Ensure required disc and tape are loaded

2. Load/Find #EXD1 — HALT:- LD

3. To start #EXD1, either:-

   a) Read parameters from paper tape: — GO #EXD1 20
   or b) Read parameters from cards: — GO #EXD1 21
   or c) Read parameters from console: — GO #EXD1 22

4. At the end of run — HALT:- HH

A10.4.8 EXCEPTION CONDITIONS

Typewriter Messages

Message	Action	Action
HALT -	PARAMETER TYPE NOT KNOWN	GO29 to abandon
	PARAMETER ERROR - TOO MANY CHARS IN HERE	
	TOO MANY PARAMETERS	
	PARAMETER ERROR - WRONG CHARACTER	
	FILE NOT FOUND	
	MT ERROR DETECTED	
	NOT AN EXEC FILE	
	MAG TAPE CORRUPT	
	FORTY CHARACTERS OF WRONG PARAMETER HERE	
	DA ERROR	
	RUN ABANDONED	