

# The Job Goes Ever On and On...

By *Graham L. Cromar*

*From a programmer's perspective few situations are more irritating than trying to implement code or apply a fix after-hours, only to discover that there are still users locking needed objects. Invariably, you know you'll discover that many or all of them are not actually at the controls! This usually means that you have to waste a lot of time calling people to find out they aren't really there. Worse, their job may be holding up important maintenance functions, not to mention the huge security risk this kind of behavior represents.*

*If this concerns you (and it should), chances are you already have a shrink-wrapped utility for ending inactive user jobs. If you don't, and that decision was based on the high cost of attaining this functionality when bundled into other packages you may not need, this article may be of interest to you. Here is a basic, "no frills" utility that's smart enough to know which users get the heave ho and which jobs to leave alone.*



*Graham Cromar*

## Solution in Brief

The general approach taken in this subsystem is to capture the Work Active Job (WRKACTJOB) statistics to a physical file. Each time the program is run it compares a snap-shot of the jobs running at that time, with statistics collected at the end of the previous run. It determines which jobs have had no activity during this interval and issues an End Job (ENDJOB) command for these jobs. You can view the WRKACTJOB statistics online by typing WRKACTJOB at the command line, hitting the <enter> key then using function key F11 to view statistics.

There are two fields in the Work Active Job statistics on which this comparison is based. One is the "INT" statistic which tracks keyboard activity. The other is the "AUX/IO" statistic which tracks things like file reads and writes associated with a running, interactive program. If both these statistics have not changed in the interval between calls, the program 'knows' that job is inactive. In practice, you would call this routine once every hour (for example) and jobs that had been inactive for the previous hour would be terminated. There is a file in which to record user names or job names to be exempted from this process. A maintenance program for this file is not provided here (you can write your own or use another utility like DBU).

