Calling Suprtool

Suprtool, including its Suprlink and Dbedit components, is a utility program, like FCOPY or QUERY. You run it, either interactively or in a batch job, and feed it commands to define a task to be done. How would a user application program invoke Suprtool to perform a desired task? One way would be to use the COMMAND intrinsic to launch a batch job that ran Suprtool. Unfortunately, the user program would have little control over when the batch job started or finished.

To solve this problem, Robelle provides an interface routine that will run Suprtool for a user program, and pass commands from the program to Suprtool (the same commands you would type into Suprtool). This routine (procedure, subroutine, intrinsic) allows user programs to "call" Suprtool. A typical use of this interface would be for a COBOL program to ask Suprtool to extract a selected subset from a large IMAGE dataset and write it to a disc file, which the COBOL program would then read and format into a report.

Suprtool2 Routine

The interface routine is called Suprtool2 (not Suprtool). User programs written in COBOL, COBOL/iX, FORTRAN, FORTRAN/iX, Pascal, Pascal/iX, SPEEDWARE, SPL, or TRANSACT can call Suprtool2 and ask Suprtool to do any of the normal Suprtool tasks such as copy, extract, or sort. The routine creates Suprtool as a son process.

The user program instructs Suprtool by calling the Suprtool2 routine repeatedly with Suprtool command lines. When the first Suprtool command is sent, the interface builds temporary files which will be used for input and output to Suprtool. When the user program sends an Exit command in a separate call to the interface, the interface creates Suprtool as a son process. Finally, the interface prints the $stdlist message file, if so directed by the user program.

Importance of the Exit Command

The interface will not invoke Suprtool until your program passes `Exit` **to it as a command line. The Exit command must be alone and left-justified in the command line. You may use Xeq to separate multiple tasks, but none of the tasks will be executed until you pass Exit to the interface. If you forget the final Exit or put it in the same command line with another command such as Xeq, your Suprtool tasks will be ignored.**

Control Record

The user program must pass a special control record to the interface on each call. For COBOL, we suggest that this control record be placed in the COPYLIB and copied into programs that require the Suprtool interface. The most common error in using the Suprtool2 interface is typing the control record incorrectly. Use the file Cobol.Qlibsrc.Robelle instead. You can copy the control record directly into your COBOL program with the following Qedit command (use /JOIN with EDIT/3000):

```
     /add 50.1=cobol.qlibsrc.robelle
```

Declaration
Status
Command-Line
Priority
Maxdata
Print-State
Total-Type
Totals
Out Count
Workspace

The definition of the control record, with the proper initializing values, is as follows:

```
01  supr-control.
    05  supr-version              pic s9(4) comp value 4.
    05  supr-status               pic s9(4) comp.
        88  supr-ok               value zeros.
        88  supr-bad-msgfiles          value 1.
        88  supr-aborted               value 2.
        88  supr-create-error          value 3.
        88  supr-bad-total-type        value 4.
    05  supr-command-line         pic x(256) value spaces.
    05  supr-flags.
        10  supr-priority         pic x(2) value spaces.
            88  supr-priority-cs       value "CS".
            88  supr-priority-ds       value "DS".
            88  supr-priority-es       value "ES".
        10  supr-maxdata          pic s9(9) comp value 0.
        10  supr-print-state      pic x(2) value "ER".
            88  supr-print-on-error    value "ER".
            88  supr-print-always      value "AL".
            88  supr-print-never       value "NE".
        10  supr-total-type       pic x(2) value "CO".
            88  supr-total-cobol  value "CO".
            88  supr-total-ascii  value "AS".
        10  supr-other-flags      pic x(18) value spaces.
    05  supr-totals pic s9(17) sign is trailing
                 separate character occurs 15 times.
    05  supr-out-count            pic s9(9) comp.
    05  supr-workspace            pic x(20) value spaces.
```

Status

The supr-status field returns a 0 if the command line was sent to Suprtool without incident or one of the error numbers shown as 88 levels.

Command Line

The supr-command-line can contain any Suprtool command. Use the same format that you use in typing commands into Suprtool. You don't need to enter commands as a single string of 256 characters in a single call to the interface. You may use ";" to send several commands in one string, or you may use the "&" mechanism to continue commands.

The final call must have Exit as the command, alone and left-justified in the command line. The final Exit command can be in uppercase or lowercase, but cannot be abbreviated. MPE commands can be passed into the interface and Suprtool will execute them.

Priority

The supr-priority flag should contain one of "CS", "DS", or "ES".   The default is "DS".

Maxdata

The supr-maxdata contains the Maxdata in words for Suprtool. The default is 32,000, but you can ask for a smaller stack using this parameter (e.g., 20,000). There is no problem with reducing Maxdata when you plan to Sort. However, if you plan to do copies without sorts, you should send Suprtool a Buffer 4096 command if you reduce Maxdata. Otherwise, you are likely to get a Stack Overflow abort within Suprtool or the error message "Unable to allocate buffer in DL area". If you want the default, this field should contain a 0.

Print State

If the supr-print-state contains "AL", the output from Suprtool will always be printed on $stdlist.   If the state is "NE", the output will never be printed. If the state is "ER" or blank, the output will be printed only if Suprtool aborts due to an error.

Total Type

The supr-total-type determines the format of the supr-totals array. If you call Suprtool2 from COBOL, you should use "CO". The COBOL format is display (with leading zeros) and a trailing sign. If the type is "AS", each total is returned left-justified in the total field with a leading sign.

Totals

If you specify the Total command as part of an extract task, Suprtool2 returns the totals in the supr-totals array.   Totals are returned in exactly the same order in which they were specified.   If you are calling Suprtool2 from COBOL, never specify the decimal-precision portion of the Total command.   If your total includes an implied decimal point, you will have to modify the supr-totals declaration to include an implied decimal point (e.g., pic s9(15)v99 ...).

If you specify "AS" as the supr-total-type, each total is formatted as an 18-byte string.   In this case, you should specify the correct decimal-precision in the Total command.   The exponent portion of real totals is truncated by the Suprtool2 interface.

Out Count

After a successful call to the Suprtool2 interface, the supr-out-count is set to the number of Suprtool output records. The supr-out-count is only returned after the Suprtool2 call with the Exit command. Suprtool also puts this count in the SuprtoolOutCount JCW, up to the maximum JCW value permitted, 65,535.

Workspace

The supr-workspace part of the record MUST contain spaces before the first call to the Suprtool2 procedure.

Compiling and Linking on MPE V

You require PH capability in order to use the Suprtool2 interface.   The classic object code for the interface resides in ST2USL.Pub.Robelle.   If your program aborts with a "Stack Overflow" you should increase the size of Maxdata.   Users who call the interface from FORTRAN programs will have to increase the Maxdata size.   COBOL programmers should wait until they get a "Stack Overflow".

The interface is installed in SL files.   The following commands demonstrate how to compile and prepare your program to use the Suprtool interface.   This example assumes that the Suprtool2 routine has been installed in the System SL:

```
     :cobol example.source
     :prep $oldpass,example.pub;cap=ph
     :save example.pub
     :run example.pub
```

Lib=

Using Lib=

The installation chapter of the manual tells how to install the interface so that it is available to every program.   You can also install the interface routine in a SL file and run your program with Lib=.   Here is an example of how to add the Suprtool segment from ST2USL.Pub to your existing SL file.

```
     :run fcopy.pub.sys
     >from=st2usl.pub.robelle;to=st2usl.pub;new
     >exit
     :segmenter
     -buildsl sl.pub,400,4     {fails if you already have an SL}
     -sl sl.pub
     -usl st2usl.pub
     -addsl suprtool
     -exit

     :cobol tool2cob.source
     :prep $oldpass,tool2cob.program;cap=ph
     :save tool2cob.program

     :run tool2cob.program;lib=p
```

The MPE :prep command prints a warning message when you specify cap=ph.   You may ignore this message or specify cap=ia,ba,ph on your :prep command.   If your program requires other capabilities, they should also be specified (e.g., cap=ph,ds).

