

OVERVIEW of ARCHITECTURE

ADASTRIP is invoked by executing the module named STRIP. To run the zIIP version, module STRIP must be APF authorised and loaded from an APF authorised library. That is the only module that is APF authorised. STRIP checks its environment and if it is running APF authorised it attempts to use the zIIP feature. If not, it disables the zIIP feature and runs completely in normal TCB mode. If the zIIP feature is possible, STRIP sets up a set of Program Call (PC) routines that are used to initialise the zIIP SRB environment and perform services that SRBs are not allowed to perform. Once initialisation is complete, STRIP turns off APF authorisation and either LINKs to (non-zIIP mode) or ATTACHes (zIIP mode) STRI8. From then on, ADASTRIP runs non-APF authorised.

The PC routines:

- +Set the address space non-swappable;
- +Schedule the zIIP SRB;
- +Delete the enclave where the SRB was running;
- +Query if the program is executing in an enclave;
- +Resume a suspended SRB;
- +Obtain the local lock;
- +Release the local lock and
- +Save the THREAD DSECT address in the TCBUSER field.

When in zIIP mode, the module STRI82 runs in a zIIP possible SRB. To interface between the mother task running on the main processor and STRI82 running under an SRB, two new modules were written. STRI82I and STRI82Z.

STRI82I is the interface for calls to STRI82 so they are diverted to the zIIP SRB process. If the zIIP feature is OFF, STRI82 is called in non-zIIP mode. If the zIIP feature is ON, STRI82I is called instead to direct STRI82 processing to the zIIP processor.

STRI82Z receives control under an SRB, sets up register 11 pointing to the globals area and register 1 pointing to the scheduling task's THREAD DSECT, then calls STRI82 to process the passed RABN. When STRI82 wants to perform IO or issue a WTO (activities prohibited under SRBs) it posts an ECB that STRI82I is waiting on and suspends the SRB. STRI82I executes the requested service then resumes the suspended SRB so that STRI82 can continue. When STRI82 finishes with a RABN it returns to STRI82Z which posts a 'RABN processed' ECB and waits for the next RABN to be passed by ADASTRIP's main processor task.

To save overhead, the SRB is scheduled only once. After each RABN is processed, the SRB suspends to await the next RABN or be told that all RABNs have been processed. Only when all RABNs have been processed does the SRB terminate.

JCL CONSIDERATIONS

The APF authorised library that contains module STRIP must be in the STEPLIB. All other ADASTRIP modules must be in the //PROGLIB DD concatenation. Here follows an example of JCL to execute ADASTRIP in zIIP mode.

```
//<<jobcard>>
//*
//* Test ADASTRIP with zIIP support against a database.
//*
//*
//DEL EXEC PGM=IDCAMS,REGION=5000K
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE 'CCA.ADASTRIP.V507AZIP.TEST2'
SET MAXCC=0
//STRIP EXEC PGM=STRIP,REGION=10M,
// PARM='provide zip code'
//STEPLIB DD DSN=HLQ.AUTH.TEST.LOAD,DISP=SHR
//*
//* Copy load module STRIP to an AUTH load lib
//PROGLIB DD DSN=CCA.ADASTRIP.ZIP.LOAD,DISP=SHR
//SYSABEND DD SYSOUT=*
//STPRINT DD SYSOUT=*
//STMESS DD SYSOUT=*
//STDEBUG DD DSN=CCA.ADASTRIP.ZIP.CNTL(STDEBUG),DISP=SHR
//*
//STASSO DD DSN=SAG.ADABAS.DB00001.ASSOR1,DISP=SHR
//STDATA DD DSN=SAG.ADABAS.DB00001.DATAR1,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,10)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,10)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,10)
//SORTOUT DD UNIT=SYSDA,SPACE=(CYL,30)
//SYSOUT DD SYSOUT=* SORT MESSAGES (MAY VARY WITH SORT PACKAGE)
//TESTA DD DSN=CCA.ADASTRIP.ZIP.TEST2,DISP=(,CATLG,CATLG),
// SPACE=(CYL,(1,1),RLSE),UNIT=SYSDA,BLKSIZE=0,
// LRECL=23000,RECFM=VB
//STPARM DD *
FIELDTAB 4
TESTA TEST 1 AC.EQ.C'SIMONE'
TESTA RULE 1
TESTA FILE 1
TESTA FIELDS ## **
```