## The Code

### SYSSTATC

```
***** Beginning of data*****
0000.01 /* Capture active jobs to SYSSTAT0 */
0000.02
0001.00          PGM
0002.00
0002.01          DCL          VAR(&JOB) TYPE(*CHAR) LEN(10)
0002.02          DCL          VAR(&USER) TYPE(*CHAR) LEN(10)
0002.03          DCL          VAR(&NBR) TYPE(*CHAR) LEN(6)
0002.04          DCL          VAR(&RESET) TYPE(*CHAR) LEN(1)
0002.05
0003.00 /* Check for temporary file SYSSTAT and clear */
0004.00
0005.00          CHKOBJ        OBJ(QTEMP/SYSSTAT) OBJTYPE(*FILE) MBR(*FIRST)
0006.00          MONMSG        MSGID(CPF0000) EXEC(CRTPF +
0007.00                      FILE(QTEMP/SYSSTAT) RCDLEN(192) +
0008.00                      OPTION(*NOSRC *NOLIST) SIZE(*NOMAX) +
0008.01                      LVLCHK(*NO))
0009.00
0010.00          CLRPFM        FILE(SYSSTAT)
0013.00
0013.01 /* Check for holding file SYSSTAT0 and clear */
0013.02
0014.00          CHKOBJ        OBJ(SYSSTAT0) OBJTYPE(*FILE) MBR(*FIRST)
0015.00          MONMSG        MSGID(CPF0000) EXEC(CRTPF +
0016.00                      FILE(QGPL/SYSSTAT0) +
0017.00                      SRCFILE(QGPL/QDSSRC) OPTION(*NOSRC +
0017.01                      *NOLIST) SIZE(*NOMAX))
0018.00
0019.00          CLRPFM        FILE(SYSSTAT0)
0020.00
0020.01 /* Check for past holding file SYSSTAT1 */
0020.02
0020.03          CHKOBJ        OBJ(SYSSTAT1) OBJTYPE(*FILE) MBR(*FIRST)
0020.04          MONMSG        MSGID(CPF0000) EXEC(CRTPF +
0020.05                      FILE(QGPL/SYSSTAT1) +
0020.06                      SRCFILE(QGPL/QDSSRC) +
0020.07                      SRCMBR(SYSSTAT0) OPTION(*NOSRC *NOLIST) +
0020.08                      SIZE(*NOMAX))
0020.09
```



```

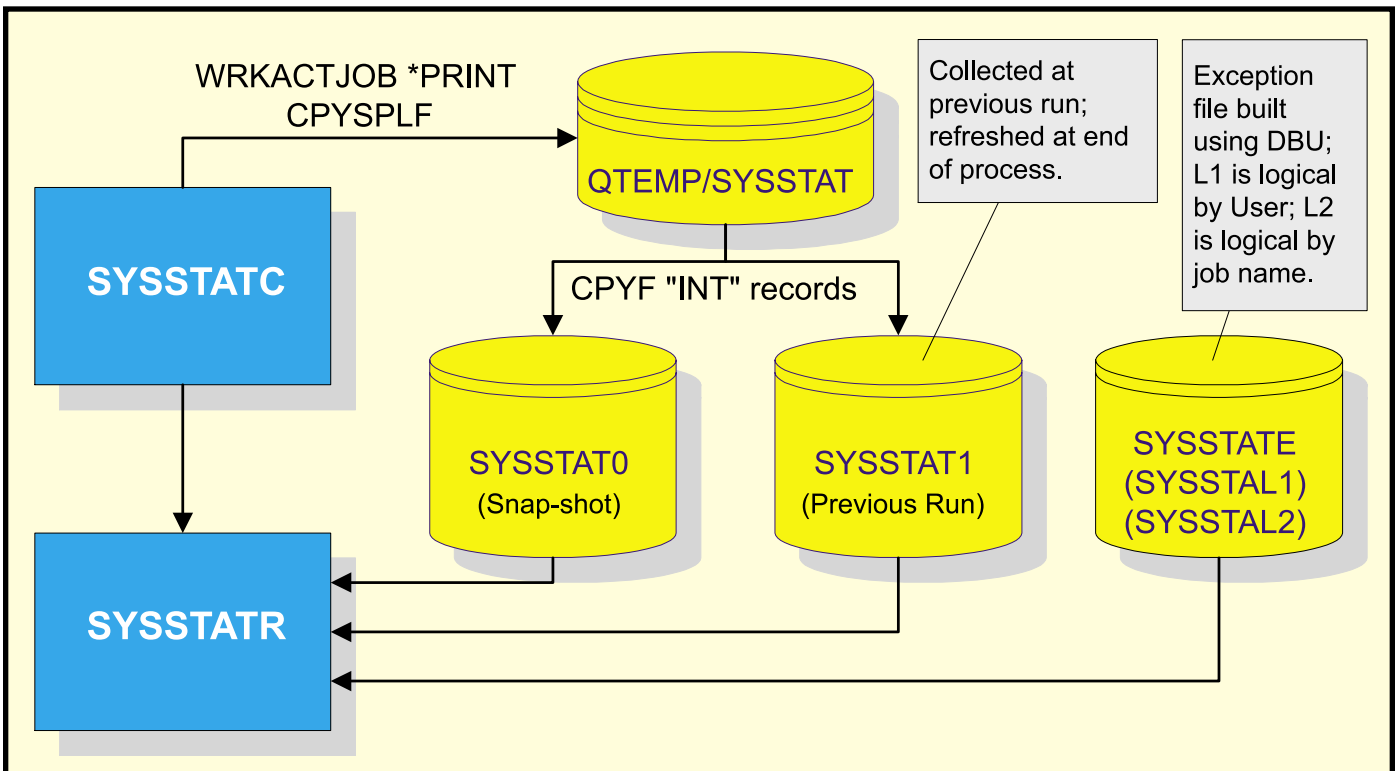
0021.00 /* Send active job data to temporary file */
0023.00
0024.00          WRKACTJOB  OUTPUT(*PRINT) SBS(QINTER)
0024.01          RTVJOBA   JOB(&JOB)  USER(&USER) NBR(&NBR)
0024.02          CPYSPLF   FILE(QPDSPAJB) TOFILE(QTEMP/SYSSTAT) +
0024.03                      JOB(&NBR/&USER/&JOB) SPLNBR(*LAST)
0024.04          DLTSPLF   FILE(QPDSPAJB) JOB(&NBR/&USER/&JOB) +
0024.05                      SPLNBR(*LAST)
0024.06          MONMSG     MSGID(CPF0000)
0026.00
0026.01 /* Send captured data to dds file */
0026.02
0027.00          CPYF        FROMFILE(QTEMP/SYSSTAT) TOFILE(SYSSTAT0) +
0027.01                      MBROPT(*REPLACE) FROMRCD(1) +
0027.02                      INCCHAR(SYSSTAT 38 *EQ 'INT') FMTOPT(*NOCHK)
0028.00          MONMSG     MSGID(CPF0000)
0030.00
0031.00 /* Compare activity - end inactive jobs */
0032.00
0034.00          CALL        PGM(SYSSTATR)
0034.01
0034.02 /* Collect reset data to SYSSTAT1 (in case stats maxed to '+++') */
0034.03
0034.07          CLRPFM     FILE(SYSSTAT)
0034.08          CLRPFM     FILE(SYSSTAT1)
0034.09          WRKACTJOB  OUTPUT(*PRINT) RESET(*YES) SBS(QINTER)
0034.10          CPYSPLF   FILE(QPDSPAJB) TOFILE(QTEMP/SYSSTAT) +
0034.11                      JOB(&NBR/&USER/&JOB) SPLNBR(*LAST)
0034.12          DLTSPLF   FILE(QPDSPAJB) JOB(&NBR/&USER/&JOB) +
0034.13                      SPLNBR(*LAST)
0034.14          MONMSG     MSGID(CPF0000)
0034.15          CPYF        FROMFILE(QTEMP/SYSSTAT) TOFILE(SYSSTAT1) +
0034.16                      MBROPT(*REPLACE) FROMRCD(1) +
0034.17                      INCCHAR(SYSSTAT 38 *EQ 'INT') FMTOPT(*NOCHK)
0034.18          MONMSG     MSGID(CPF0000)
0034.19
0034.20
0035.00  SKIP:        ENDPGM
***** End of data *****