Warning: The Suprtool2 interface will change over time.   The interface should never be installed directly into a program file.   It should only be installed into SL files.

Compiling and Linking on MPE/iX

You require PH capability in order to use the Suprtool2 interface.   The interface is installed in XL files.   The following commands demonstrate how to compile and link your program to use the native mode Suprtool interface.

```
     :cob85xl example.source
     :link from=$oldpass;to=example.pub;cap=ph
     :run example.pub;xl='st2xl.pub.robelle'
```

User XL File

User XL Files

You can use MPE/iX's Linkedit command to add the Suprtool2 interface to your own XL file. For example,

```
:linkedit
-xl xl.pub
-copyxl from=st2xl.pub.robelle;replace
-exit
:run example.pub;xl='xl.pub'
```

The Replace option is not in all versions of Linkedit.   If your version does not allow Replace, and you already have Suprtool2 in your XL file, you will need to manually purge the existing module before copying the new one:

```
:linkedit
-xl xl.pub
-purgexl module=suprtool.asm
```

Suprtool2 can successfully be installed into the System XL file, but this is not recommended by HP.   Suprtool2 can be combined into an XL file with other Robelle XL files, except Qcompxl (a part of Qedit).

Suprtool Run Parameters

By default, the Suprtool2 interface runs Suprtool as Suprtool.Pub.Robelle with Lib=S. We have provided JCWs that allow Suprtool to be moved or run with a different library.

Program File name
Lib=
Slist234 File

Moving Suprtool

To change the location of the Suprtool program file, you must specify a JCW and use a :File command for the program file:

```
     :setjcw suprtool2filecommand = 2
     :file suprtool = suprtool.pub.dev
     :run example.pub;lib=p
```

We suggest you add the Setjcw and :File commands to a logon UDC.   This way, every invocation of the Suprtool2 interface will use the copy of Suprtool in the Pub group of the Dev account.

Previous versions of the Suprtool2 interface accepted a value of 1 for the suprtool2filecommand JCW.   In this case, you must specify a file command for Suprtool.Pub.Robelle.   We don't recommend you do this, since this only works on MPE/iX and not on MPE V.

Lib=

The Suprtool2 interface looks at the Suprtool2lib JCW for which library to use. The values of this JCW correspond to the description of the library portion of the flag variable of the create intrinsic. These values are:

0	Lib=S
1	Lib=P
2	Lib=G
3	<invalid>

If the Suprtool2lib JCW does not exist, the Suprtool2 interface runs Suprtool with Lib=S. To have the Suprtool2 interface always invoke Suprtool with Lib=P, you would use this command:

```
     :setjcw suprtool2lib = 1
```

Slist234 File

Some previous versions of the Suprtool2 interface created a temporary file called Slist234, which contained the $stdlist from the Suprtool run. A few programs were written to read this file. By default, the Suprtool2 interface no longer creates this file. If your program needs it, use the following JCW to force the interface to create and save the Suprtool listing in the temporary Slist234 file:

```
      :setjcw saveslist234 = 1
```

Lockwords on Suprtool

To tell the Suprtool2 interface that the Suprtool program has a lockword you can use the suprtoolfilecommand facility.

```
file suprtool=suprtool/secret.pub.robelle
setjcw suprtool2filecommand = 2
run prog;lib=p
```

Examples of Calling Suprtool

This chapter contains working examples of source code that calls Suprtool2. You can copy the examples from the manual, but typing them from scratch would be tedious and error-prone.   The best way to copy the examples is to take them from the on-line Suprtool2 documentation file.   The file is stored in Robelle Qedit format.   If you have Qedit, just Text a copy of Suprcall.Doc.Robelle and extract the portions that you are interested in.

If you don't have Qedit, use the Qcopy program to copy the documentation file to a new file in non-Qedit format, then use your favorite text editor to edit this file.

```
:run qcopy.qlib.robelle
>from=suprcall.doc.robelle;to=mysupr2.source;new
>exit
```

Cobol
Fortran
Transact
Spl
Pascal
Speedware

COBOL Example

Below is a sample COBOL program named TOOL2COB. It calls the Suprtool2 interface procedure. The purpose of TOOL2COB is to print selected item master entries from an inventory database, sorted by item number. The program uses Suprtool to create a disc file named SELITEM, filled with the selected item numbers and their descriptions. Then it reads the disc file and prints a report on the line printer.

The database used in this example is called INVORY, and the dataset is called INVREC. Here is a QUERY FORM listing of the dataset:

Sample Database
Example

```
:run query.pub.sys
 >base=invory.data
 PASSWORD = >>
 MODE = >>1
 >form invrec

 SET NAME:
    INVREC,DETAIL

       ITEMS:
          ITEM,                 X8       <<SEARCH ITEM>>
          DESCRIPT,             X50
          WEIGHT,               J2
          STD-PKG,              X12
          SELLPRCE,             J2
          SPUOM,                X2
          SPLEVEL,             3J2
          SPMULTIR,            3J1
          CPMULTIR,             J1
          INDIC,                X2
          VENDORPR,             X6
          SUPPNUM,              X2
          PRCEDATE,             X6

       CAPACITY: 44200           ENTRIES: 35659
```

