

SHARE PROGRAM LIBRARY AGENCY



PROGRAM NUMBER

04003

University of Miami

1365 MEMORIAL DRIVE - CORAL GABLES, FLORIDA
(305) - 284-6257

360 Operating System Accounting Routine

by

Donald Jacobs

Direct Inquiries To: Donald Jacobs

Technical Assistance
Not Currently Available

DISCLAIMER

Triangle Universities Computation Center (TUCC) serves solely as the distribution agent for contributed programs and does not test or maintain them. They are distributed essentially in the original form submitted by the author. Neither TUCC nor SHARE, INC., makes any warranty, expressed or implied, as to the documentation, function, or performance of the contributed programs.

Abstract (Cards 10-99, Columns 12-72)

Purpose:

- (1) Determine task time of each step of Job.
- (2) Determine Wait/Overhead Time of Job.
- (3) Inform operator of tape assignments by DD Name.
- (4) Determine starting address of problem program RB.
- (5) Determine maximum tapes and disks used in Job.
- (6) Print accounting information of Job on sysout and system residence pack.
- (7) Punch Accounting Records from Disk.

Method:

Step Initiator has been modified so that it accomplishes (3), (4), and (5) above. It also issues a timer Macro prior to issuing the "XCTL" to the problem program. Step termination accomplishes (1) and (2) above. Job termination accomplishes (7) above.

Write-Up:

There are 7 routines involved. Each is written up individually.

(Please attach additional pages, if necessary)

Pages Attached: Keypunchable Abstract Continuation (AC) 2Non-Keypunchable Supplement (NK) 16

Signature of Submitter

*Donald Jacobs*Date 12/13/66

Signature of Installation Addressee

Donald Jacobs

SHARE PROGRAM LIBRARY

(AT) ATTACHMENT TO PROGRAM PACKAGE SUBMITTAL

Please use this form for aspects of the program, such as special symbols, charts, etc., that cannot properly be described in the keypunchable abstract. May also be used for continuation of the keypunchable abstract. Check which.

INST. CODE, PROGRAM NUMBER/DESIGNATION, SUFFIX:		To be filled in by SDA:	
V	D B	A C T	S D A
DATE SUBMITTED 11/23/66		DATE DISTRIBUTED	

☒ (NK) Non-Keypunchable Supplement Page 1 of 16
☐ (AC) Abstract Continuation, (Keypunchable) Page of

IEFSD004 (Revised)

Purpose:

- (1) Call "TSKTMSUB"
- (2) XCTL to the problem program

Implementation:

The following lines have been added:

```
STM 14,1,HOLD      08360100
CALL TSKTMSUB      08360200
LM 14,1,HOLD        08360300
HOLD DS 4F          10240100
```

The source and object modules included are for Release 6. Before utilizing with any future releases, it should be determined if the module has changed, and, if so, applying the above changes to the new module.

SHARE PROGRAM LIBRARY

(AT) ATTACHMENT TO PROGRAM PACKAGE SUBMITTAL

Please use this form for aspects of the program, such as special symbols, charts, etc., that cannot properly be described in the keypunchable abstract. May also be used for continuation of the keypunchable abstract. Check which.

INST. CODE, PROGRAM NUMBER/DESIGNATION, SUFFIX:		To be filled in by SDA:	
V	D B	A C T	S D A
DATE SUBMITTED 11/21/66		DATE DISTRIBUTED	

☒ (NK) Non-Keypunchable Supplement Page 2 of 16
☐ (AC) Abstract Continuation, (Keypunchable) Page of

TSKTMSUB

Purpose:

- (1) Issues timer macro.
- (2) Determines maximum tape and disk drives used by job.
- (3) Issues allocation messages to operator informing him of drive assignments.
- (4) Determines starting address of problem program RB.

Method:

- (1) A change was made in IEFSD004 to branch to this subroutine prior to issuing the "XCTL" to the problem program.
- (2) This subroutine accomplishes the above and stores items (1), (2), and (4) in the resident part of core (refer to IGC255).