```

## The Details

The Work Active Jobs command has a \*print option that allows you to send the output of the command to a spooled file “QPDSPAJB” in the current jobs spool. Using the Retrieve Job Attributes (RTVJOBA) command it is possible to capture the job number, user name, and job name of the current job and construct a Copy Spooled File (CPYSPLF) command to move the spooled report into a flat physical file “SYSSTAT”.

This is of no use unless the “INT” and “AUXIO” statistics for the interactive jobs can be isolated. So, it is necessary to build a data description specification (DDS) for another physical file “SYSSTAT0” and then Copy File (CPYF) the records with type “INT” into that file. These become the snap-shot statistics for the currently running interactive jobs. In order to retain information for comparison with the next run, an identical file “SYSSTAT1” built using the same DDS is used.



**SYSSTAT0**

\*\*\*\*\* Beginning of data \*\*\*\*\*

```

0002.00      R SYSOR
0003.00      A          SFILL1          3A
0004.00      A          SJOBNM          10A
0005.00      A          SFILL2          3A
0006.00      A          SUSER           10A
0006.01      A          SFILL3          2A
0007.00      A          SJOBNR          6A
0008.00      A          SFILL4          3A
0008.01      A          STYPE           3A
0008.02      A          SFILL5          4A
0008.03      A          SPOOLN          2A
0008.04      A          SFILL6          4A
0008.05      A          SPRIOR          2A
0008.06      A          SFILL7          4A
0008.07      A          SCPUVL          6A
0008.08      A          SFILL8          3A
0008.09      A          SINTER          3A
0008.10      A          SFILL9          2A
0008.11      A          SRSPVL          5A
0008.12      A          SFIL10          5A
0008.13      A          SAUXIO          3A
0008.14      A          SFIL11          2A
0008.15      A          SCPUPC          4A
0008.16      A          SFIL12          3A
0008.17      A          SDESCR          15A
0008.18      A          SFIL13          3A
0008.19      A          SSTATU          4A
0008.20      A          SFIL14          6A
0008.21      A          STHRED          3A
0008.22      A          SFIL15          69A
0008.25      A          K SUSER
0052.00      A          K SJOBNM
0054.00      A          K SJOBNR

```



\*\*\*\*\* End of data \*\*\*\*\*

**SYSSTATE**

\*\*\*\*\* Beginning of data \*\*\*\*\*

```

0002.00      R SYSER
0002.01      A          EUSER           10A
0004.00      A          EJOBNM          10A
0008.19      A          EDESC           60A

```

\*\*\*\*\* End of data \*\*\*\*\*

**SYSSTAL1**

\*\*\*\*\* Beginning of data \*\*\*\*\*

```

0001.00      A*-----+
0002.00      A* Object.....: SYSSTAL1          |
0003.00      A* Description.....: SYSTEM STATUS EXCEPTIONS BY USER |
0004.00      A*-----+
0005.00      A*
0006.00      A          R SYSER                PFILE (QGPL/SYSSTATE)
0007.00      A*
0008.00      A          K EUSER