The listing for the COBOL program is:

```
$control source,uslinit,errors=15,list
$control debug,bounds
 identification division.
 program-id.    tool2cob.
 author.        Robert M. Green.
 date-written.  19 apr83.
 date-compiled.
 installation.  Robelle Solutions Technology Inc.
 security.      Copyright 1983-2001, Robelle Solutions Technology Inc.
 remarks.

************************************************************
*                                                          *
*                        TOOL2COB                          *
*                                                          *
*  FUNCTIONAL OVERVIEW:  uses Suprtool to extract and print*
*                                                          *
*  INPUT:  Invrec set of INVORY database.                  *
*          All items with spec prefix (4 char), sorted.    *
*                                                          *
*  OUTPUT: print report on LINE-PRINTER                    *
*                                                          *
*  TERMINATED BY:  eof or error                            *
*                                                          *
*  PREP WITH:      cap = ph                                *
*                                                          *
*  REV  ,INIT,DATE    , REASON:                            *
*  00,  rmg, 19 Apr83, started first implementation.       *
*  01,  djg, 25 Jan84, modified the control record to      *
*                      include the version number.         *
*  02,  djg, 09 Nov87, modified to include totals in the   *
*                      control record.                     *
*  03,  djg, 10 Aug90, modified for version 4 control rec. *
*                                                          *
************************************************************

$page "environment division"
 environment division.
 configuration section.
 source-computer. hp-3000.
 object-computer. hp-3000.
 special-names.
     top is new-page.

 input-output section.
 file-control.
     select line-printer assign to "LINEPRT,UR,A,LP".
     select selitem assign to "SELITEM".

 data division.
 file section.
 fd  selitem
     data record is item-record.
 01  item-record.
```

```
     05  item-number.
         10  item-prefix         pic x(4).
         10  item-suffix         pic x(4).
     05  item-description        pic x(50).
 fd  line-printer
     data record is line-record.
 01  line-record                 pic x(132).

$page "constants"
 working-storage section.
 01  true-value                  pic x  value "T".
 01  false-value                 pic x  value "F".
 01  revision-no                 pic 99 value 03.

$page "logical variables"
 01  end-of-items-flag           pic x.
     88  end-of-items            value "T".

$page "variables"
 01  if-command.
     05  filler                  pic x(10) value
                "IF  ITEM='".
     05  sel-prefix              pic x(4).
     05  filler                  pic x(2) value "' ".

$page "input area"
 01  accept-buffer.
     05  input-buf               pic x(80).
         88  answer-spaces       value spaces.

$page "report layouts"
 01  report-record.
     05  filler                  pic x(5) value spaces.
     05  report-prefix           pic x(4).
     05  filler                  pic x value "-".
     05  report-suffix           pic x(4).
     05  filler                  pic x(3) value spaces.
     05  report-description      pic x(50).

$page "Suprtool Control Parameter"

* suprtool control area
 01  supr-control.
     05  supr-version              pic s9(4) comp value 4.
     05  supr-status               pic s9(4) comp.
         88  supr-ok               value zeros.
         88  supr-bad-msgfiles          value 1.
         88  supr-aborted               value 2.
         88  supr-create-error          value 3.
         88  supr-bad-total-type        value 4.
     05  supr-command-line         pic x(256) value spaces.
     05  supr-flags.
         10  supr-priority         pic x(2) value spaces.
             88  supr-priority-cs       value "CS".
             88  supr-priority-ds       value "DS".
             88  supr-priority-es       value "ES".
         10  supr-maxdata          pic s9(9) comp value 0.
         10  supr-print-state      pic x(2) value "ER".
             88  supr-print-on-error    value "ER".
             88  supr-print-always      value "AL".
             88  supr-print-never       value "NE".
         10  supr-total-type       pic x(2) value "CO".
             88  supr-total-cobol  value "CO".
             88  supr-total-ascii  value "AS".
         10  supr-other-flags      pic x(18) value spaces.
     05  supr-totals pic s9(17) sign is trailing
                  separate character occurs 15 times.
     05  supr-out-count            pic s9(9) comp.
     05  supr-workspace            pic x(20) value spaces.

$page "procedure division"

************************************************************
*                                                          *
*            P R O C E D U R E   D I V I S I O N           *
*                                                          *
************************************************************

 procedure division.
 00-main                         section.

* Ask for selection criteria.

     display "TOOL2COB ", revision-no.
     display " ".

     display "Enter 4-character item prefix to select:".

     move spaces to input-buf.
     accept input-buf.
     if answer-spaces then go to 00-main-exit.

     move input-buf to sel-prefix.

* Use suprtool to build the extract file.

     move "AL" to supr-print-state.

     move "base invory.data,5,dev" to supr-command-line.
                                  perform 01-call-suprtool.
     move "get invrec"            to supr-command-line.
                                  perform 01-call-suprtool.
     move if-command              to supr-command-line.
                                  perform 01-call-suprtool.
     move ":purge selitem"        to supr-command-line.
                                  perform 01-call-suprtool.
     move "output selitem"        to supr-command-line.
                                  perform 01-call-suprtool.
     move "extract item,descript" to supr-command-line.
                                  perform 01-call-suprtool.
     move "sort item"             to supr-command-line.
                                  perform 01-call-suprtool.
     move "exit"                  to supr-command-line.
                                  perform 01-call-suprtool.
```

```
* At this point, the file "SELITEM" contains the sorted
* data. This file must be read and printed on the line-
* printer.

     open output line-printer.
     open input selitem.
     move false-value to end-of-items-flag.

     perform 02-print-item
       until end-of-items.

     close selitem.
     close line-printer.

 00-main-exit. goback.

 01-call-suprtool.
     call "SUPRTOOL2" using supr-control.
     if not supr-ok then
         display "Error:  Unable to sort the invrec dataset"
         display " "
         display "Suprtool interface error number: ",
                 supr-status.

 02-print-item.

     read selitem at end
       move true-value to end-of-items-flag.

     if not end-of-items then
       move item-prefix       to report-prefix
       move item-suffix       to report-suffix
       move item-description  to report-description
       move report-record to line-record
       write line-record after advancing 1 lines.
```

FORTRAN Example

FORTRAN programmers may use the Suprtool2 interface to invoke Suprtool from within a FORTRAN program. The control area may be in COMMON or it may be local to one routine. If the control area is declared local to one routine, then all calls to Suprtool2 must be from within the same routine. The following example declares the control area in a common block. One routine initializes the Suprtool2 control area. The second routine is repeatedly passed command lines which copy the first 100 lines of catalog.pub.sys to the temporary file "tempfile". This example program is compatible with FORTRAN 66, FORTRAN 77/V, and FORTRAN 77/iX.

FORTRAN 66

FORTRAN 66 users may need to increase the size of Maxdata for your program in order to use the interface. The $HP3000_16 ALIGNMENT$ option is not needed by FORTRAN 66 and if it is not deleted a warning is produced by the compiler. FORTRAN 66 produces a warning on the line `call setjcw`, but this warning may be safely ignored.

FORTRAN 77/V and FORTRAN 77/iX

The example program is valid for FORTRAN 77/V and FORTRAN 77/iX, but you must use the $HP3000_16 ALIGNMENT$ compiler option with FORTRAN 77/iX. The $HP3000_16 ALIGNMENT$ compiler option is required for FORTRAN 77/iX, but the option is not needed for FORTRAN 77/V. If the option is not deleted a warning is produced by the FORTRAN 77/V compiler. In FORTRAN 77, the `suprcommand` variable may be declared as `character*256` eliminating the need for the `suprfiller`.

Example

```
$HP3000_16 ALIGNMENT$
      program test
      character*254 command
      call initsuprcontrol
      command = "input catalog.pub.sys"
      call invokesuprtool(command)
      command = "numrecs 100"
      call invokesuprtool(command)
      command = "output tempfile,temp"
      call invokesuprtool(command)
      command = "exit"
      call invokesuprtool(command)
20    stop
      end
C  This subroutine initializes the control record for SUPRTOOL.
C  It should be called once at the beginning of the program.
C
      subroutine initsuprcontrol
      common /suprtool/ suprcontrol
      integer*2 suprcontrol(291)
C
      integer*2     suprversion
      integer*2     suprstatus
      character*254 suprcommand
      character*2   suprfiller
      character*2   suprpriority
      integer*4     suprmaxdata
      character*2   suprprintstate
      character*18  suprotherflags
      character*2   suprtotaltype
      character*18  suprtotals(15)
      integer*4     suproutcount
      character*20  suprworkspace
C
      equivalence(suprversion   ,suprcontrol(1))
      equivalence(suprstatus    ,suprcontrol(2))
      equivalence(suprcommand   ,suprcontrol(3))
      equivalence(suprfiller    ,suprcontrol(130))
      equivalence(suprpriority  ,suprcontrol(131))
      equivalence(suprmaxdata   ,suprcontrol(132))
      equivalence(suprprintstate,suprcontrol(134))
      equivalence(suprtotaltype ,suprcontrol(135))
      equivalence(suprotherflags,suprcontrol(136))
      equivalence(suprtotals    ,suprcontrol(145))
      equivalence(suproutcount  ,suprcontrol(280))
      equivalence(suprworkspace ,suprcontrol(282))
C
      suprversion   = 4
      suprstatus    = 0
      suprcommand   = " "
      suprfiller    = "  "
      suprpriority  = "  "
      suprmaxdata   = 0
      suprprintstate= "ER"
      suprtotaltype = "AS"
      suprworkspace = " "
C
```

```
      return
      end
C  Call the "invokesuprtool" subroutine for each command line
C  that must be passed to SUPRTOOL.
C
      subroutine invokesuprtool(command)
      character*254 command
      common /suprtool/ suprcontrol
      integer*2 suprcontrol(291)
C
      integer*2     suprversion
      integer*2     suprstatus
      character*254 suprcommand
      character*2   suprfiller
      character*2   suprpriority
      integer*4     suprmaxdata
      character*2   suprprintstate
      character*18  suprotherflags
      character*2   suprtotaltype
      character*18  suprtotals(15)
      integer*4     suproutcount
      character*20  suprworkspace
C
      equivalence(suprversion   ,suprcontrol(1))
      equivalence(suprstatus    ,suprcontrol(2))
      equivalence(suprcommand   ,suprcontrol(3))
      equivalence(suprfiller    ,suprcontrol(130))
      equivalence(suprpriority  ,suprcontrol(131))
      equivalence(suprmaxdata   ,suprcontrol(132))
      equivalence(suprprintstate,suprcontrol(134))
      equivalence(suprtotaltype ,suprcontrol(135))
      equivalence(suprotherflags,suprcontrol(136))
      equivalence(suprtotals    ,suprcontrol(145))
      equivalence(suproutcount  ,suprcontrol(280))
      equivalence(suprworkspace ,suprcontrol(282))
C
      system intrinsic setjcw
C
      if(suprstatus.ne.0) goto 900
      suprcommand = command
      call suprtool2(suprcontrol)
      if (suprstatus.eq.0) goto 900
C     display "SUPRTOOL interface failed with command:"
C     display command
      call setjcw(-1)
900   return
      end
```