Implementation:

The object modules, "TSKTMSUB" and "IEFSD004" have to be linked into IEFXA. The following cards should be used in the linked step:

Include SYSLMOD(IEFXA)
Entry IEFXA
Name IEFXA(R)

Please use this form for aspects of the program, such as special symbols, charts, etc., that cannot properly be described in the keypunchable abstract. May also be used for continuation of the keypunchable abstract. Check which.

INST. CODE, PROGRAM NUMBER/DESIGNATION, SUFFIX:		To be filled in by SDA:	
V	D	B	
A	C	T	
DATE SUBMITTED 11/21/66		DATE DISTRIBUTED	
		SDA NO. S D A	

☒ (NK) Non-Keypunchable Supplement Page 3 of 16
☐ (AC) Abstract Continuation, (Keypunchable) Page of

IGC255

Purpose:

This "Type 1 SVC Routine" allows the accounting routine that is called in at step termination to store accounting information in a pre-assigned table in resident core. The accounting routine that is called in at job termination is thus able to pick this information up and write it out.

Method:

When this subroutine is called, it places the address of the accounting table in general register 1 and then returns to the caller.

Implementation:

Refer to the 360 SRL C28-6554, pages 46-47 for particulars on including SVC routines at system generation. The operand which must appear in the SVC TABLE macro is as follows:

SYC-255-T1-S0

It is recommended that the load module be placed in SYSL.LINKLIB of DLIB01 under the member name IGC255.

Please use this form for aspects of the program, such as special symbols, charts, etc., that cannot properly be described in the keypunchable abstract. May also be used for continuation of the keypunchable abstract. Check which.

INST. CODE, PROGRAM NUMBER/DESIGNATION, SUFFIX:		To be filled in by SDA:	
V	D	B	
A	C	T	
DATE SUBMITTED 11/21/66		DATE DISTRIBUTED	
		SDA NO. S D A	

☒ (NK) Non-Keypunchable Supplement Page 4 of 16
☐ (AC) Abstract Continuation, (Keypunchable) Page of

IEFACTRT (Step Termination)

PURPOSE:

This accounting routine is called in at step termination and serves the function of recording accounting information of the step. Specifically it records:

1. Start Time
2. Stop Time
3. Assembly Time
4. FORTRAN Time
5. COBOL Time
6. Linked Time
7. Execution Time
8. Other Time (Utility Programs, Sort, etc.)
9. Wait Time
10. Date
11. Programmer
12. Account Number
13. Rework Code
14. Problem Program RB Address
15. Task Time

METHOD:

When this subroutine receives control, it issues an SVC 255 macro (refer to IGC255) to receive the address in resident core of the accounting table. It then issues a timer macro to pickup the time (in binary) and the date (in Julian days). The date, programmer, account number and rework code is then stored in the table if this is the first step. Step time minus wait/overhead time is then computed and added to one of the words of the table (dependent upon what program was run). Wait/overhead time is also computed and added to wait/overhead time. The routine then exits.

IMPLEMENTATION:

Refer to the 360 SRL C28-6550, pages 45-48, for particulars on including an accounting routine into your system. In general, if this routine is being included before system generation it should be linked into SYSL.MDLIB of DLIB01 under the member name of GO. The following cards should come after the step termination object module

Cont'd.

320-1927-0 U/M 050

Printed in U.S.A.

SHARE PROGRAM LIBRARY

(AT) ATTACHMENT TO PROGRAM PACKAGE SUBMITTAL

Please use this form for aspects of the program, such as special symbols, charts, etc., that cannot properly be described in the keypunchable abstract. May also be used for continuation of the keypunchable abstract. Check which.

INST. CODE, PROGRAM NUMBER/DESIGNATION, SUFFIX:

V	D	B	A	C	T						
---	---	---	---	---	---	--	--	--	--	--	--

 To be filled in by SDA: SDA NO.

S	D	A							
---	---	---	--	--	--	--	--	--	--

 DATE SUBMITTED 11/21/66 DATE DISTRIBUTED

If routine is being added after generation: ☒ (NK) Non-Keypunchable Supplement ☐ (AC) Abstract Continuation, (Keypunchable) Page 5 of 16

Include SYSLMOD(GO)
ALIAS IEFW21SD,IEFYN
ENTRY IEFSDOLL
NAME GO(R)
/*

RESTRICTIONS:
All utility routines written by the user should be named having IE as the first two letters. This will then allow the step termination accounting routine to record the time under other time and not under execution time.
Wait/overhead time records the time the job has been in the wait state due to operator intervention (setup time only) as well as the overhead required to load in each step.
Storage Protect cannot be generated if this routine is to interface with IGC255 as it stands.

