

```

*****
*
* ASMPIPI
*
* Must be called from a REXX driver
*
* 1 - Load and Call CEEPIPI to initialize a subroutine environment.
*
* 2 - Call IRXEXCOM to set REXX variables to return the
* address of CEEPIPI and the address of the environment handle.
*
* Note: ASMPIPI is NOT reentrant.
* Note: Assemble with SYSPARM(DEBUG) to enable debugging messages
*
* 2012/03/15 - Barry Lichtenstein - Clean up
* 2011/02/25 - Barry Lichtenstein - Original
*
*****
* ++++++
*          GBLA          &DEBUG
&DEBUG   SETA (',&SYSPARM,' INDEX ',DEBUG,')
* ++++++
*
* =====
* Standard program entry conventions.
* =====
ASMPIPI  CSECT
ASMPIPI  AMODE 31
ASMPIPI  RMODE ANY
          STM   R14,R12,12(R13)   Save caller's registers
          LR    R12,R15           Get base address
          USING ASMPIPI,R12      Identify base register
          ST    R13,SAVE+4       Back-chain the save area
          LA    R15,SAVE         Get addr of this routine's save area
          ST    R15,8(R13)       Forward-chain in caller's save area
          LR    R13,R15          R13 -> save area of this routine
*
          LR    R3,R0            Save usable ENVblock
          USING ENVBLOCK,R3
*
* *****
* Load LE CEEPIPI service routine into main storage.
* *****
          LOAD  EP=CEEPIPI       Load CEEPIPI routine dynamically
          ST    R0,PPRTNPTR      Save the addr of CEEPIPI routine
*
* =====
* Initialize an LE Preinitialization subroutine environment.
* =====
INIT_ENV EQU *
          AIF (&DEBUG EQ 0).NDBG001
          WTO  'ASMPIPI : about to (INIT_SUB)',ROUTCDE=11
.NDBG001 ANOP
          LA   R5,PPTBL          Get address of Preinitialization Tabl
          ST   R5,@CEXPTBL      Ceexptbl-addr->Preinitialization Tabl
          L    R15,PPRTNPTR     Get address of CEEPIPI routine
*                               Invoke CEEPIPI routine
          CALL (15), (INITSUB,@CEXPTBL,@SRVRTNS,RUNTMOPT,TOKEN)
*                               Check return code:
          LTR  R2,R15           Is R15 = zero?
          BZ   SETVAR           Yes (success).. go to next section
*                               No (failure).. issue message, abend
          WTO  'ASMPIPI : INIT_SUB failure RC is not 8.',ROUTCDE=11
          ABEND (R2),DUMP       Abend with bad RC and dump memory
*
* =====
* Set the REXX variables PIPIADDR and PIPITOKN for access

```

```

* to the REXX driver, for use on subsequent calls into CEEPIPI
* =====
SETVAR    EQU    *
          L      R0,TOKEN
          ST     R0,TOKENCPY
*
* *****
* First save the PIPI information needed later,
* in order to return in REXX vars
* *****
          MVC    vvar1(4),PPRTNPNTR
          MVC    vvar2(4),TOKENCPY
*
* ++++++
          AIF (&DEBUG EQ 0).NDBG003
          WTO   'ASMPIPI : about to ASMPIPC',ROUTCDE=11
          WTO   'ASMPIPI about to SETVAR',ROUTCDE=11
.NDBG003 ANOP
* ++++++
* *****
* Create a shared variable request block for last variable
* *****
          LA     R6,shvb2
shvr2     USING SHVBLOCK,r6
          MVC    shvr2.SHVNEXT,=F'0'
          MVC    shvr2.SHVUSER,=F'0'
          MVI    shvr2.SHVCODE,SHVSTORE
          MVC    shvr2.SHVNAMA,=A(var2)
          MVC    shvr2.SHVNAML,=A(evar2-var2)
          MVC    shvr2.SHVVALA,=A(vvar2)
          MVC    shvr2.SHVVALL,=A(evvar2-vvar2)
*
* *****
* Chaing & Create a shared variable request block for next variable
* *****
          LR     R0,R6
          LA     R6,shvb1
shvr1     USING SHVBLOCK,R6
          ST     R0,shvr1.SHVNEXT
          MVC    shvr1.SHVUSER,=F'0'
          MVI    shvr1.SHVCODE,SHVSTORE
          MVC    shvr1.SHVNAMA,=A(var1)
          MVC    shvr1.SHVNAML,=A(evar1-var1)
          MVC    shvr1.SHVVALA,=A(vvar1)
          MVC    shvr1.SHVVALL,=A(evvar1-vvar1)
* *****
* *****
          LA     R5,=CL8'IRXEXCOM'
          ST     R5,parm1
* *****
          LHI    R5,0
          ST     R5,parm2
          ST     R5,parm3
* *****
          ST     R6,parm4
* *****
          OI     parm4,X'80'
* *****
          LR     R0,R3                restore ENVBLOCK for the call!
          LA     R1,plist
*
* *****
          LINK   EP=IRXEXCOM
          ST     R15,myret
*
* ++++++
          AIF (&DEBUG EQ 0).NDBG004