TRANSACT Example

To call the Suprtool2 interface from a TRANSACT program, you must first define the Suprtool2 control record. We suggest that the control record be included from the file Transact.Qlibsrc.Robelle. The TRANSACT definition of the control record, with the proper initializing values, is as follows:

Control Record
Example

```
define(item)

 supr-control          x(582),init  = " ":
   supr-version        i(4,0,2)     = supr-control(1):
   supr-status         i(4,0,2)     = supr-control(3):
   supr-command        x(256)       = supr-control(5):
   supr-pri            x(2)         = supr-control(261):
   supr-maxdata        i(9,0,4)     = supr-control(263):
   supr-print          x(2)         = supr-control(267):
   supr-tot-type       x(2)         = supr-control(269):
   supr-other          x(18)        = supr-control(271):
   supr-totals      15 x(18)        = supr-control(289):
     supr-total        9(17)        = supr-totals(1):
     supr-tot-sign     x(1)         = supr-totals(18):
   supr-out-count      i(9,0,4)     = supr-control(559):
   supr-workspace      x(20)        = supr-control(563);


 list supr-control;

 << set up initial values >>

 let  (supr-version)  = 4;   << don't change >>
 let  (supr-status)   = 0;   << check after each call >>
 move (supr-command)  = " ";
 move (supr-pri)      = " "; << "CS","DS","ES"  >>
                             << "DS" is default >>
 let  (supr-maxdata)  = 0;
 move (supr-print)    = "ER";<< print on error only >>
                             << "NE" - never,       >>
                             << "AL" - always       >>
 move (supr-tot-type) = "CO";<< total type    >>
                             << "CO" - cobol, >>
                             << "AS" - ascii  >>
 move (supr-other)    = " ";
 move (supr-out-count)= 0;
 move (supr-workspace)= " ";
```

Below is a sample program that uses the TRANSACT include file to call the Suprtool2 interface. The program uses Suprtool to find two totals and the output count of selected records from the sales-hist dataset of the SH database. Be careful not to create commands greater than 256 characters in length.

```
SYSTEM SUPREX;

SET(OPTION) NOHEAD;
DEFINE(ITEM)
       TOTAL1       I (9,0,4),  EDIT = "ZZZZZZZZ^":
       TOTAL2       I (9,0,4),  EDIT = "ZZZZZZZZ^":
       CUSTNO       X (6),      INIT = "000000";

LIST   TOTAL1,
       TOTAL2,
       CUSTNO;

!INCLUDE(TRANSACT.QLIBSRC.ROBELLE)

DATA(SET)  CUSTNO;

MOVE (SUPR-COMMAND)           = "BASE SH,5,READ";
PERFORM CALL-SUPRTOOL2;

MOVE (SUPR-COMMAND)           = "GET SALES-HIST";
PERFORM CALL-SUPRTOOL2;

MOVE (SUPR-COMMAND)           = "DEF CUST-NO,HIST-KEY[1],6";
PERFORM CALL-SUPRTOOL2;

MOVE (SUPR-COMMAND)           = "IF CUST-NO = '" +
                                    (CUSTNO) + "'";
PERFORM CALL-SUPRTOOL2;

MOVE (SUPR-COMMAND)           = "TOTAL CU-PER1-QTY";
PERFORM CALL-SUPRTOOL2;

MOVE (SUPR-COMMAND)           = "TOTAL CU-PER2-QTY";
PERFORM CALL-SUPRTOOL2;

MOVE (SUPR-COMMAND)           = "OUTPUT $NULL";
PERFORM CALL-SUPRTOOL2;

MOVE (SUPR-COMMAND)           = "EXIT";
PERFORM CALL-SUPRTOOL2;

IF (SUPR-TOT-SIGN(1)) = "-" THEN
   LET (TOTAL1)               = (SUPR-TOTAL(1)) * -1
ELSE
   LET (TOTAL1)               = (SUPR-TOTAL(1));

IF (SUPR-TOT-SIGN(2)) = "-" THEN
   LET (TOTAL2)               = (SUPR-TOTAL(2)) * -1
ELSE
   LET (TOTAL2)               = (SUPR-TOTAL(2));
```

```
DISPLAY
  "There are ": SUPR-OUT-COUNT, EDIT="ZZZZZZZZ^":
  "records for customer ": CUSTNO:
  "Period 1 Qty this year", COL=5, LINE: TOTAL1:
  "Last year ": TOTAL2;

EXIT;

CALL-SUPRTOOL2:

  PROC SUPRTOOL2((SUPR-CONTROL));
  IF (SUPR-STATUS) <> 0 THEN
     DISPLAY "SUPRTOOL2 INTERFACE ERROR   ", LINE = 2:
               SUPR-STATUS, JOIN = 1:
             "ON COMMAND:",LINE:
              SUPR-COMMAND;
  RETURN;

END SUPREX;
```

