

```

COPY $HASPGBL          Copy HASP GLOBALS
$MODULE TITLE='SAPIWTR',
    CTOKEN,            Generate MVS CTOKEN
    CVT,               Generate MVS CVT
    JESCT,            Generate MVS JESCT
    SSOB,             Generate MVS SSOB
    SSS2,            Generate MVS IAZSSS2 Dsect
    $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
SPACE 1
SAVE (14,12)         Declare entry point
SPACE 1
LR R12,R15           Local base
GETMAIN R, LV=WORKLEN Working memory
LR R10,R1            Work area base
LR R0,R1             Zero
LHI R1,WORKLEN       work
SLR R15,R15          area
MVCL R0,R14          ...
LA R1,SAVEAREA       Addr new save area
ST R13,4(,R1)        Link
ST R1,8(,R13)        save
LR R13,R1            areas

```

```

* -----
* Build the SSOB
* -----

```

```

LA R0,SSOB           Set pointer to
ST R0,SSOBPTR        SSOB base
LA R0,SSS2           Set address
ST R0,SSOBINDV       of extension
MVC SSOBID,=CL4'SSOB' Set eye catcher
MVC SSOBLEN,=Y(SSOBHSIZ) Set size
MVC SSOBFUNC,=Y(SSOBSOU2) Set SAPI function

MVC SSS2LEN,=Y(SSS2SIZE) Set extension length
MVI SSS2TYPE,SSS2PUGE Type is PUTGET
MVC SSS2EYE,=C'SSS2' Set eye catcher

```

```

* -----
* Set selection criteria
* -----

```

```

MVI SSS2SEL1,SSS2SCLS Select by class
MVI SSS2CLSL,C' '     Blank class
MVC SSS2CLSL+1(L'SSS2CLSL-1),SSS2CLSL list
MVI SSS2CLSL,C'J'     Use Class A
MVI SSS2DSP1,SSS2DKPE+SSS2RNPT Keep the data set

```

```

MODESET MODE=SUP      Enter Supervisor State

```

```

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

```

X00500 LA R1,SSOBPTR Pass address of SSOB
IEFSSREQ , Make SSI call

LTR R15,R15 Continue
JZ X01000 if OK
WTO 'SAPI - X00500 - Bad SAPI Call'
DC H'0' else take a dive

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

X01000 EQU *
wto 'SAPI1 - X01000'
CLC SSOBRETN,=A(SSS2EODS) Br if no more
JE X02000 data sets
LH R1,DATATOKN Bump
AHI R1,1 the data set
STH R1,DATATOKN number
MHI R1,CTKNSIZE Compute offset in token
LA R1,DATATOK-CTKNSIZE(R1) Addr of sink
L R15,SSS2DSTR Addr of source token
MVC 0(CTKNSIZE,R1),0(R15) Copy token
J X00500 Get next data set

* -----
* Check if any datasets are found
* -----

X02000 OC DATATOKN,DATATOKN Continue if at
JNZ X02500 least one data set
WTO 'SAPI - X02000 - No datasets found'
DC H'0' else take a dive

* -----
* Selection Criteria on Class Y output
* -----

X02500 MVI SSS2SEL5,SSS2SCTK Token select
MVI SSS2DCLS,C'P' Change class to P
LA R0,DATATOK+2*CTKNSIZE Third token
ST R0,SSS2CTKN Set address of token
LH R1,DATATOKN Index of last data set
MHI R1,CTKNSIZE Offset of last token
LA R1,DATATOK-CTKNSIZE(R1) Addr last token
ST R1,SSS2CTKN Tell SAPI

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

LA R1,SSOBPTR Set SSOB pointer
IEFSSREQ , Fetch one data set
LTR R15,R15 Continue
JZ X03000 if OK
WTO 'SAPI - X02500 - Bad request'
DC H'0' else take a dive

X03000 CLC SSOBRETN,=A(SSS2RTOK) Continue
JE X04000 if OK
DC H'0' else take a dive

* -----

* Make a SAPI Request

* -----
X04000 LA R1,SSOBPTR Fetch the data set again

IEFSSREQ , Fetch one data set
CLC SSOBRETN,=A(SSS2EODS) Continue
JE X05000 if OK
DC H'0' else take a dive

* -----
* Clean up and exit

* -----
X05000 LR R1,R13 Address of gotten memory
L R13,4(,R13) Pick up Save Area address
FREEMAIN R,A=(1),LV=WORKLEN Free work area
RETURN (14,12),RC=0 Return to the caller

WORK DSECT
SAVEAREA DS 18A
SSOBPTR DS A Address of SSOB
SSOBH DS XL(SSOBHSIZ) SSOB header
SSS2D DS XL(SSS2SIZE) SSOB extension
DATATOKN DS H Number of data sets
DATATOK DS 1024XL(CTKNSIZE) Data set tokens
WORKLEN EQU *-WORK
HASSAPW CSECT
LTORG ,
END ,