FORMAT OF JOB CARD:
The following format should be used on the job card:
//jn b JOB b (an,rc), 'pn', MSGLEVEL = 1
where:
jn = Job Name (max. of 8 characters, first of which must be a letter)
b = at least one blank space
an = account number (5 characters)
rc = rework code (6 digits) of following format:
"pttTW"
where:
p = no. of printers used
tt = no. of tapes used } ignored
T = Type code having following possibilities:

SHARE PROGRAM LIBRARY

(AT) ATTACHMENT TO PROGRAM PACKAGE SUBMITTAL

Please use this form for aspects of the program, such as special symbols, charts, etc., that cannot properly be described in the keypunchable abstract. May also be used for continuation of the keypunchable abstract. Check which.

INST. CODE, PROGRAM NUMBER/DESIGNATION, SUFFIX:

V	D	B	A	C	T						
---	---	---	---	---	---	--	--	--	--	--	--

 To be filled in by SDA: SDA NO.

S	D	A							
---	---	---	--	--	--	--	--	--	--

 DATE SUBMITTED 11/21/66 DATE DISTRIBUTED

☒ (NK) Non-Keypunchable Supplement ☐ (AC) Abstract Continuation, (Keypunchable) Page 6 of 16

- 0 = production
- 1 = test
- 2 = assembly/compile
- 3 = parallel
- 4 = support for code 1, 2, or 3
- 5 = preventative maintenance
- 6 = down and/or repair
- 7 = miscellaneous
- 8 = demonstration or instruction
- 9 = outside use
- WW = work code having following possibilities:
 - 00 - non rework
 - 10 - customer
 - 20 - analyst
 - 30 - standards
 - 4x - operations
 - 41 - control
 - 42 - coord. and scheduling
 - 43 - keypunch
 - 44 - EAM
 - 45 - EDP Operations
 - 46 - Librarian
 - 5x - facilities

Please use this form for aspects of the program, such as special symbols, charts, etc., that cannot properly be described in the keypunchable abstract. May also be used for continuation of the keypunchable abstract. Check which.

INST. CODE	PROGRAM NUMBER/DESIGNATION	SUFFIX	To be filled in by SDA:									
V D B	A C T		SDA NO. S D A									
DATE SUBMITTED 11/21/66			DATE DISTRIBUTED									

☒ (NK) Non-Keypunchable Supplement Page 7 of 66
☐ (AC) Abstract Continuation, (Keypunchable) Page of

52 - Power
53 - Cable
54 - Other

6x - Machine

61 - Printer
62 - Card Read Punch
63 - Typewriter
64 - Tape Unit
65 - Tape
66 - Other than 1401

90 - Diagnostic

- (1) NOTE: Contents of rc is obviously at discretion of user and is merely presented here as an example of how it is used at Boeing-Vertol.

Please use this form for aspects of the program, such as special symbols, charts, etc., that cannot properly be described in the keypunchable abstract. May also be used for continuation of the keypunchable abstract. Check which.

INST. CODE	PROGRAM NUMBER/DESIGNATION	SUFFIX	To be filled in by SDA:									
V D B	A C T		SDA NO. S D A									
DATE SUBMITTED 11/21/66			DATE DISTRIBUTED									

☒ (NK) Non-Keypunchable Supplement Page 8 of 16
☐ (AC) Abstract Continuation, (Keypunchable) Page of

IEFACTRT (Job Termination)

PURPOSE:

This accounting routine is called in at job termination and serves the function of writing accounting information of the job on SYSOUT and on SYS1.ACCREC (an existing data set residing on the system residence pack).

METHOD:

When this subroutine receives control, the following functions must be performed:

1. An SVC 255 macro issued (refer to IGC255) to receive the address in resident core of the accounting table.
2. A check to see if this is the first job after IPL. If it is, it zeros out wait time, issues a timer macro and records the time in "stop time" and then exits. Beside returning to the calling routine the exit code transfers "stop time" to "start time".