SPL Example

This SPL program invokes Suprtool through the interface routine:

```
$control errors=5,main=tool2spl
begin
<<
Copyright 1982-2001 Robelle Solutions Technology Inc.

    name:        tool2spl
    purpose:     test the suprtool interface.
    prep:        cap = ph
    run:         run with parm=2 for print stdlist always.
                 run with parm=3 to print stdlist never.
    programmer:  Robert Green and David Greer
>>
define current'version = "(Version 0.6)"#;
$page "Global variables of the tool2spl program"

  << suprtool2 interface layout >>

  equate
     wl'supr'control      = 290
    ,wl'supr'command'line = 128
    ,wl'supr'totals       = 9
    ,bl'supr'totals       = wl'supr'totals * 2
    ,max'supr'totals      = 15
    ;

  integer array supr'control(0:wl'supr'control);
  integer array supr'version(*)      = supr'control(000);
  integer array supr'status(*)       = supr'control(001);
  integer array supr'command'line(*) = supr'control(002);
     byte array b'command'line(*)      = supr'command'line;
  integer array supr'flags(*)        = supr'control(130);
     byte array supr'priority(*)       = supr'flags(0);
   double array supr'maxdata(*)        = supr'flags(1);
     byte array supr'print'state(*)    = supr'flags(3);
     byte array supr'total'type(*)     = supr'flags(4);
  integer array supr'totals(*)       = supr'control(144);
   double array supr'out'count(*)    = supr'control(279);
  integer array supr'workspace(*)    = supr'control(281);

  equate supr'ok          = 0
        ,bad'msgfiles     = 1
        ,suprtool'aborted = 2
        ,create'error     = 3
        ;

  << standard global variables for Robelle SPL programs >>

integer
   copy'of'parm     << copy of "ACTUAL'PARM" >>
  ,actual'parm=q-4  << run ;parm=nn (use in mainline) >>
  ,input'length     << used only with input defines >>
  ;
logical
   batch            << run in batch mode >>
  ;
equate
   rtn = 13  << carriage return in ascii >>
  ,bell = 7  << ring the bell    in ascii >>
  ,bl'inbuf = 86
  ,bl'outbuf = 132
  ,wl'outbuf = bl'outbuf/2
  ,wl'inbuf = bl'inbuf/2
  ;
define
   end'if   = end#  ,end'else = end#  ,end'while= end#
  ,end'case = end#  ,end'do   = end#  ,end'proc = end#
  ,end'subr = end#
  ,p = begin move outbuf := #
  ,output=,2;output'(*);end#,err=,2;err'(*);end#
  ,ask=,2;ask'(*);end#, warn =,2;warn'(*);end#
  ,trim = while input'length>0 and
           inbuf'(input'length-1)=" " do
       input'length:=input'length-1#
  ,echo = if batch then
     begin trim; print(inbuf,-input'length,0); end#
  ,input = begin input'length := readx(inbuf,-bl'inbuf);
                 if > then quit(10); echo;
                 inbuf'(input'length):= rtn;
           end#
  ;

integer array inbuf (0:wl'inbuf);
  byte array inbuf'(*)=inbuf;

integer array outbuf  (0:wl'outbuf);
  byte array outbuf'(*) = outbuf;

intrinsic
   ascii, aritrap, binary, calendar, clock, dascii,
   dateline, dbinary, debug, genmessage, print, quit, readx,
   terminate, who, xaritrap, xcontrap, xlibtrap, xsystrap,
   setjcw;

  << external definition of suprtool2 routine >>
    procedure suprtool2(supr'definition);
       array supr'definition;
       option external, check 3;

$page "Standard Robelle Utility Procedures"

procedure output'( address ) ;
  value address; integer address; << in outbuf >>
begin   integer output'length;
  << check length for overflow of buf >>
  output'length := (address-@outbuf)*2;
  if output'length > bl'outbuf then begin
    print(outbuf,-bl'outbuf,0);
    quit(111);
    end'if;
  print(outbuf,-output'length,0);
  end'proc; <<output'>>

procedure ask' (address);
  value address; integer address;
begin   integer output'length;
  << check length for overflow of buf >>
  output'length := (address-@outbuf)*2;
  if output'length > bl'outbuf then begin
    print(outbuf,-bl'outbuf,0);
    quit(111);
    end'if;
  print(outbuf,-output'length,%320);
  end'proc; <<ask'>>

procedure err'(address);
  value address; integer address; << in outbuf >>
begin   integer output'length;
  integer array errbuf(0:10);
  move errbuf:=([8/bell ,8/10]<<lf>>,"ERROR:  ");
  print(errbuf,5,%320);
  << check length for overflow of buf >>
  output'length := (address-@outbuf)*2;
  if output'length > bl'outbuf then begin
    print(outbuf,-bl'outbuf,0);
    quit(111);
    end'if;
  print(outbuf,-output'length,0);
  print(errbuf,0,0);
  end'proc; <<err'>>

procedure blank(buf,count);
  value count;
  integer count;
  integer array buf;
begin
  if count > 0 then begin
    buf:="  ";
    if (count:=count-1)>0 then
      move buf(1):=buf,(count);
    end'if;
  end'proc; <<blank>>

procedure b'blank(buf,count);
  value count;
  integer count;
  byte array buf;
begin
  if count > 0 then begin
    buf:=" ";
    if (count:=count-1)>0 then
      move buf(1):=buf,(count);
    end'if;
  end'proc; <<b'blank>>
```

```
$page "level 1:  print'totals"
<<  Print the Suprtool2 totals.
>>

procedure print'totals;
begin
   byte array
      blank'total(0:bl'supr'totals)
     ;
   byte pointer
      supr'totals'
     ;
   integer
      total'index
     ,total'offset
     ;
   logical
      first'time
     ;

   b'blank(blank'total,bl'supr'totals);
   first'time  := true;
   total'index := 0;

   while total'index < max'supr'totals do
   begin
      total'offset := total'index * wl'supr'totals;
      @supr'totals':= @supr'totals(total'offset) & lsl(1);
      if supr'totals' <> blank'total,(bl'supr'totals) then
      begin
         if first'time then
         begin
            p "Totals:" output;
            first'time := false;
         end'if;
         print(supr'totals(total'offset),wl'supr'totals,0);
      end'if;
      total'index := total'index + 1;
   end'while;

end'proc;   <<print'totals>>
$page "call'suprtool2"
<<
  pass a command line to suprtool
>>

procedure call'suprtool2;
begin
  integer length;

  subroutine print'int(val);
    value val; integer val;
  begin
    val := ascii(val,10,outbuf');
    print(outbuf,-val,0);
    end'subr; <<print'int>>
  subroutine print'command;
  begin
    length := 256;
    do length:=length-1 until
    length<0 or b'command'line(length)<>" ";
    length:=length+1;
    print(supr'command'line,-length,0);
    end'subr; <<print'command>>

  suprtool2 (supr'control);
  if supr'status <> supr'ok then begin
    p "Suprtool2 interface call has failed!" err;
    print'command;
    p "Error number = " ask;
    print'int(supr'status);
    setjcw(-1);
    end'if;
  blank(supr'command'line,wl'supr'command'line);

  end'proc; <<call'suprtool2>>
$page "init'tool2spl"
<<
    check logon mode, parm values, etc.
    see if we can send one command to suprtool.
>>

logical procedure init'tool2spl;
begin
  integer temp;

  init'tool2spl := false;

  who(temp);
  batch := temp.(12:2) = 2;

  p "TOOL2SPL/Copyright Robelle 1982-2001"output;
  p current'version output;
  print(outbuf,0,0); <<blank line>>

  blank(supr'control,wl'supr'control);
  supr'version := 4;
  supr'maxdata := 0d;
  move supr'print'state := "ER";
  move supr'total'type  := "AS";
  if copy'of'parm=2 then
    move supr'print'state := "AL"
  else
  if copy'of'parm=3 then
    move supr'print'state := "NE";
  p "Please hit RETURN for EXIT at end. Thanks" output;
  move supr'command'line :=
    ":COMMENT Test of Suprtool Interface";
  call'suprtool2;
  if supr'status=0 then begin
    init'tool2spl := true;
    print(outbuf,0,0);
    end'if;
  end'proc;  <<init'tool2spl>>
$page "mainline"

  copy'of'parm := actual'parm;

  if init'tool2spl then begin
    << main loop of program >>
    do begin
      p ">" ask;
      input;
      if input'length<>0 then begin
        move b'command'line:=inbuf',(input'length);
        call'suprtool2;
        end'if;
      end'do
    until input'length=0 or supr'status<>0;
    move supr'command'line := "EXIT";
    call'suprtool2;
    if supr'status = 0 then
      print'totals;
    end'if;

end.  <<tool2spl>>
```

Pascal Example

Below is a Pascal program to invoke Suprtool through the interface routine. The example as given is for use with Pascal/iX. Note that the $HP3000_16$ compiler option is set. This is required because Suprtool2's control record uses 16-bit alignment instead of Pascal/iX's default 32-bit alignment.