```

\*\*\*\*\* End of data \*\*\*\*\*

**SYSSTAL2**

\*\*\*\*\* Beginning of data \*\*\*\*\*

```

0001.00      A*-----+
0002.00      A* Object.....: SYSSTAL2          |
0003.00      A* Description.....: SYSTEM STATUS EXCEPTIONS BY JOB NAME |
0004.00      A*-----+
0005.00      A*
0006.00      A          R SYSER                PFILE (QGPL/SYSSTATE)
0007.00      A*
0008.00      A          K EJOBNM

```

\*\*\*\*\* End of data \*\*\*\*\*

Program “SYSSTATC” takes care of the house-keeping and general program flow, while “SYSSTATR” handles the comparison between “SYSSTAT0” and “SYSSTAT1”.

Upon discovering a candidate job, “SYSSTATR” checks to see if the user or job name is present in the exception file “SYSSTATE”. If the job and user are not excluded, a command string is assembled for the inactive job [e.g. ENDJOB JOB(job#/user/job name) OPTION(\*CNTRL) LOGLMT(0)] and this is passed to the QCMDXC API.

**Special Problems**

If a job is ended manually (e.g. using option ‘4’ on the WRKACTJOB display) there is a delay before the status of that job is updated in the WRKACTJOB statistics. Although program “SYSSTATR” will not attempt to end a job with a status of “END”, it could conceivably attempt to end a job that is already ending by virtue of this delay.

A CPF message will be generated which would normally cause the program to crash. This situation has been trapped using a \*PSSR subroutine. The error will not be fatal and the program will proceed to the next job comparison.

A job that is very active during the interval between runs of the SYSSTAT utility may roll past the upper bounds of the “INT” or “AUXIO” statistics.

In this case, the activity would be represented as “+++”. If there is additional activity beyond “+++” it will not be recorded in that job session.

This creates a problem for the SYSSTAT utility in that a job reaching this level of activity would always appear to be inactive (i.e. will not have changed) on subsequent runs of the utility whether it was active or not.

## Object List

### SYSSTATC – (CL)

Program that takes care of housekeeping and captures activity data.

### SYSSTAT – (PF)

A temporary flat file built in QTEMP, contains WRKACTJOB data copied from spool.

### SYSSTAT0 – (PF)

Both SYSSTAT0 and SYSSTAT1 use same DDS (maps WRKACTJOB data to file).

### SYSSTAT1 – (PF)

Built using DDS for SYSSTAT0.

### SYSSTATE – (PF)

Exception file.

### SYSSTAL1 – (LF)

Logical over the exception file by user name.

### SYSSTAL2 – (LF)

Logical over the exception file by job name.

### SYSSTATR – (RPG)

Program that compares activity data and ends inactive jobs.

## SYSSTATR

```

***** Beginning of data *****
0001.00 *****
0002.00 * Program Name: SYSSTATR *
0003.00 *****
0004.00 * Description: Compares SYSSTAT0 and SYSSTAT1 to *
0005.00 * determine if any jobs have been *
0006.00 * inactive since the program was last *
0007.00 * run. If the job is inactive and the *
0008.00 * user or jobname is not in the *
0009.00 * exception file, the job is ended. *
0012.00 *
0013.00 * Written by: Source Code Technologies Inc. *
0014.00 *
0015.00 * Date: April 4, 2002 *
0016.00 *
0017.00 * Called by: SYSSTATC *
0017.01 *****
0017.02 * Notes: *
0017.03 * ~~~~~ *
0017.04 * Consider the following command string: *
0017.05 * *
0017.06 * ENDJOB JOB(JOB###/USERXXXXXX/JOBXXXXXX) OPTION *
0017.07 * (*CNTRLD) LOGLMT(0) *
0017.08 * The maximum size of the command string is 66 char. *
0017.09 * Within this string there are three variable length *
0017.10 * elements, namely the job number, user name and *
0017.11 * job name. The command will error out if there *
0017.12 * are inappropriate spaces in the string. *
0017.13 *
0017.14 * FINPOS returns the position of the nth occurrence *
0017.15 * of a target character in an array. This is used *
0017.16 * to build a command string without unwanted spaces *
0023.00 *****
0024.00 * MAINTENANCE LOG *
0025.00 * ~~~~~ *
0026.00 * PROJ DATE INIT DESCRIPTION *
0027.00 * ~~~~ ~~~~ ~~~~ ~~~~~ *
0028.00 * $XXX XX/XX/XX XXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX *
0029.00 * *
0030.00 *****
0031.00 *
0032.00 FSYSSTAT0IP E K DISK
0033.00 FSYSSTAT1IF E K DISK
0033.02 F SYSOR KRENAMESYS1R
0033.03 FSYSSTAL1IF E K DISK
0033.04 FSYSSTAL2IF E K DISK
0033.05 F SYSER KRENAMESYSE2
0036.00 *
0036.01 E A 1 1 11
0036.02 E B 1 1 27
0036.03 E CM 66 1
0036.04 *
0037.00 ISYS1R
0037.02 I SJOBNM JOBNM1
0037.04 I SUSER USER1
0037.05 I SJOBNR JOBNR1
0037.06 I SINTER INTER1
0037.07 I SAUXIO AUXIO1
0037.08 I SSTATU STATU1
0037.09 *
0037.10 IXCMD DS
0037.11 I 1 66 CM
0037.12 IPSSR SDS
0037.13 I 1 10 PROGRM
0037.14 I 11 150PSTATS
0037.15 I 16 200PPSTS
0037.16 I 21 28 SRC#