SHARE PROGRAM LIBRARY

(AT) ATTACHMENT TO PROGRAM PACKAGE SUBMITTAL

Please use this form for aspects of the program, such as special symbols, charts, etc., that cannot properly be described in the keypunchable abstract. May also be used for continuation of the keypunchable abstract. Check which.

INST. CODE, PROGRAM NUMBER/DESIGNATION, SUFFIX:									
V	D	B	A	C	T		S	D	A
DATE SUBMITTED						DATE DISTRIBUTED			
11/21/66									

☒ (NK) Non-Keypunchable Supplement Page 9 of 16
☐ (AC) Abstract Continuation, (Keypunchable) Page of

3. If this job is not the first after IPL, editing of the accounting information is performed. Included on this is the picking up of all times, converting it to thousandths of an hour, zeroing out wait time, converting the Julian date to "MM/DD/YY", etc. Each print line is moved to the sysout buffer as the line becomes ready. This moving is accomplished according to the coding described in the 360 SRL C28-6550, page 47. No writing on the disk is performed until all editing has been completed and the appropriate record generated.

4. SYS1.ACCREC is then written on. This is accomplished by doing the following:

- The address of the CVT (for format refer to the 360 SRL Z28-6605, pages 27-30) is found from location 1610. From this the address of the start of the UCB pointers is picked up. With this address the pointer to the printer and the system residence is found and stored in the appropriate places in a new TIOT (for format refer to the 360 SRL Z28-6605, pages 98-99) already set up in the sub-routine.
- The address of the TIOT in the TCB (for format refer to the 360 SRL Z28-6605, pages 95-97) is next saved and the pointer changed to point to the new TIOT.
- An Open-J macro is then issued. The address of the JFCB (for format refer to the 360 SRL C28-6550, pages 124-129) is picked up from the exit list of the DCB being opened. This JFCB has been completely set up in the subroutine so that the control program will not attempt to write it back on the disk.
- The record is then written on SYS1.ACCREC. Since the data set was opened with "DISP=MOD" in the JFCB, the record is added to the end of the existing data set.

SHARE PROGRAM LIBRARY

(AT) ATTACHMENT TO PROGRAM PACKAGE SUBMITTAL

Please use this form for aspects of the program, such as special symbols, charts, etc., that cannot properly be described in the keypunchable abstract. May also be used for continuation of the keypunchable abstract. Check which.

INST. CODE, PROGRAM NUMBER/DESIGNATION, SUFFIX:									
V	D	B	A	C	T		S	D	A
DATE SUBMITTED						DATE DISTRIBUTED			
11/21/66									

☒ (NK) Non-Keypunchable Supplement Page 10 of 16
☐ (AC) Abstract Continuation, (Keypunchable) Page of

e. Finally, the data set is closed and the address of the old TIOT restored.

5. A branch to the exit coding is the last thing done.

ERROR CONDITIONS

If the system residence cannot be located, the operator is notified via the operator's console. If an I/O error results upon writing of the record on SYS1.ACCREC this record is written out on the operator's console.

IMPLEMENTATION

This routine should be linked into SYS1.LINKLIB of the generated system under the member name of IEFZA. The following cards should come after the job termination accounting module:

```
INCLUDE SYSIMOD(IEFZA)
ENTRY IEFZA
NAME IEFZA(R)
/*
```

RESTRICTIONS

This routine was assembled assuming a "360-Mod 50". As such, SYSOUT comes out with "Model = 50" and Syspunch with "LC" (indicating Mod 50). To change this without re-assembling change card 26 (col's. 17-18) and card 33 (col's. 64-65) of the object module. Finally, it is at the user's discretion to punch the accounting cards (refer to IEFACDS) whenever he desires. The only limitation he should be aware of is that volume switching is not provided here and thus errors will ensue when the system residence becomes filled.

Storage protect cannot be generated if this routine is to interface with IGC255 as it stands.

SHARE PROGRAM LIBRARY

(AT) ATTACHMENT TO PROGRAM PACKAGE SUBMITTAL

Please use this form for aspects of the program, such as special symbols, charts, etc., that cannot properly be described in the keypunchable abstract. May also be used for continuation of the keypunchable abstract. Check which.