For Pascal/V, change the $set nm$ line to $set 'nm=false'$

```
{ Example of Calling Suprtool2 from Pascal }

{ Set 'nm=false' for MPE V }
$set 'nm=true'$
$list off$
$if 'nm'$
  $HP3000_16$
$endif$

program test(input,output);

const
   MaxSuprCmdLen = 256;

type

$if 'not nm'$
   shortint = -32768 ..  32767;        (* 16-bit integer *)
$endif$

   SuprCmdStr   = string[MaxSuprCmdLen];
   SuprTotalStr = packed array[1..18] of char;

   SuprControlRec = record
      version    : shortint;
      status     : shortint;
      command    : packed array[1..MaxSuprCmdLen] of char;
      priority   : packed array[1..2] of char;
      maxdata    : integer;
      print_state: packed array[1..2] of char;
      total_type : packed array[1..2] of char;
      other_flags: packed array[1..18] of char;
      totals     : array[1..15] of SuprTotalStr;
      out_count  : integer;
      workspace  : packed array [1..20] of char;
   end;

var
   SuprControl : SuprControlRec;

procedure suprtool2( var control: SuprControlRec );
   external spl;

procedure InitST2;
var
```

```
   i: integer;
begin
   SuprControl.version        := 4;
   SuprControl.status         := 0;
   SuprControl.priority[1]    := 'C';
   SuprControl.priority[2]    := 'S';
   SuprControl.maxdata        := 0;
   SuprControl.print_state[1] := 'A';
   SuprControl.print_state[2] := 'L';
   SuprControl.total_type[1]  := 'A';
   SuprControl.total_type[2]  := 'S';
   for i:=1 to 20 do
      SuprControl.workspace[i] := ' ';
   for i:=1 to MaxSuprCmdLen do
      SuprControl.command[i] := ' ';
end;

procedure CallSuprtool2;
begin
  suprtool2( SuprControl );
  if (SuprControl.status<>0) then
  begin
    writeln('Suprtool2 interface call has failed!');
    writeln('Error number = ', SuprControl.status:1 );
  end;
end;

procedure ST2Cmd( cmd: SuprCmdStr );
var i: integer;
begin
  for i:=1 to MaxSuprCmdLen do
    SuprControl.command[i]:=' ';
  for i:=1 to strlen( cmd ) do
    SuprControl.command[i]:=cmd[i];
  CallSuprtool2;
end;


begin
  InitST2;
  ST2Cmd( 'in infile;out $null');
  ST2Cmd( 'exit');
  writeln( SuprControl.out_count );
end.
```

SPEEDWARE Example

You can call the Suprtool2 interface from a SPEEDWARE V.6 or V.7 program.   Use the Supr-Control area to pass commands to Suprtool2.

There are 3 ways to get Speedware to find the Suprtool2 routine.

1. **Using Designer.**   In the Operating Options (OPEO) screen add the XL to Local 4GL Parms.   : XL="ST2XL.PUB.ROBELLE"

   This is the recommended approach for SPEEDWARE V.6.

2. **Using Linkedit.**   Copy the routines into SPWXL.SPWxxxxx.SPEEDWRE.   This is the XL that Speedware provides for this purpose.

   This could become a problem as SUPRTOOL2 changes over the years and the new version is not updated into SPWXL.

3. **Using RUN.**   Add the XL to the run command.   n.b.   If you do this you will also need to include Speedware's database interface XL called DBMSXL.

   ```
   :RUN SPW4GL.SPWxxxxx.SPEEDWRE
      ;XL="DBMSXL.SPWxxxxx.SPEEDWRE,ST2XL.PUB.ROBELLE"
   ```

   This is the recommended approach for SPEEDWARE V.7 and higher.

Example

Below is a sample SPEEDWARE process that calls the Suprtool2 interface.   The program uses Suprtool to read a dataset, then writes the dataset to a file.

```
GLOBAL:
SET TITLE("Calling Suprtool Test");
EXIT;

MENU:
SET WINDOW(ROW=4,COL=4,HEIGHT=15,WIDTH=40);
BRANCH "Copy D-SALES to file"  DO REPORT-SUPRTOOL2;
EXIT;

REPORT-SUPRTOOL2:LISTING;
SET TITLE("Calling Suprtool2");
SET WINDOW;
DCL
  SUPR-CONTROL      FORMAT (TYPE = CHAR, BYTES =582),
  SUPR-VERSION      FORMAT (TYPE = INT , BYTES = 2, LEN= 4)
                    REDEFINES (SUPR-CONTROL[1:2]),
  SUPR-STATUS       FORMAT (TYPE = INT , BYTES = 2, LEN= 4)
                    REDEFINES (SUPR-CONTROL[3:4]),
  SUPR-COMMAND-LINE FORMAT (TYPE = CHAR, BYTES = 256)
                    REDEFINES (SUPR-CONTROL[5:260]),
  SUPR-PRIORITY     FORMAT (TYPE = CHAR, BYTES = 2)
                    REDEFINES (SUPR-CONTROL[261:262]),
  SUPR-MAXDATA      FORMAT (TYPE = INT , BYTES = 4, LEN= 9)
                    REDEFINES (SUPR-CONTROL[263:266]),
  SUPR-PRINT-STATE  FORMAT (TYPE = CHAR, BYTES = 2)
                    REDEFINES (SUPR-CONTROL[267:268]),
  SUPR-TOTAL-TYPE   FORMAT (TYPE = CHAR, BYTES = 2)
                    REDEFINES (SUPR-CONTROL[269:270]),
  SUPR-FUTURE-FLAGS FORMAT (TYPE = CHAR, BYTES = 18)
                    REDEFINES (SUPR-CONTROL[271:288]),
  SUPR-TOTALS  (15) FORMAT (TYPE = ZONE, BYTES = 18,)
                    REDEFINES (SUPR-CONTROL[289:558]),
  SUPR-OUT-COUNT    FORMAT (TYPE = INT , BYTES = 4, LEN= 9)
                    REDEFINES (SUPR-CONTROL[559:562]),
  SUPR-WORKSPACE    FORMAT (TYPE = CHAR, BYTES = 20)
                    REDEFINES (SUPR-CONTROL[563:582]);

(* initial values *)

CALCUL 4                      = SUPR-VERSION,
       0                      = SUPR-STATUS,
       'DS'                   = SUPR-PRIORITY,
       'ER'                   = SUPR-PRINT-STATE,
       'AS'                   = SUPR-TOTAL-TYPE,
       ' '                    = SUPR-WORKSPACE;

CALCUL 'base store,1,READER'  = SUPR-COMMAND-LINE;

CALL SUPRTOOL2            USING SUPR-CONTROL MODIFY;

IF SUPR-STATUS <> 0  THEN
   PRINT 'Error Suprtool', SUPR-STATUS, SUPR-COMMAND-LINE;
ELSE
```

```
BEGIN
   CALCUL 'get d-sales;out dsfile' = SUPR-COMMAND-LINE;

   CALL SUPRTOOL2         USING SUPR-CONTROL MODIFY;

END;

IF SUPR-STATUS <> 0  THEN
   PRINT 'Error Suprtool', SUPR-STATUS, SUPR-COMMAND-LINE;
ELSE
BEGIN
   CALCUL 'exit '                  = SUPR-COMMAND-LINE;

   CALL SUPRTOOL2         USING SUPR-CONTROL MODIFY;

END;
EXIT;
```

Calling Suprlink

User programs may "call" Suprlink via the Suprtool2 interface using the Suprtool Link command.   A typical use of this interface would be for a COBOL program to ask Suprtool to extract selected subsets from two or more large IMAGE datasets and to use Suprlink to link the two files.   The COBOL program would then read and format the Suprlink output file into a report.

Restrictions
Example
Sample Database
Cobol Program

Restrictions

When calling the Suprtool2 interface, you signal the end of commands and the start of processing by passing the Exit command. If you specify the Link command with no parameters, the interface switches to Suprlink. When you have passed all the Suprlink commands, do not end the Suprlink task by passing Exit. This will end the Suprlink task, but it will also result in a Suprtool error. There are two solutions to this problem:

1. Specify a substring of Exit (for example, E).

2. Specify all Suprlink commands with Suprtool's Link command (e.g., Link Exit).

COBOL Example

Below is a sample COBOL program named LINK2COB, which calls the Suprtool2 interface procedure.   Its purpose is to print the notes for a selected product with the associated company name sorted by serial number.   The program uses the Suprtool2 interface to do the following:

1. Select and sort records from the notes dataset.

2. Extract specific fields and sort the entire Customer dataset.

3. Use Suprlink to link the Notes and Customer files into the Notecust file.

The program then reads the Notecust file and prints a report on the line printer.