```

Therefore, it is very significant that during the collection of the statistics to the "SYSSTAT1" file at the end of the "SYSSTATC" process, the \*reset option is used on this WRKACTJOB \*print command (and only on this instance of the command).

## Maintenance considerations

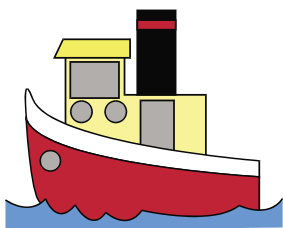
The utility as it appears here was written on V5R1M0. Potentially, any new release might change the format of the Work Active Jobs display, which would shift the fields that are mapped into file "SYSSTAT0", and "SYSSTAT1". Be prepared to revise the "SYSSTAT0" DDS if the program stops working after an operating system upgrade.

Some library names have been changed in the code examples. Anywhere that you see QGPL you may wish to customize the code to reflect the libraries appropriate to your environment.



Graham Cromar is an independent contractor based in Toronto, Ontario. He can be reached via email to: [gcromar@source-code-technologies.com](mailto:gcromar@source-code-technologies.com).

\* Special thanks to David Hanlan for research and additional testing.



```

0037.17 I 29 36 RTNNAM
0037.18 I 37 390PARMS#
0037.19 I 40 46 CPFMSG
0037.20 I 47 50 MI#
0037.21 I 51 80 FIL1
0037.22 I 81 90 PGMLIB
0037.23 I 91 170 RTVDTA
0037.24 I 171 174 FAILID
0037.25 I 244 253 JOBNAM
0037.26 I 254 254 ZCHAR
0037.27 I 255 259 ZDEALR
0037.28 I 254 263 JOBUSR
0037.29 I 264 269 JOBNBR
0037.30 I 270 2750DATEU
0037.31 I 276 2810DATEEX
0037.32 I 282 2870TIMEX
0037.33 I 288 428 FIL2
0047.00 *
0048.00 * Mainline
0049.00 *
0050.00 *
0050.01 C SETOF 1314 \
0050.02 * \
0051.00 C STAKEY CHAINSYSSTAT1 30 \
0052.00 * \
0053.00 C *IN30 IFNE *ON \
0053.01 C SINTER COMP INTER1 1313 \
0053.02 C SAUXIO COMP AUXIO1 1414 \
0053.04 C *IN13 IFEQ *OFF \
0053.05 C *IN14 ANDEQ*OFF \
0053.06 C SSTATU ANDNE'END \
0053.08 C SUSER CHAINSYSSTAL1 31 \
0053.09 C SJOBNM CHAINSYSSTAL2 32 \
0053.10 C *IN31 IFEQ *ON \
0053.11 C *IN32 ANDEQ*ON \
0053.12 C EXSR ENDJOB \
0053.13 * \
0053.14 C RESUME TAG \
0053.15 * \
0054.00 C ENDIF \
0054.01 C ENDIF \
0054.02 C ENDIF \
0054.03 * \
0056.01 *****
0056.02 * ENDJOB Subroutine *
0056.03 *****
0056.04 C ENDJOB BEGSR \
0056.06 *
0056.07 * Assemble CL command string
0056.08 *
0056.09 *
0056.10 C MOVEA*BLANKS CM,1 \
0056.11 * \
0056.12 C MOVEAA,1 CM,1 \
0056.13 C MOVEASJOBNR CM,12 \
0056.14 * \
0056.15 C Z-ADD2 N 30 \
0056.16 C MOVE \ \ C 1 \
0056.17 C EXSR FINPOS \
0056.18 * \
0056.19 C MOVEA'/' CM,X \
0056.20 C ADD 1 X \
0056.21 C MOVEASUSER CM,X \
0056.22 * \
0056.23 C Z-ADD2 N 30 \
0056.24 C MOVE \ \ C 1 \
0056.25 C EXSR FINPOS \
0056.26 * \
0056.27 C MOVEA'/' CM,X \
0056.28 C ADD 1 X \
0056.29 C MOVEASJOBNM CM,X \
0056.30 * \
0056.31 C Z-ADD2 N 30 \
0056.32 C MOVE \ \ C 1 \
0056.33 C EXSR FINPOS \
0056.34 * \
0056.35 C MOVEAB CM,X \