INST. CODE	PROGRAM NUMBER/DESIGNATION, SUFFIX:	To be filled in by SDA:	
V D B	A C T	SDA NO.	S D A
DATE SUBMITTED	11/21/66	DATE DISTRIBUTED	

☒ (NK) Non-Keypunchable Supplement Page 11 of 16
☐ (AC) Abstract Continuation, (Keypunchable) Page ____ of ____

FORMAT OF RECORD WRITTEN ON SYSLACREC
FROM IEFACTRT (JOB TERMINATION)CONTENTS

BYTES	CONTENTS
1-9	Programmer's Name
10-10	Blank
11-15	Charge Number
16-18	Blank
19-24	Date (MMDDYY)
25-30	Start Time in Thousandths of Hours (OXX,XXX)
31-36	Stop Time (OXX,XX)
37-42	Assembly Time (OXX,XXX)
43-48	FORTAN Time (OXX,XXX)
49-54	COBOL Time (OXX,XXX)
55-60	LINKEDIT Time (OXX,XXX)
61-66	Execution Time (OXX,XXX)
67-72	Rework Code
73-78	Other Time (OXX,XXX)
79-80	IC (I=Card 1, C=Mod 50)
81-86	Wait Time (OXX,XXX)

SHARE PROGRAM LIBRARY

(AT) ATTACHMENT TO PROGRAM PACKAGE SUBMITTAL

Please use this form for aspects of the program, such as special symbols, charts, etc., that cannot properly be described in the keypunchable abstract. May also be used for continuation of the keypunchable abstract. Check which.

INST. CODE	PROGRAM NUMBER/DESIGNATION, SUFFIX:	To be filled in by SDA:	
V D B	A C T	SDA NO.	S D A
DATE SUBMITTED	11/21/66	DATE DISTRIBUTED	

☒ (NK) Non-Keypunchable Supplement Page 12 of 16
☐ (AC) Abstract Continuation, (Keypunchable) Page ____ of ____

FORMAT OF ACCOUNTING CARDS PUNCHED

CARD 1

BYTES	CONTENTS
1-9	Programmer's Name
10-10	Blank
11-15	Charge Number
16-18	Blank
19-24	Date (MMDDYY)
25-30	Start Time in Thousandths of Hours (OXX,XXX)
31-36	Stop Time (OXX,XXX)
37-42	Assembly Time (OXX,XXX)
43-48	FORTAN Time (OXX,XXX)
49-54	COBOL Time (OXX,XXX)
55-60	LINKEDIT Time (OXX,XXX)
61-66	Execution Time (OXX,XXX)
67-72	Rework Code
73-78	Other Time (OXX,XXX)
79-80	IC (I=Card 1, C=Model 50)

SHARE PROGRAM LIBRARY

(AT) ATTACHMENT TO PROGRAM PACKAGE SUBMITTAL

Please use this form for aspects of the program, such as special symbols, charts, etc., that cannot properly be described in the keypunchable abstract. May also be used for continuation of the keypunchable abstract. Check which.

INST. CODE, PROGRAM NUMBER/DESIGNATION, SUFFIX		To be filled in by SDA:	
V	D	R	SDA NO. S D A
A	C	T	DATE DISTRIBUTED
DATE SUBMITTED 11/21/66			

☒ (NK) Non-Keypunchable Supplement Page 13 of 16
☐ (AC) Abstract Continuation, (Keypunchable) Page ____ of ____

CARD 2

BYTES

1-9

10-10

11-15

16-18

19-24

25-30

31-36

37-42

43-78

79-80

CONTENTS

Programmer's Name

Blank

Charge Number

Blank

Date (MMDDYY)

Start Time in Thousandths of Hours (OXX,XXX)

Stop Time (OXX,XXX)

Wait Time (OXX,XXX)

Blank

2C (2=Card, C=Model 50)