The database used in this example is called Cust, and the datasets are called Notes and Customer.   Here is a Suprtool Form listing of the datasets:

```
:run suprtool.pub.robelle
>ba cust.history
>form notes
    NOTES            Detail
       Entry:
          SERIAL-NO            U4     (!CUSTOMER(SORT-ITEM))
          PRODUCT-CODE         U2
          TEXT-LINE            X80
          SORT-ITEM            K2
    Capacity: 75010  Entries: 40459  Entry Length: 45
```

Customer

```
>form customer
    CUSTOMER         Master
       Entry:
          SERIAL-NO            U4     <<Search-Field>>
          COMPANY-NAME         X60
          ALTERNATE-NAME       X60
          CONTACT-NAME         X40
          CONTACT-TITLE        X40
          PHONE-NO             X20
          CONTACT-NOTE         X40
          ADDRESS             4X40
          COUNTRY              X12
          CUSTOMER-NOTE        X40
          RELATED-NAME         X60
          SORTING-SEQ          U16
          OMNIDEX-SI           J2
          TIMESTAMP-INIT       U16
    Capacity: 1499  Entries: 1250  Entry Length: 286
```

The listing for the COBOL program is:

```
$control source,errors=15,nolist
$control debug,bounds
 identification division.
 program-id.    Link2cob.
 author.        David J. Greer
 date-written.  15 Sep88.
 date-compiled.
 installation.  Robelle Solutions Technology Inc.
 security.      Copyright 1988-2001 Robelle Solutions Technology Inc.
 remarks.

************************************************************
*                                                          *
*                     link2cob                             *
*                                                          *
*  functional overview:  Uses Suprtool to extract          *
*                        and print.                        *
*  input:  Notes dataset of the cust database.             *
*          All notes with a specified product              *
*          code and the related customer name              *
*          from the customer dataset.                      *
*                                                          *
*  output: print report on line-printer                    *
*                                                          *
*  terminated by:  eof or error                            *
*                                                          *
*  prep with:      cap = ph;maxdata=15000                  *
*                                                          *
*  rev#,init,date    , reason:                             *
*  01  , djg,15 Sep88, started first implementation.       *
*  02  , djg,10 Aug90, version 4 of the control record.    *
*                                                          *
************************************************************

$page "environment division"
 environment division.
 configuration section.
 source-computer.  hp-3000.
 object-computer.  hp-3000.
 special-names.
     top is new-page.

 input-output section.
 file-control.
     select line-printer assign to "LINEPRT,,,LP".
     select notecust-file assign to "NOTECUST".

 data division.
 file section.
 fd  notecust-file
     data record is item-record.
 01  notecust-record.
     05  notecust-serial-no      pic x(4).
     05  notecust-product-code   pic x(2).
     05  notecust-text-line      pic x(80).
     05  notecust-company-name   pic x(60).

 fd  line-printer
     data record is line-record.
 01  line-record                 pic x(132).

$page "constants"
 working-storage section.
 01  true-value                  pic x value "T".
 01  false-value                 pic x value "F".
 01  revision-no                 pic 99            value 2.

$page "logical variables"
 01  end-of-notecust-flag        pic x.
     88  end-of-notecust         value "T".

$page "variables"
 01  if-command.
     05  filler                  pic x(17) value
                "IF PRODUCT-CODE='".
     05  if-product-code         pic x(2).
     05  filler                  pic x(2) value "'".

$page "input area"
 01  accept-buffer.
     05  input-buf               pic x(80).
         88  answer-spaces       value spaces.

$page "report layouts"
 01  report-header.
     05  filler                  pic x(9) value "Serial      : ".
      05  head-serial-no          pic x(4).
     05  filler                  pic x(2) value spaces.
     05  head-company-name       pic x(60).

 01  report-detail.
     05  rept-text-line          pic x(80).

$page "suprtool control parameter"

$include cobol.qlibsrc.robelle

$page "procedure division"

************************************************************
*                                                          *
*          p r o c e d u r e    d i v i s i o n            *
*                                                          *
************************************************************

 procedure division.
 00-main                         section.

* ask for selection criteria.

     display "LINK2COB ", revision-no.
     display " ".
     display "Enter 2-character product code:".

     move spaces to input-buf.
     accept input-buf.
     if answer-spaces then go to 00-main-exit.

     move input-buf to if-product-code.

      perform 10-create-notecust-file
         thru 10-create-notecust-file-exit.

* Now, the file "notecust" contains the sorted data.
* This file must be read and printed on the line printer.

      perform 20-produce-report
         thru 20-produce-report-exit.

 00-main-exit.  goback.

$page "[10]  create-notecust-file"

 10-create-notecust-file       section.

     move "base cust.history,5" to supr-command-line.
                                perform 10-90-call-suprtool.

     perform 10-10-create-notefile.

     perform 10-20-create-custfile.

     perform 10-30-link-notefile-custfile.

     move "exit"                to supr-command-line.
                                perform 10-90-call-suprtool.

     go to 10-create-notecust-file-exit.

 10-10-create-notefile.
     move "get notes"           to supr-command-line.
                                perform 10-90-call-suprtool.
     move if-command            to supr-command-line.
                                perform 10-90-call-suprtool.
     move "extract serial-no"   to supr-command-line.
                                perform 10-90-call-suprtool.
     move "extract product-code" to supr-command-line.
                                perform 10-90-call-suprtool.
     move "extract text-line"   to supr-command-line.
                                perform 10-90-call-suprtool.
     move ":purge notefile"     to supr-command-line.
                                perform 10-90-call-suprtool.
     move "output notefile,link"  to supr-command-line.
                                perform 10-90-call-suprtool.
     move "sort serial-no"      to supr-command-line.
                                perform 10-90-call-suprtool.
     move "sort sort-item,desc" to supr-command-line.
                                perform 10-90-call-suprtool.
     move "xeq"                 to supr-command-line.
                                perform 10-90-call-suprtool.
 10-20-create-custfile.
     move "get customer"        to supr-command-line.
                                perform 10-90-call-suprtool.
     move ":purge custfile"     to supr-command-line.
                                perform 10-90-call-suprtool.
     move "output custfile,link"  to supr-command-line.
                                perform 10-90-call-suprtool.
     move "sort serial-no"      to supr-command-line.
                                perform 10-90-call-suprtool.
     move "extract serial-no"   to supr-command-line.
                                perform 10-90-call-suprtool.
     move "extract company-name" to supr-command-line.
                                perform 10-90-call-suprtool.
     move "xeq"                 to supr-command-line.
                                perform 10-90-call-suprtool.

 10-30-link-notefile-custfile.
     move "link"                to supr-command-line.
                                perform 10-90-call-suprtool.
     move "input notefile by serial-no"
                                to supr-command-line.
                                perform 10-90-call-suprtool.
     move "link custfile"       to supr-command-line.
                                perform 10-90-call-suprtool.
     move ":purge notecust"     to supr-command-line.
                                perform 10-90-call-suprtool.
     move "output notecust"     to supr-command-line.
                                perform 10-90-call-suprtool.
     move "e"                   to supr-command-line.
                                perform 10-90-call-suprtool.

 10-90-call-suprtool.
     call "SUPRTOOL2" using supr-control.
     if not supr-ok then
         display "Error:  Unable to call Suprtool2"
         display " "
         display "Suprtool interface error number: ",
                 supr-status.

 10-create-notecust-file-exit.  exit.

$page "[20]  produce-report"
*
* Read the notecust file which contains the detail records
* from the NOTES dataset and the company-name from the
* CUSTOMER dataset.  We produce a break on each new customer
* serial number; otherwise, we just print the text of the
* note lines.
*

 20-produce-report               section.

     open input  notecust-file.
     open output line-printer.
```

```
    move spaces to head-serial-no.
    move false-value to end-of-notecust-flag.

    perform 20-10-read-notecust
      until end-of-notecust.

    go to 20-produce-report-exit.

20-10-read-notecust.
    read notecust-file at end
        move true-value to end-of-notecust-flag.
    if not end-of-notecust then
       perform 20-20-process-record.

20-20-process-record.
    if notecust-serial-no <> head-serial-no then
       perform 20-30-print-header.
    move notecust-text-line to rept-text-line.
    write line-record from report-detail.

20-30-print-header.
    move spaces to line-record.
    write line-record.
    move notecust-serial-no to head-serial-no.
    move notecust-company-name to head-company-name.
    write line-record from report-header.

20-produce-report-exit.  exit.
```

Installing the Suprtool2 Interface

You do not need to do any installation steps in order to use the native mode version of Suprtool2.   These instructions describe how to install the Suprtool2 interface into the System SL on Classic versions of MPE, and on MPE/iX for access by compatibility mode programs.

