

```

COPY $HASPGBL          Copy HASP GLOBALS
$MODULE TITLE='SAPIWTR',
    CTOKEN,            Generate MVS CTOKEN
    CVT,               Generate MVS CVT
    SSS2,              Generate MVS IAZSSS2 Dsect
    JESCT,             Generate MVS JESCT
    $HASPEQU           Generate HASP equates
SPACE 1
HASSAPW CSECT
HASSAPW AMODE 31
using HASSAPW,R12      Local addressability
using work,r10         Work area
using sss2,sss2d       SSS2 addressability
using ssob,ssobh       SSOB header addressability

bakr r14,0            save callers registers

lr r12,r15            Local base
lhi r2,workklen       Length of work area
storage OBTAIN,sp=0,length=(r2)
lr r10,R1             Work area base
lr r0,R1              Zero
lhi r1,workklen       work
slr r15,R15           area
mvcl r0,R14          ...

* -----
*   Get storage for SVC 99 RB
* -----
using s99rb,r11        Address SVC 99 RB
la r2,rblen           Length of SVC 99 area
storage OBTAIN,length=(r2),loc=24
lr r11,r1             Address of the storage
lr r0,R1              Zero
la r1,rblen           work
slr r15,R15           area
mvcl r0,R14          ...

* -----
*   Build the SSOB Control Block
* -----
la r0,ssob            set pointer to
st r0,ssobptr         ssob base
la r0,sss2            set address
st r0,ssobindv        of extension
mvc ssobid,=c14'SSOB' set eye catcher
mvc ssoblent,=y(ssobhsiz) set size
mvc ssobfunc,=y(ssobsou2) set sapi function

mvc sss2lent,=y(sss2size) set extension length
mvi sss2type,sss2puge type is putget
mvc sss2eye,=c'SSS2'  set eye catcher

* -----
*   Set selection criteria
* -----
tput jobreq,20        request jobname
tget jobname,8        get jobname
oc jobname,=c18' '    convert to upper case

```

```

mvi    sss2sell1,sss2sjbn+sss2sawt    select by jobname any type
mvi    sss2dsp1,sss2dkpe+sss2rnpt    keep the data set
mvc    sss2jobn,jobname              Jobname to get
tput   sss2jobn,8                    Echo back the jobname

```

```

modeset mode=SUP                      Enter Supervisor State

```

```

* -----
*      Make a SAPI Request
* -----

```

```

nextds equ    *
la      r1,ssobptr                    Pass address of SSOB
IEFSSREQ ,                            Make SSI call

ltr     R15,R15                      Continue
jz      X01000                       if OK
la      r8,badsapi                   Bad SAPI Request
j       abend                         Abend with error

```

```

* -----
*      Loop if more datasets, bump count
* -----

```

```

X01000 EQU    *
clc     ssobretn,=a(sss2eods)        Any more datasets?
je      exit                          n, we are all done

```

```

* -----
*      Allocate the JES output dataset
* -----

```

```

X02500 equ    *
tput    blank,1
TPUT    sss2dsn,44                   Say dataset name
la      r8,badrlen                   Assume size too large for WTO
slr     r6,r6                         Clear R6
icm     r6,b'0011',sssomlrl          Get max record length
ch      r6,=h'150'                   record too big?
jnl     abend                         y, abend
sth     r6,reclen                    save record length

xc      s99rb(rblen),s99rb           clear the RB
mvi     s99rbln,rblen                Set length of the RB
mvi     s99verb,s99vrbal             Set RB Verb to "Allocate"
mvi     s99flg11,s99jbsys           Sysout dataset
la      r1,my99tpta                  Address SVC Alloc TU Ptrs
st      r1,s99txtpp                  Store TU Ptrs in RB
la      r1,s99rb                     Get Address of RB
st      r1,myrbptr                   Save RB Address
oi      myrbptr,x'80'                Turn on high bit
mvc     txtdsn,sss2dsn               Move dataset name to be allocated
la      r0,txtbrtkn                  Target Address Dyn Browse Token
la      r1,btoksize                  Target Length Browse Token
l       r14,sss2btok                 From Address
lr      r15,r1                       From Length
mvcl   r0,r14                       Move the Browse Token
mvc     txtjes,=c14'JES2'           Set the subsystem
la      r1,myrbptr                   -> to RB Pointer
dynamalloc ,                          Allocate the JES Dataset
la      r8,bads99a                   Assume it didn't work
lr      r9,r1                         Copy for dump

```

```

l      r7,s99error      RSN Code / Info
ltr    r15,r15          Allocation Work?
jnz    abenddyn         n, say so

```

```

*
*
*
*

```

```

-----
SYSOUT Dataset allocated? Move returned DDName into
DCB prior to opening it.
-----

```

```

la      r4,indcb        -> to the input dcb
using  ihadcb,r4        address the DCB
mvc     dcbddnam(8),txtdda99 move in returned ddname
mvc     txtddu99,txtdda99 save for unallocation
mvc     dcblrecl,sssomlrl move max length record in

```

```

open   indcb           open the dcb
la      r8,badopen     assume the open failed
lr      r9,r4          copy for dump
tm      dcboflgs,dcbofopn File open?
jno    abend          n, take a dump

```

```

getnext equ  *          loop for reading/displaying
jas     r14,set24      set 24 bit mode
get     indcb          r1==> record after the get
jas     r14,set31      return to 31 bit mode

```

```

lh      r6,reclen      point to record for output
mvi     rectext,c' '   clear record out...
mvc     rectext+1(1'rectext-1),rectext ..for next one
moveit  mvc  rectext(0),0(R1) move the record
ahi     r6,-1          decrement it for the move
ex      r6,moveit     do it for real
tput   rectext,133    write the line
j       getnext       go get next record