```

```

0056.44 *-----/
0056.45 * End the job \
0056.46 *-----/
0056.47 C          Z-ADD66      XLEN   155 \
0056.48 C          CALL 'QCMDEXC' \
0056.49 C          PARM          XCMD \
0056.50 C          PARM          XLEN \
0056.53 *-----/
0056.54 C          ENDSR \
0083.00 * \
0083.01 ***** \
0083.02 * FINPOS Subroutine * \
0083.03 ***** \
0083.04 * Returns position of Nth occurrence of a character * \
0083.05 * in the array. * \
0083.06 * \
0083.07 *           X = Position of Nth occurrence * \
0083.08 *           N = Desired occurrence to search for * \
0083.09 *           C = Character to search for * \
0083.10 * \
0083.11 ***** \
0083.12 C          FINPOS      BEGSR \
0083.13 *-----/
0083.14 * Return position for next move \
0083.15 *-----/
0083.16 C          Z-ADD0      X       30 \
0083.17 C          Z-ADD0      OCCUR   30 \
0083.18 * \
0083.19 C          DO          50      POS    30 \
0083.20 * >>>>>> \
0083.21 C          CM,POS      IFEQ C \
0083.22 C          ADD          1      OCCUR \
0083.23 C          ENDIF \
0083.24 * \
0083.25 C          OCCUR      IFEQ N \
0083.26 C          ADD POS      X \
0083.27 C          LEAVE \
0083.28 C          ENDIF \
0083.29 * <<<<<< \
0083.30 C          ENDDO \
0083.31 *-----/
0083.32 C          ENDSR \
0084.00 ***** \
0085.00 * Initialization Subroutine * \
0086.00 ***** \
0087.00 C          *INZSR      BEGSR \
0088.00 *-----/
0089.00 C          STAKEY      KLIST \
0090.00 C          KFLD          SUSER \
0091.00 C          KFLD          SJOBNM \
0091.01 C          KFLD          SJOBNR \
0091.02 * \
0091.03 C          MOVE '0'      SWITCH 1 \
0098.00 *-----/
0099.00 C          ENDSR \
0099.01 ***** \
0099.02 * Program status subroutine * \
0099.03 ***** \
0099.04 C          *PSSR      BEGSR \
0099.05 *-----/
0099.06 * If have already been here, get out to avoid looping' \
0099.07 *-----/
0099.08 C          SWITCH      IFEQ '1' \
0099.09 C          SETON          LR \
0099.10 C          ENDIF \
0099.11 *-----/
0099.12 * Set on switch to indicate we've been here \
0099.13 *-----/
0099.14 C          MOVE '1'      SWITCH \
0099.15 *-----/
0099.16 * Ignore error message on call to QCMDEX \
0099.17 *-----/
0099.18 * \
0099.20 C          GOTO RESUME \
0099.21 *-----/
0099.22 * \
0099.23 C          ENDSR \
0123.00 ***** \
0124.00 ** \
0125.00 ENDJOB JOB ( \
0126.00 ** \
0127.00 ) OPTION(*CNTRL) LOGLMT(0) \
          ***** End of data *****

```

**COMMON<sup>®</sup>**  
A USERS GROUP



## Next Conference:

**FALL 2002:**  
October 13-17, 2002

**Denver, Colorado**



## Future Conferences:

**SPRING 2003:**

March 9-13, 2003  
Indianapolis

**FALL 2003:**

September 7-11, 2003  
Orlando

**SPRING 2004:**

May 2-6, 2003  
San Antonio

**FALL 2004:**

October 17-21, 2004  
Toronto