```

```

        WTO      'ASMPIPI called   SETVAR',ROUTCDE=11
.NDBG004 ANOP
* ++++++
*
* =====
* Standard exit code.
* =====
DONE      EQU      *
*         LA       R15,0           Passed return code for system
*         L        R15,myret
*         L        R13,SAVE+4      Get address of caller's save area
*         L        R14,12(R13)     Reload caller's register 14
*         LM       R0,R12,20(R13)  Reload caller's registers 0-12
*         BR       R14            Branch back to caller
*
* =====
* CONSTANTS and SAVE AREA.
* =====
SAVE      DC       18F'0'
PPRTNPTR DS       A              Save the address of CEEPIPI routine
TOKENCPY DS       F              Save a copy of the TOKEN
*
* =====
* Parameters passed to a (INIT_SUB) call.
* =====
INITSUB   DC       F'3'          Function code to initialize for subr
@CEXPTRL DC       A(PPTBL)      Address of Preinitialization Table
@SRVRTNS DC       A(0)          Addr of service-rtns vector, 0 = none
* -----
* why don't we get stdout output from the HLL? we do get stderr !
* -----
RUNTMOPT  DC       CL255''       Fixed length string of runtime optns
          ORG      RUNTMOPT
          DC       C'RPTOPTS(ON) '
          DC       C'MSGFILE(SYSPRINT) '
*         DC       C'ENVAR(_CEE_ENVFILE_S='
*         DC       C'/u/barryl/binder/SHARE/SHARE118/PIPI/envfile )'
          ORG      RUNTMOPT+L'RUNTMOPT
TOKEN     DS       F              Unique value returned (output)
*
* =====
* Parameters passed to a (CALL_SUB) call.
* =====
CALLSUB   DC       F'4'          Function code to call subroutine
PTBINDEXT DC       F'0'          The row number of Preinitialization
*                               Table entry
PARMPTR   DC       A(0)          Pointer to @PARMLIST or zero if none
SUBRETC   DS       F              Subroutine return code (output)
SUBRSNCD  DS       F              Subroutine reason code (output)
SUBFBC    DS       3F            Subroutine feedback token (output)
*
* =====
* Parameters passed to a (TERM) call.
* =====
TERM      DC       F'5'          Function code to terminate
ENV_RC    DS       F              Environment return code (output)
*
* =====
* Preinitialization Table.
* =====
PPTBL     CEEXPIT ,              Preinitialization Table with index
          CEEXPITY HLLPIPI2,0   0 = dynamically loaded routine
*
          CEEXPITS ,              End of PreInit table
*
* =====
* This should all be AUTODATA!
* =====

```

```

MYAREA EQU *
*
* =====
* REXX stuff
* =====
*
* *****
* Shared Variable 1 name and value
* *****
var1 DC C'PIPIADDR'
evar1 EQU *
* *****
vvar1 DC CL4'addr' will be overwritten
evvar1 EQU *
*
* *****
* Shared Variable 2 name and value
* *****
var2 DC C'PIPITOKN'
evar2 EQU *
* *****
vvar2 DC CL4'tokn' will be overwritten
evvar2 EQU *
*
* *****
* Parameters for IRXEXCOM call
* *****
myret DS F
mysize DS F
**
plist DS 0D
parml DS F
parm2 DS F
parm3 DS F
parm4 DS F
*
* *****
* Shared variable block mappings
* *****
shvb1 ORG *+SHVBLEN
shvb2 ORG *+SHVBLEN
*
* *****
MYAREASZ EQU *-MYAREA
*
*
* LTORG
*
* =====
* Control block macro mappings
* =====
envb0 IRXENVB
evalb IRXEVALB
parmb IRXPARB
shrvar IRXSHVB
*
* *****
* Map register names to variables
* *****
YREGS
*
* *****
END ASMPIPI

```