```

```

set24   icm  14,8,=x'00' High order byte cleared
bsm     0,14          return in 24 bit mode

```

```

set31   icm  14,8,=x'80' Set high order bit
bsm     0,14          return in 31 bit mode

```

```

myeodad ds  0h          end-of-dataset
close  indcb          close the input dcb
jas     r14,set31     return to 31 bit mode
drop   r4             ihadcb

```

```

*
*
*

```

```

-----
Unallocate the SYSOUT Dataset
-----

```

```

xc      s99rb(rblen),s99rb zero the rb
mvi     s99rbln,rblen   rb length
mvi     s99verb,s99vrbuf rb verb code=unalloc
la      r1,my99tptu     addr svc 99 alloc tu ptrs
st      r1,s99txtpp     stored in rb
la      r1,s99rb        Get Address of RB
st      r1,myrbptr      Save RB Address
oi      myrbptr,x'80'   Turn on high bit
la      r1,myrbptr      pt to rb pointer
dynalloc ,             issue dynamic unallocation
la      r8,bads99u     assume it didn't work
lr      r9,r1           copy for dump

```

```

ltr   r15,r15          svc 99 work okay??
jnz   abend            n, take a dump
j     nextds          go get next data set

```

```

* -----
*      Clean up and exit
* -----

```

```

exit  equ   *
      la    r2,rblen          Length of storage to free
      storage RELEASE,length=(r2),addr=(r11)

      storage RELEASE,length=worklen,addr=(r10)

      pr    ,

```

```

abend equ   *
      l     r6,ssobretn      Get return code
      slr   r7,r7            Clear Register 7
      ic    r7,sss2reas      Get reason code
      abend (r8),DUMP        Abend with a dump

```

```

abenddyn equ   *
      abend (r8),DUMP        Abend with a dump

```

```

      ds    0f
jobreq  dc   cl20'Enter Jobname'
dsnnotf dc   cl30'No datasets found'
jobname dc   cl8' '

```

```
ltorg ,
```

```

* -----
*      S99 REQUEST BLOCK
* -----

```

```

my99rb ds cl(rblen)          my svc 99 rb
rblen  equ (s99rbend-s99rb)  length of rb for my99rb

```

```

* -----
*      Text unit pointers for allocation
* -----

```

```

my99tpta dc a(txtalds)          tu for dataset name
          dc a(txtbrtkn)        JES Browse Token
          dc a(txtssreq)        name of subsystem tu ptr
          dc a(equhobon+txtrtdn) return dd name tu

```

```

* -----
*      Text unit pointers for unallocation
* -----

```

```

my99tptu dc a(txtdundd)        tu for unalloc by ddname
          dc a(equhobon+txtdunnh) nohold tu

```

```

* -----
*      Allocate Text Units
* -----

```

```

txtalds dc al2(daldsnam)       dataset name key
          dc x'0001'            number
          dc al2(44)            dsname length
txtdsnam ds cl44                dsname from iefssreq

```

```

txtbrtok dc al2(dalbrtkn)          JES Browse Token
          dc al2(7)                # parameters
txtbrtkn ds cl(btksize)           Token Length and data

txtssreq dc al2(dalssreq)         request of subsystem
          dc x'0001'               # field (0001 required)
          dc x'0004'               len of ss name following
txtjes   ds cl4                   name of subsystem

txtrtdn  dc al2(dalrtdn)         return ddname field
          dc x'0001'               # field (0001 required)
          dc x'0008'               len of parm
txtdda99 dc cl8' '               returned ddname parm field

txtclose dc al2(dalclose)        unallocate at close key
          dc x'0000'               # field (0000 required)

```

```

* -----
*      Unallocate Text Units
* -----

```

```

txtdundd dc al2(dunddnam)        tu for ddname unalloc
          dc x'0001'               number
          dc al2(8)                ddname length
txtddu99 ds cl8                  ddname from dynalloc
txtdunnh dc al2(dunovsnh)        tu for nohold
          dc x'0000'               # field (0000 required)

```

```

myssobpt ds f                    pointer to ssob for iefssreq
nomore   dc al2(sssoeods)        no more datasets from jes
reclen   ds h                    length of output record
rectext  ds cl256                up to 256 bytes of sysout
blank    dc c' '                 blank

```

```

* -----
*      DCB to read SYSOUT from JES
* -----

```

```

INDCB    DCB DSORG=PS,MACRF=GL,BUFNO=2,EODAD=MYEODAD,      X
          DDNAME=WILLCHNG

```

```

* -----
*      Abend Codes
* -----

```

```

badr15   equ 1                   iefssreq r15 non-zero
badretn  equ 2                   ssobretn non-zero and not 8
bads99a  equ 3                   dynalloc alloc failed
badopen  equ 4                   open dcb failed
bads99u  equ 5                   dynalloc unallc failed
badrlen  equ 6                   pso dataset too large (reclen)
nodsns   equ 10                  No Dataset Found
badsapi  equ 11                  Bad SAPI Call

```

```

equhobon equ x'80000000'         High Order bit on

```

```

DCBD DSORG=PS
IEFZB4D0
IEFZB4D2
IEFSSOBH
SSOBGN EQU *
IEFSSSO SOEXT=YES

```

IAZBTOKP ,

WORK	DSECT		
SSOBH	DS	XL(SSOBHSIZ)	SSOB header
SSS2D	DS	XL(SSS2SIZE)	SSOB extension
MYRBPTR	DS	F	My SVC 99 RB Pointer
SSOBPTR	DS	A	Address of SSOB
DATATOKN	DS	H	Number of data sets
DATATOK	DS	50XL(CTKNSIZE)	Data set tokens
WORKLEN	EQU	*-WORK	
HASSAPW	CSECT		
	END	,	