The job stream Suprcall.Suprjob.Robelle installs the Suprtool2 interface so that it can be used by any program on the system.   Use it either to update the interface when you receive a new version of Suprtool or to re-install the interface after a MIT update from HP.   You will need a small tape for a new COLDLOAD tape to contain the Suprtool interface segment.   You can also install the Suprtool2 interface in a pub or group SL (see the *Compiling and Linking* sections).

System SL
Documentation

Installing into the System SL

1. Make sure that the Robelle account has been created and all files have been restored.

2. Ensure that no one will use the interface until the installation is complete. No one can be running a program which uses the interface. Stop all jobs and send an operator warning.

```
:showjob
:warn @;please stop for 20 minutes
:abortjob #snnn
```

3. STREAM the installation job. Supply any passwords requested by the STREAM command.

```
:stream suprcall.suprjob.robelle
```

4. SUPR2JOB uses the SEGMENTER to add the Suprtool Interface Library into SL.Pub.Sys. It then requests a tape ("COLDLOAD") to create a new cold load tape containing MPE plus the Suprtool interface library. Mount a tape with a write ring and :REPLY. Save this tape and use it for any future cold loads.

5. If everything goes well, SUPR2JOB prints a final message on the console.

6. Please save the job listing for future reference.

Suprtool2 is now installed and you should be able to use it in your application programs.

Documentation

You can use the Printdoc program to print the the user manual for the Suprtool2 Interface (Calling Suprtool). Printdoc is menu-driven and very easy to use. After you run the program, Printdoc will ask you a few questions about your printer. If you are not sure of your answer, ask for help by typing a question mark (?). Printdoc supports all types of LaserJet printers and regular line printers.

```
     :run printdoc.pub.robelle;info="suprcall.doc.robelle"
```

Suprtool2 Error Messages

Suprtool2 returns error numbers in the status parameter of the workspace.   For most errors, a message is also displayed on $stdlist.   The following summarizes the form of Suprtool2 error messages and the error numbers returned.

Stdlist
1
2
3
4
5

Messages On $Stdlist

Most Suprtool2 errors result in a message being displayed on $stdlist.   All Suprtool2 error messages start with "ST2 -" making them easier to identify. For example,

```
     Error:  ST2 - Caller lacks PH capability
```

1 - Unable to Access Files

Suprtool2 uses two temporary files called SI*nnn* and SL*nnn*. The *nnn* corresponds to the process identification number of the calling process. If Suprtool2 is unable to open, read, write, or close these files, a file system message is displayed on $stdlist and error #1 is returned.

2 - Suprtool Aborted

If the Suprtool task has any errors, Suprtool is terminated with the fatal JCW set. In this case, Suprtool2 returns error #2 to indicate that the requested task failed. If the print-state in the workspace was set to "ER" (error) or "AL" (always), the output from Suprtool will be displayed on $stdlist. If the print-state is "NE" (never), nothing is displayed on $stdlist.

3 - Unable to Create Suprtool Process

The Suprtool2 routine creates the Suprtool program as a son process. If this fails for any reason, error #3 is returned. The most common reason for this is forgetting to :PREP or :LINK your program with CAP=PH. If the Suprtool2filecommand JCW is set to 2, check that the :File command for Suprtool correctly specifies the name of the Suprtool program file.

4 - Invalid Total Type

The Suprtool2 control record has a field to indicate what kind of totals the interface should return. The valid values for this field are "CO" for COBOL totals and "AS" for Ascii totals. The total type must be in uppercase. This error is returned when the control record has been passed a value other than "CO" or "AS" for the total type.

5 - Unable to Create Suprtool Process

Suprtool2 calls the PROCINFO intrinsic to obtain the process identification number of the current process. If this routine fails for any reason, the error number returned from PROCINFO is printed on $stdlist and Suprtool2 returns error #5.

How to Contact Robelle

In the United States, in Canada, and in places not listed below, contact us at the following address:

Robelle Solutions Technology Inc.
Suite 201, 15399-102A Ave.
Surrey, B.C. Canada   V3R 7K1

Toll-free: 1.888.robelle
:   (1.888.762.3553)
Phone   :   604.582.1700
Fax   :   604.582.1799

E-mail   :   solutions@robelle.com
E-mail   :   support@robelle.com
Web   :   www.robelle.com

For our international distributors listing, note that the phone and fax numbers shown are for out-of-country dialing.

Europe
Africa
Asia and Australia
North America

Europe

France, Belgium

ARES
Attention: Renee Belegou
Phone: 33 1 69 86 60 24
Fax: 33 1 69 28 19 18
E-mail: rbelegou@ares.fr
Web: www.ares.fr

Germany

SWS SoftWare Systems GmbH
Attention: Renate Pfund
Phone: 49 7621 689 190
Fax: 41 31 981 32 63
E-mail: info@sws.ch
Web: www.sws.ch

The Netherlands, Belgium

Samco Automation b.v.
Attention: Marius Schild
Phone: 31 13 5215655
Fax: 31 13 5288815
E-mail: marius@samco.nl
Web: www.samco.nl

Nordic Countries

Ole Nord AB
Attention: Ole Nord
Phone: 46 8 623 00 50
Fax: 46 8 35 42 45
E-mail: info@olenordab.se
Web: www.olenordab.se

Switzerland, Austria

SWS SoftWare Systems AG
Attention: Renate Pfund
Phone: 41 31 981 06 66
Fax: 41 31 981 32 63
E-mail: info@sws.ch
Web: www.sws.ch

United Kingdom, Ireland

Robelle Consulting
Attention: Clive Oldfield
Phone: +44 20 7473 2558
Fax: +44 20 7473 2558
E-mail: robelle_oldfield@email.msn.com

Africa

South Africa

Synergy Computing (Pty) Ltd
Attention: Paul Howard
Phone: 27 21 685 7809
Fax: 27 21 685 7927
E-mail: synergy@synergy.co.za

Asia and Australia

Australia, New Zealand

MRFM Pty. Ltd.
Attention: Michael Redmond
Phone: +61 3 9629 8633
Fax: +61 3 9629 8062
E-mail: mredmond@mrfm.com.au
Web: www.mrfm.com.au

Hong Kong

SCS Computer Systems Ltd.
Attention: Steven Lai
Phone: 852 2609 1338
Fax: 852 2607 3042

Singapore, Malaysia

Singapore Computer Systems Ltd.
Attention: Toh Tiau Hong
Phone: 65 441 2688
Fax: 65 441 2811
E-mail: tohth@scs.com.sg
Web: www.scs.com.sg

North America

Mexico

Infosistemas Financieros SA de CV
Attention: Anita De Urquijo
Phone: 52 5 813 1325
Fax: 52 5 813 3026
E-mail: adeurquijo@if.com.mx
Web: www.if.com.mx