PROGRAMMER XXXXXXXX
 JOB BEGAN AT XX.XXX
 MODEL = 50

CHARGE NUMBER XXXX
 JOB ENDED AT XX.XXX

ASSEMBLY TIME XX.XXX
 FORTRAN TIME XX.XXX
 FORTRAN TIME XX.XXX
 LINKEDIT TIME XX.XXX
 EXECUTION TIME XX.XXX
 OTHER TIME XX.XXX

DATE XX/XX/XX
 REWORK PAGE XXXX
 PD RB ADD XXXX

Please use this form for aspects of the program, such as special symbols, charts, etc., that cannot properly be described in the keypunchable abstract. May also be used for continuation of the keypunchable abstract. Check which:

INST. CODE	PROGRAM NUMBER/DESIGNATION	SUFFIX	To be filled in by SDA:			
V D B	A C T		SDA NO.	S	D	A
DATE SUBMITTED 11/21/66			DATE DISTRIBUTED			

☒ (NK) Non-Keypunchable Supplement Page 14 of 16
☐ (AC) Abstract Continuation (Keypunchable) Page of

IEFPACDS

Purpose:

This routine punches cards which it reads from SYS1.ACCREC. The records on this data set were put there by the job terminating accounting routine (refer to IEFACRT - job termination). After punching all records, it closes the data set such that only the first record exists (considered strictly a dummy record).

Method:

When the routine receives control, it opens SYS1.ACCREC, reads one record, issues a note macro, reads all succeeding records punching the first 80 bytes into punch staker 1 and the last 6 bytes (wait/overhead time) plus the first 36 bytes into punch staker 2 (provided wait/overhead time is not zero), issues a point macro, and then closes.

Implementation:

This routine should be linked into SYS1.LINKLIB. To execute, the following DD cards should be used to point to SYS1.ACCREC and the punch:

```
//SYSUT1 DD DSN=SYS1.ACCREC, DISP=OLD, C
//
//UNIT=2311, VOLUME=REF=SYS1.SVCLIB
//SYSPUNCH DD UNIT=2540-2
```

Please use this form for aspects of the program, such as special symbols, charts, etc., that cannot properly be described in the keypunchable abstract. May also be used for continuation of the keypunchable abstract. Check which:

INST. CODE	PROGRAM NUMBER/DESIGNATION	SUFFIX	To be filled in by SDA:			
V D B	A C T		SDA NO.	S	D	A
DATE SUBMITTED 11/21/66			DATE DISTRIBUTED			

☒ (NK) Non-Keypunchable Supplement Page 15 of 16
☐ (AC) Abstract Continuation (Keypunchable) Page of

Restrictions:

This routine was assembled assuming a "360-Mod 50". As such, SYSPUNCH comes out with "2C" - the wait card only (indicating Mod 50). To change this without re-assembling, change card 10 (col's. 53-54) of the object module.

SHARE PROGRAM LIBRARY

(A1) ATTACHMENT TO PROGRAM PACKAGE SUBMITTAL

Please use this form for aspects of the program, such as special symbols, charts, etc., that cannot properly be described in the keypunchable abstract. May also be used for continuation of the keypunchable abstract. Check which.

INST. CODE, PROGRAM NUMBER/DENOMINATION, SUFFIX										To be filled in by SDA:										
V	D	B	A			C	T				SDA NO. S D A									
DATE SUBMITTED 11/21/66										DATE DISTRIBUTED										

☒ (NF) Non-Keypunchable Supplement Page 16 of 16
☐ (AC) Abstract Continuation, (Keypunchable) Page _____ of _____

INITIAL

Purpose:

This routine creates a data set on the system residence pack entitled SYS1.ACCREC and then writes one record on it. It does this so that another routine, which reads this data is able to issue a note macro after the first read, and then later point back to it and close at this point. This one record has no other significance but is merely a dummy record.

Method:

SYS1.ACCREC is created via a DD card supplied by the user. When the routine receives control, it opens the data set, writes one record, and then closes.

Implementation:

It is recommended that the load module be linked into a temporary load module and executed in the next job step. The following DD card should be used to create SYS1.ACCREC:

```
//SYSUT1 DD DSN=SYS1.ACCREC,DISP=(KEEP),          C
//          UNIT=2311,VOLUME=REF=SYS1.SVCLIB,        C
//          SPACE=(TRK,(30,10)RISE)
```